Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# Investigation of genotype and phenotype interactions using computational statistics

A thesis presented in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Statistics
at Massey University

**Olivia Angelin-Bonnet**

School of Fundamental Sciences
Massey University, Palmerston North
New Zealand

July 2021

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE
RESEARCH
SCHOOL

# Note for Examiners
# Explanation of COVID-19 Impacts

Thank you for taking the time to examine this thesis, which has been undertaken during the Covid-19 pandemic. The New Zealand Government's response to Covid-19 includes a system of Alert Levels which have impacted upon researchers. Our University's pandemic plan applied the Government's expectations to our research environment to ensure the health and safety of our researchers, however, research was impacted by restrictions and disruptions, as outlined below.

For a six-week period from March 26 to April 27 2020, New Zealand was placed under very strict lockdown conditions (Level 4 – Lockdown), with students and staff unable to physically access University facilities, unless they were involved in essential research related to Covid-19. All field work ceased and data collection with humans was restricted to online methods, if appropriate. The restrictions were partially lifted on April 27, but students and staff were not generally allowed back into University facilities until May 13.

Ongoing disruptions have also been encountered for some students due to uncertainties over the potential for future Covid-19-related restrictions on activities, and a Covid-19 cluster outbreak based in Auckland in New Zealand on 12 August 2020 led to the imposition of rolling Level 2 (Reduce) and Level 3 (Restrict) conditions until 23 September 2020. Auckland campus based students remained on Level 2 until 7 October 2020. This Alert Level system continues to be utilised throughout 2021.

These changing Alert Levels have meant that some research students had experimental, clinical, laboratory, field work, and/or data collection or analysis interrupted, and consequently may have had to adjust their research plans. For some students, the impacts of Covid-19 stretched far beyond the lockdown period in April/May 2020, as they may have had to significantly revise their research plans.

Overseas travel is not permitted by the University and restrictions have been placed on the New Zealand borders which are closed to non-New Zealand citizens and permanent residents. This meant that international students who were based offshore at the time of lockdown, were unable to return to New Zealand. A small number of offshore students were provided permission to return to New Zealand in early 2021. Many students have also suffered from anxiety and stress-related issues, and have had financial impacts, meaning their research progress has been significantly delayed.

This form, as completed by the supervisor and student, outlines the extent that the research has been affected by Covid-19 conditions.

Approved by DRC 10/Feb/2021
DRC 21/02/03

**Please consider the factors listed below in your assessment of the work.**

This statement has been prepared by the candidate's supervisor in consultation with the student and has been endorsed by the relevant Head of Academic Unit.

| | | | |
|---|---|---|---|
| Student Name: | Olivia Angelin-Bonnet | ID Number: | 17323636 |
| Supervisor Name: | Dr Matthieu Vignes | Date: | 25-Mar-21 |

Thesis title:

Investigation of genotype and phenotype interactions using computational statistics

**Considerations to be taken into account**. Note: This statement will remain in the final copy of the thesis which will be available from the Massey University Library following the examination process. [*Enter key considerations here for the examiners. This can include but is not limited to change of scope, scale, topic, focus; limitations in relation to data collection, access to necessary literature or archival materials, laboratories, field sites; disruptions as a result of lockdown and various alert levels, medical or health considerations etc*]

The national lockdown in March/April 2020 resulted in a delay of the metabolomics data processing, which I received later than expected. As I was traveling back from an overseas conference two weeks before the lockdown, I had to self-isolate for two weeks, which also delayed my ability to work on my thesis during this time due to difficulties of accessing necessary resources. As a consequence of these delays, I successfully applied for a suspension of studies of one month.

| **Confidential for Examiners Only**: [*Please enter any other considerations which are confidential for examiners only and should not be placed in the final thesis version submitted to Library following the examination process]* |
|---|
| |

Signed, confirming this is a fair reflection of the impact of Covid-19 on this research.

| | | |
|---|---|---|
| Student | Olivia Angelin-Bonnet | Digitally signed by Olivia Angelin-Bonnet<br>Date: 2021.03.25 13:02:33 +13'00' |
| Supervisor | Matthieu Vignes | Digitally signed by Matthieu Vignes<br>Date: 2021.03.25 14:26:00 +13'00' |
| Head of Academic Unit (or nominee) | Catherine Whitby | Catherine Whitby<br>2021.03.26<br>07:41:57 +13'00' |

# Abstract

Deciphering the precise mechanisms by which variations at the DNA level impact measurable characteristics of organisms, coined phenotypes, through the actions of complex molecular networks is a critical topic in modern biology. Such knowledge has implications spanning numerous fields, from plant or animal breeding to medicine. To this end, statistical methods must be leveraged to extract information from molecular measurements of different cellular scales, allowing us to reconstruct the regulatory networks mediating the impact of genotype variations on a phenotype of interest.

In this thesis, I investigate the use of causal inference methods, to infer relationships amongst a set of biological entities from observational data. More specifically, I tackled the reconstruction of multi-omics molecular networks linking genotype to phenotype. In the first part, I developed a simulator that generates benchmark gene expression data, i.e. RNA and protein levels, from synthetic gene regulatory networks. The originality of my work is that it includes transcriptional and post-transcriptional regulation amongst genes. I used the developed simulation tool to evaluate and compare the performance of state-of-the-art causal inference methods in reconstructing causal relationships between the genes. The evaluation focused on the ability of the methods to reconstruct relationships mediated by post-transcriptional regulations from observational transcriptomics data. I also evaluated the methods performance to detect different types of causal relationships between genes via a catalogue of causal queries, and highlighted the shortcomings associated with using transcriptomics data alone in reconstructing gene regulatory networks. In the second part, I developed an analysis framework to shed light on the biological mechanisms underlying tetraploid potato tuber bruising. I first integrated a GWAS analysis with a differential expression analysis on transcriptomics data, to uncover genomic regions in which variations affect the response of tubers to mechanical bruising. I then used a multi-omics integration tool to jointly analyse genomics, transcriptomics, metabolomics and phenotypic data and to identify molecular features across the omics datasets involved in tuber bruising, including some not identified with traditional differential expression analyses. Finally, I made use of causal inference tools to reconstruct a multi-omics causal network linking these features

to decipher the molecular relationships involved in tuber bruising. I used causal queries to extract information from the reconstructed causal networks and interpret the uncovered relationships.

# Acknowledgements

First, I would like to thank my supervisors for their invaluable help and support during my PhD journey: Dr Matthieu Vignes, A/Prof Patrick Biggs, Dr Susan Thomson and Dr Samantha Baldwin. Matthieu, thank you for all the help you provided, not only for the thesis but also everything that comes with it and after. Thank you for your feedback, encouragements, for our discussions and for the games of 7 Wonders. Patrick, I am very grateful for your help throughout this work, your enthusiasm for the project and many ideas, for your eye for the small details that matter, and for your rigour that inspired me. Susan, it was really nice to work with you! Thank you for letting me visit you several times, for the rides to and from the airport, and for all the things you taught me about bioinformatics and polyploids. Sam, thank you for making me a part of this project and for your help about all things potatoes. I am extremely lucky to have had such a great supervisory team. I really enjoyed working with you and learning from you. Our weekly and then fortnightly meetings, and the discussions we had, fuelled my enthusiasm for the project and gave me the inspiration and energy I needed to get to the finish line. Thank you for trusting me with this project, helping me grow as a scientist, and for enduring with good humour the thousand plots I created.

I am grateful for the financial support of the School of Fundamental Sciences, Massey University, for the financial travel support from the Massey University Postgraduate Travel Grant, and for the financial support of the Royal Society of New Zealand (Catalyst CSG-MAU1902 for NIM$^{3}$D project). This work was supported by The New Zealand Institute for Plant and Food Research Limited Strategic Science Investment Fund (SSIF) from the New Zealand Ministry of Business, Innovation and Employment (MBIE). I would also like to thank Plant and Food Research, and in particular Samantha Baldwin and Susan Thomson, for providing the genomics, transcriptomics, metabolomics and phenotypic datasets used in this thesis.

Thank you to the School of Fundamental Sciences team, in particular Anne Truter, Debbie Cresswell, Debbie McKnight, Fiona Richmond, Mark Bebbington, Peter Lewis and Keith Whitehead, for their help and support throughout my PhD. I am also grateful for the support of the Bioinformatics group at Massey University.

Thank you to everyone at Plant and Food that helped me and shared their knowledge with me: Dr Rebecca Bloomer, I really enjoyed the time we spent in the lab together! I really appreciated you trusting me with a pipette (even though it confirmed that I prefer a computer), thank you for showing me what it means to generate data and let me play a small part in it. Thank you also to Katrina Monaghan for supervising my time with the potatoes. Thank you to Martin Shaw and Nigel Joyce for introducing me to metabolomics, for our discussions about the data and your help during their processing. Tim Millar, thank you for showing me really useful tips in Python and GitHub, and sharing fries and beers with the group after a long day of work.

I would like to thank Dinindu Senanayake and Megan Guidry, for their help in setting me up to NeSI, for their enthusiasm for my project and their willingness to let me share this work with others. I also wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities, consulting support and/or training services as part of this research. New Zealand's national facilities are provided by NeSI and funded jointly by NeSI's collaborator institutions and through the Ministry of Business, Innovation & Employment's Research Infrastructure programme.

I am grateful for the help and responsiveness of John Chambers, author of the `XRJulia` R package, and Alfonso Landeros, author of the `BioSimulator` Julia module.

Thank you to my fellow statistics PhD students, Gabriele, Jing, Ahmad and Ghazaleh (all Dr or soon-to-be!) for sharing the good times and the bad, for your advice, help and encouragements, and for all the laughter and adventures we shared. Thank you also to my friends Claire and Rose, for your support and all the good memories we have made along the way.

I am very grateful for the love, help and support of my parents and my family. Thank you for encouraging me when I moved to the other side of the world to work on a topic I was passionate about. This experience helped me grow, and even if I am far away from you, you are always in my heart. Thank you to my cat, Milo, for being my alarm most mornings (even if sometimes said alarm wakes me up at 2am). Finally, saving the best for last, thank you Alexis, my partner and best friend, for your love, kindness and support. You are my biggest supporter, you are always encouraging me during the hard times, and celebrating with me the small and big milestones. You have made this PhD journey and my move to New

Zealand more exciting and enjoyable, and I am extremely grateful to have had the chance to share this experience with you. I love you.

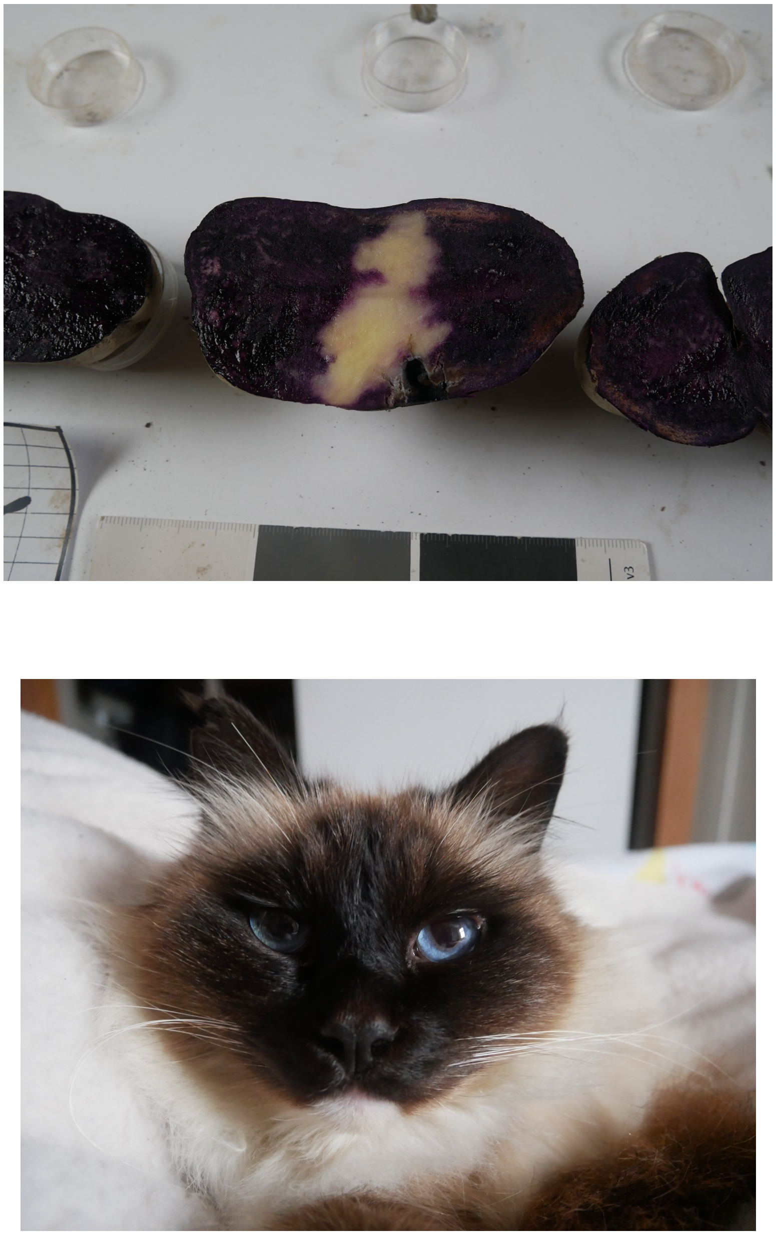


Figure 1: Two pillars of my PhD thesis: potatoes and Milo.

# Table of Contents

**Abstract** . . . . . . . . . . **v**

**Acknowledgements** . . . . . . . . . . **vii**

**List of Figures** . . . . . . . . . . **xiii**

**List of Tables** . . . . . . . . . . **xv**

**List of Abbreviations** . . . . . . . . . . **xvii**

**List of Publications** . . . . . . . . . . **xxi**

**Introduction** . . . . . . . . . . **1**
- 0.1 Aims of the thesis . . . . . . . . . . 3
- 0.2 Structure of the thesis . . . . . . . . . . 3

**Chapter 1: Literature Review** . . . . . . . . . . **7**
- 1.1 Introduction . . . . . . . . . . 7
- 1.2 A biological primer for biostatisticians: from genotype to phenotype . . . . . . . . 8
- 1.3 Statistical tools to bridge genotype and phenotype . . . . . . . . . . 13
  - 1.3.1 Linking genotype to phenotype with QTL mapping and association studies 13
  - 1.3.2 Reconstructing regulatory networks from observational data . . . . . . . . 18
- 1.4 Causal inference . . . . . . . . . . 22
  - 1.4.1 Graph terminology . . . . . . . . . . 22
  - 1.4.2 Causality – mathematical concepts . . . . . . . . . . 23
  - 1.4.3 Causal learning algorithms . . . . . . . . . . 27
  - 1.4.4 Structural Equation Modelling . . . . . . . . . . 33
  - 1.4.5 Causal inference in biological systems . . . . . . . . . . 34
- 1.5 Simulation of Gene Regulatory Networks . . . . . . . . . . 36
  - 1.5.1 Topology of the simulated GRNs . . . . . . . . . . 37
  - 1.5.2 Mathematical framework . . . . . . . . . . 38

1.5.3 Mathematical representation of gene regulation 39
1.5.4 Model simplifications 40
1.6 Conclusion 41

**Chapter 2: Simulating gene regulatory networks and gene expression with the R package sismonr 43**
2.1 Introduction 43
2.2 Creating the *in silico* system 46
2.2.1 Creating the genes 47
2.2.2 Creating the regulatory network 49
2.2.3 Customising the system 54
2.3 Creating the *in silico* population 54
2.3.1 Modelling genetic variation 54
2.3.2 Creating the segregating gene alleles 57
2.3.3 Creating the *in silico* individuals 58
2.4 Simulating the system 59
2.4.1 Generating the stochastic system 59
2.4.2 Stochastic simulation 68
2.5 Comparison of `sismonr` with existing tools 68
2.5.1 Existing GRN simulators 69
2.5.2 Comparison of `sismonr` and `sgnesR` Stochastic Simulation Algorithm implementations 72
2.5.3 Comparison of the simulations of a same regulatory network 73
2.6 Concluding remarks 80

**Chapter 3: Investigating causal inference methods performance in uncovering transcription and post-transcriptional gene regulatory networks 83**
3.1 Introduction 83
3.2 Materials and methods 85
3.2.1 Simulation settings 86
3.2.2 NeSI High Performance Computing Platform 88
3.2.3 Causal inference methods investigated 88
3.2.4 Types of output causal graph 91
3.2.5 Methods tuning parameters 92
3.2.6 Different causal queries and evaluation metrics 94
3.3 Results and Discussion 96
3.3.1 Running time 97
3.3.2 Methods performance when changing the tuning parameters 97
3.3.3 Comparison of methods' performance 100

3.3.4 Reconstruction of post-transcriptional regulation . . . . . . . . . . . . . . 107
3.4 Concluding remarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
3.5 Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116

**Chapter 4: Investigating genotype-phenotype relationships for tuber bruising in autotetraploid potatoes using genomics and transcriptomics data . . . . . . . . . . . . . . 117**
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
4.2 Contribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
4.3 Materials and Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
4.3.1 Plant materials and phenotyping . . . . . . . . . . . . . . . . . . . . . . . . 120
4.3.2 Genotyping . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
4.3.3 Transcriptomics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
4.3.4 Population structure analysis . . . . . . . . . . . . . . . . . . . . . . . . . . 124
4.3.5 GWAS analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125
4.3.6 Differential expression analysis . . . . . . . . . . . . . . . . . . . . . . . . 127
4.3.7 Network co-expression reconstruction . . . . . . . . . . . . . . . . . . . . . 128
4.4 Results and Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
4.4.1 Phenotypes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
4.4.2 Baits quality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
4.4.3 Genomics data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
4.4.4 Transcriptomics data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
4.4.5 Population structure and individual relatedness . . . . . . . . . . . . . . . 136
4.4.6 Impact of GWAS settings on the marker scores . . . . . . . . . . . . . . . . 142
4.4.7 Markers-trait association . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
4.4.8 Comparing GWAS results with a differential expression analysis . . . . . . 149
4.4.9 Genes co-expression network . . . . . . . . . . . . . . . . . . . . . . . . . 151
4.5 Concluding remarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
4.6 Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156

**Chapter 5: Uncovering causal relationships from genotype to phenotype using multi-omics data in autotetraploid potatoes . . . . . . . . . . . . . . . . . . . . . . . . . . . 157**
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 157
5.2 Contribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
5.3 Materials and methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
5.3.1 Plant materials and phenotyping . . . . . . . . . . . . . . . . . . . . . . . . 159
5.3.2 Metabolomics dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
5.3.3 Genomics dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
5.3.4 Transcriptomics dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
5.3.5 Multi-omics data integration . . . . . . . . . . . . . . . . . . . . . . . . . . 164

5.3.6 Causal inference . . . 167
5.4 Results and Discussion . . . 170
5.4.1 Metabolomics . . . 170
5.4.2 Multi-omics data integration with DIABLO . . . 174
5.4.3 Causal inference . . . 181
5.5 Conclusion . . . 199
5.6 Acknowledgements . . . 202

**Chapter 6: General Discussion . . . 203**
6.1 Review of present omics data integration analyses for GRN reconstruction . . . 203
6.2 Benchmarking of statistical causal inference methods for GRN reconstruction . . . 204
6.3 Inference of causal multi-omics networks in tetraploid potato . . . 206
6.4 Future work . . . 208
6.5 Concluding remarks . . . 210

**Appendix A: Supplementary File for Chapter 1 . . . 211**
Abstract . . . 212
A.1 Introduction . . . 212
A.2 Biological processes: from gene to protein . . . 213
A.2.1 Regulation of transcription . . . 214
A.2.2 Regulation of translation . . . 216
A.2.3 Regulation of mRNA decay . . . 218
A.2.4 Regulation of protein activity . . . 218
A.2.5 Regulation of protein decay . . . 219
A.2.6 The role of genetic variation . . . 220
A.2.7 An example: long noncoding RNAs . . . 221
A.3 Modelling gene expression . . . 223
A.3.1 Topological properties of regulatory networks . . . 223
A.3.2 Mathematical frameworks and regulation functions . . . 226
A.3.3 Model simplifications . . . 234
A.3.4 Assigning values for model parameters . . . 236
A.3.5 Experimental noise in *in silico* data . . . 237
A.4 Concluding remarks . . . 238
Acknowledgements . . . 239

**Appendix B: Supplementary Material for Chapter 2 . . . 241**

**Appendix C: Supplementary File 1 for Chapter 2 . . . 245**
Abstract . . . 246
Introduction . . . 246

C.1 Methods: biological system data simulation . . . . . . . . . . . . . . . . . . . . 247
C.2 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 248
C.3 Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 248
Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 249
Funding . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 250
Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 250

**Appendix D: Supplementary File 2 for Chapter 2** . . . . . . . . . . . . . . . . . . . **251**
Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251
D.1 Example 1: Post-transcriptional regulation in a 10-gene system . . . . . . . . . . . 251
D.2 Example 2: A small transduction network . . . . . . . . . . . . . . . . . . . . . 256
D.3 Example 3: The anthocyanin biosynthesis regulation pathway . . . . . . . . . . . . 259
D.4 Example 4: A system of 50 genes . . . . . . . . . . . . . . . . . . . . . . . . . 268
D.5 Anthocyanin biosynthesis regulation pathway simulation settings . . . . . . . . . . 272
  D.5.1 Basal expression rates . . . . . . . . . . . . . . . . . . . . . . . . . . . 272
  D.5.2 Regulation rates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 273
  D.5.3 Complexes rates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 274
  D.5.4 Genetic mutations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 275

**Appendix E: Supplementary Material for Chapter 3** . . . . . . . . . . . . . . . . . . **277**

**Appendix F: Supplementary Material for Chapter 4** . . . . . . . . . . . . . . . . . . **285**
F.1 RNA extraction protocol . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 285

**Appendix G: Supplementary Material for Chapter 5** . . . . . . . . . . . . . . . . . . **295**
G.1 Metabolomics data generation protocol . . . . . . . . . . . . . . . . . . . . . . . 295
  G.1.1 Samples preparation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 295
  G.1.2 Liquid Chromatograph Mass Spectrometry (LCMS) conditions . . . . . . 295

**Appendix H: Contribution to publications and copyright permission** . . . . . . . . . . . **307**

**References** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **319**

# List of Figures

1.1 The different steps of the expression process of a protein-coding gene and its regulation 9
1.2 Example of a DAG . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
1.3 Example of $d$-separation in a DAG . . . . . . . . . . . . . . . . . . . . . . . 24
1.4 Example of Markov equivalent DAGs and corresponding CPDAG . . . . . . . . . 25
1.5 Example of a DAG with latent variables, and corresponding MAG and PAG . . . . 26
1.6 Possible transcription rate law functions . . . . . . . . . . . . . . . . . . . . 39

2.1 Schema of the `sismonr` pipeline . . . . . . . . . . . . . . . . . . . . . . . 46
2.2 The `sismonr` stochastic model – with kinetic parameters . . . . . . . . . . . . . 60
2.3 Simulated regulatory network for `sismonr` and `sgnesR` comparison . . . . . . . . 73
2.4 Simulated expression profiles for `sismonr` and `sgnesR` comparison . . . . . . . . 74
2.5 Boxplot of simulated genes expression for `sismonr` and `sgnesR` comparison . . . . 75
2.6 Simulated regulatory network for simulators comparison . . . . . . . . . . . . . 76
2.7 Distribution of simulated abudance with the different simulators . . . . . . . . . . 77
2.8 Distribution of RNA and protein correlations for the different simulators . . . . . . 78
2.9 Example of RNA vs protein abundance for the different simulators . . . . . . . . . 79

3.1 Schema of analysis workflow for Chapter 3 . . . . . . . . . . . . . . . . . . . 87
3.2 Running time of the different inference methods . . . . . . . . . . . . . . . . . 98
3.3 Optimal tuning parameters values, for the parent and ancestor queries . . . . . . . 101
3.4 Optimal tuning parameters values, for the potential parent and potential ancestor queries 102
3.5 Optimal performance of the methods across configurations and queries . . . . . . . 103
3.6 Methods performance for the parent and ancestor queries . . . . . . . . . . . . . 104
3.7 Methods performance for the potential parent and potential ancestor queries . . . . 105
3.8 Consensus skeleton of inferred graphs for an example network . . . . . . . . . . . 108
3.9 Performance and similarity of skeleton reconstruction . . . . . . . . . . . . . . . 109
3.10 Potential parent query performance for post-transcriptional regulation . . . . . . . 111
3.11 Causal inference on RNA vs protein measurements for two example networks . . . 112

4.1 Schema of analysis workflow for Chapter 4 . . . . . . . . . . . . . . . . . . . 121

4.2 Correlation between measured phenotypes . . . . . . . . . . . . . . . . . . . . . . 130
4.3 Comparison of baits normalised read counts between parent samples . . . . . . . . 131
4.4 Variants depth of coverage, quality and QD score . . . . . . . . . . . . . . . . . . 133
4.5 Upset plot of polymorphic variants in the progeny samples . . . . . . . . . . . . . 135
4.6 PCA plot of the first two components of samples PCA on genotype . . . . . . . . . 138
4.7 PCA plot of the the second and third components of samples PCA on genotype . . 139
4.8 Population structure of parent samples . . . . . . . . . . . . . . . . . . . . . . . 140
4.9 Kinship matrix of the progeny samples . . . . . . . . . . . . . . . . . . . . . . . 141
4.10 Correlation between GWAS marker scores for different GWAS settings . . . . . . . 143
4.11 Inflation factors for the different phenotypes and GWAS settings . . . . . . . . . . 145
4.12 Manhattan plot of GWAS scores for the bruising mean score . . . . . . . . . . . . 147
4.13 Association between a high-scoring marker and neighbouring genes' expression . . 148
4.14 Position of GWAS high-scoring markers and differentially expressed genes . . . . . 152
4.15 WGCNA modules enrichment for genes of interest . . . . . . . . . . . . . . . . . 154

5.1 Schema of analysis workflow for Chapter 5 . . . . . . . . . . . . . . . . . . . . . 160
5.2 tSNE reduced space for metabolomics data . . . . . . . . . . . . . . . . . . . . . 171
5.3 Metabolomics WGCNA modules enrichment scores . . . . . . . . . . . . . . . . . 173
5.4 Two WGCNA modules from the metabolomics co-abundance network . . . . . . . 175
5.5 Features mean rank score in the full DIABLO analysis . . . . . . . . . . . . . . . 178
5.6 Weighted consensus reduced space of the full and GWAS-only DIABLO analyses . 180
5.7 Correlation between selected features and DIABLO latent components . . . . . . . 182
5.8 Number of edges inferred with different tuning parameter values . . . . . . . . . . 184
5.9 Methods skeleton similarity matrix . . . . . . . . . . . . . . . . . . . . . . . . . 185
5.10 Type of edges inferred by the different methods . . . . . . . . . . . . . . . . . . 186
5.11 Consensus skeleton of inferred graphs . . . . . . . . . . . . . . . . . . . . . . . 188
5.12 Number of positive answers to the causal queries per method . . . . . . . . . . . 193
5.13 Type of positive causal queries answers inferred by the different methods . . . . . 194
5.14 Consensus skeleton of inferred graphs with a blacklist . . . . . . . . . . . . . . . 198

A.1 Biological versus statistical representation of a GRN . . . . . . . . . . . . . . . . 214
A.2 The different steps of an algorithm for expression data simulation . . . . . . . . . 222

B.1 Individual runs of `sismonr` simulations . . . . . . . . . . . . . . . . . . . . . . 242
B.2 One individual run of `sismonr` simulations . . . . . . . . . . . . . . . . . . . . 243

C.1 `sismonr` stochastic model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 249

D.1 Anthocyanin biosynthesis regulation schema . . . . . . . . . . . . . . . . . . . . 260
D.2 Anthocyanin biosynthesis regulation pathway simulation . . . . . . . . . . . . . . 266

E.1 Number of edges in the inferred causal graphs . . . . . . . . . . . . . . . . . . . . 278
E.2 Inferred graphs for the parent query for one example network . . . . . . . . . . . . 279
E.3 Inferred graphs for the ancestor query for one example network . . . . . . . . . . . 280
E.4 Inferred graphs for the potential parent query for one example network . . . . . . . 281
E.5 Inferred graphs for the potential ancestor query for one example network . . . . . . 282
E.6 Causal inference on RNA vs protein measurements for two example networks . . . 283
E.7 Causal inference on RNA vs protein measurements for two example networks . . . 284

F.1 Distribution of the measured phenotypes . . . . . . . . . . . . . . . . . . . . . . . 286
F.2 Density of variants, baits and transposons across the chromosomes . . . . . . . . . 287
F.3 Population structure for two families . . . . . . . . . . . . . . . . . . . . . . . . . 288
F.4 Population structure for parent samples for several number of clusters . . . . . . . 289
F.5 Relationship between a high-scoring marker, the corresponding gene'expression and the bruising mean score . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 293

G.1 GWAS score of selected variants with and without accounting for population structure 297
G.2 Weighted consensus reduced space of the full and GWAS-only DIABLO analyses with crosses information . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 298
G.3 Genomic position of variants selected with the full and GWAS-only DIABLO analyses 299
G.4 Consensus skeleton of inferred graphs with feature status and latent component . . 300
G.5 Comparison between WGCNA topological overlap scores and causal inference results for the transcriptomics dataset . . . . . . . . . . . . . . . . . . . . . . . . . . 301
G.6 Comparison between WGCNA topological overlap scores and causal inference results for the metabolomics dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . 302
G.7 Confidence score distribution of positive answers to each causal query . . . . . . . 303

# List of Tables

2.1 List of abbreviations used for `sismonr` . . . . . . . . . . . . . . . . . . . . . . . . 47
2.2 Biological functions of protein-coding and non-coding genes . . . . . . . . . . . . 48
2.3 Sampling distribution of genes kinetic parameters . . . . . . . . . . . . . . . . . . . 48
2.4 Sampling distribution of regulatory complexes kinetic parameters . . . . . . . . . 52
2.5 Genetic mutations modelled with `sismonr` . . . . . . . . . . . . . . . . . . . . . . 55
2.6 Sampling distribution of edges kinetic parameters . . . . . . . . . . . . . . . . . . . 56
2.7 Sampling distribution of QTL effect coefficients . . . . . . . . . . . . . . . . . . . 58
2.8 Possible active forms of regulators in the stochastic model . . . . . . . . . . . . . 61

3.1 Different configurations for gene expression simulation . . . . . . . . . . . . . . . 86
3.2 Output graph and tuning parameters of the causal and network inference methods evaluated . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
3.3 Tested values for causal inference tuning parameters . . . . . . . . . . . . . . . . . 93
3.4 Types of edges considered to answer each causal query . . . . . . . . . . . . . . . 95

5.1 Tuning parameters values tested for the causal inference . . . . . . . . . . . . . . . 183
5.2 High-confidence edges in consensus skeletons . . . . . . . . . . . . . . . . . . . . 190

A.1 Overview of existing methods of expression data simulation and their characteristics 235

B.1 Genes' properties for the `sismonr` vs `sgnesR` comparison . . . . . . . . . . . . . 244

D.1 Anthocyanin biosynthesis pathway RNA ratios . . . . . . . . . . . . . . . . . . . . 267
D.2 Genes kinetic parameters for the anthocyanin biosynthesis regulation pathway . . . 273
D.3 Regulation kinetic parameters for the anthocyanin biosynthesis regulation pathway 274
D.4 Regulatory complexes kinetic parameters for the anthocyanin biosynthesis regulation pathway . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 274
D.5 QTL effect coefficients for the anthocyanin biosynthesis regulation pathway . . . . 275

F.1 Detected significant markers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 290
F.2 Enrichment of co-expression modules for GO terms . . . . . . . . . . . . . . . . . 292

G.1 High-confidence edges in consensus skeleton with blacklist . . . . . . . . . . . . . . 304

# List of Abbreviations

**Statistical/mathematical terms**

| | |
|---|---|
| CME | Chemical Master Equation |
| FDR | False Discovery Rate |
| FN | False negative |
| FP | False positive |
| IHW | Independent Hypothesis Weighting |
| ODE | Ordinary Differential Equation |
| PC | Principal component |
| RMSE | Root Mean Squared Error |
| SD | Standard deviation |
| TN | True negative |
| TP | True positive |

**Biology-related terms**

| | |
|---|---|
| DA | Differentially abundant |
| DE | Differentially expressed |
| DNA | Deoxyribonucleic acid |
| eQTL | expression QTL |
| GO | Gene Ontology |
| GRN | Gene Regulatory Network |
| GWAS | Genome-Wide Association Study |
| KEGG | Kyoto Encyclopedia of Genes and Genomes |
| LCMS | Liquid Crystallography-Mass Spectrometry |
| lncRNA | long non-coding RNA |
| miRNA | micro RNA |
| mQTL | metabolite QTL |
| mRNA | messenger RNA |

| | |
|---|---|
| PD | Protein decay |
| PPO | Polyphenol oxidase |
| PTM | Post-transcriptional modification |
| QD score | Quality score normalised by allele depth |
| QTL | Quantitative Trait Locus |
| RBP | RNA-binding protein |
| RD | RNA decay |
| RISC | RNA-induced silencing complex |
| RNA | Ribonucleic acid |
| RPKM | Reads Per Kilobase (of transcripts) per Million mapped reads |
| siRNA | silencing RNA |
| SNP | Single Nucleotide Polymorphism |
| TC | transcription |
| TF | Transcription factor |
| TL | Translation |
| UTR | Untranslated region |
| VCF | Variant Call Format |

**Graphs terminology**

| | |
|---|---|
| BIC | Bayesian Information Criterion |
| CPDAG | Completed Partially Directed Acyclic Graph |
| DAG | Directed Acyclic Graph |
| MAG | Maximal Ancestral Graph |
| PAG | Partial Ancestral Graph |
| SEM | Structural Equation Modelling |

**Algorithms**

| | |
|---|---|
| ARACNe | Algorithm for the Reconstruction of Accurate Cellular Networks |
| ARGES | Adaptive Restricted Greedy Equivalence Search |
| DAPC | Discriminant Analysis of Principal Components |

| | |
|---|---|
| DIABLO | Data Integration Analysis for Biomarker discovery using Latent cOmponents |
| FCI | Fast Causal Inference |
| FGEs | Fast Greedy Equivalence Search |
| GENIE3 | Gene Network Inference with Ensemble of Trees |
| GES | Greedy Equivalence Search |
| GNW | GeneNetWeaver |
| MMHC | Max-Min Hill Climbing |
| MMPC | Max-Min Parent and Children |
| PC | Peter-Clark |
| PCA | Principal Component Analysis |
| RFCI | Really Fast Causal Inference |
| SGS | Spirtes-Glymour, Scheines |
| SSA | Stochastic Simulation Algorithm |
| tSNE | t-distributed Stochastic Neighbour Embedding |
| VST | Variant Stabilising Transformation |
| WGCNA | Weighted Correlation Network Analysis |

**Other**

| | |
|---|---|
| DREAM challenge | Dialogue for Reverse Engineering Assessments and Methods challenge |
| HPC | High Performing Computer |
| PFR | Plant and Food Research |
| QC | Quality control |
| SBML | Systems Biology Markup Language |
| SLURM | Simple Linux Utility for Resource Management |
| TOM | Topological Overlap Matrix |

# List of Publications

**Presented in Appendix A** (Supplementary File for Chapter 1)

Olivia Angelin-Bonnet, Patrick J Biggs, and Matthieu Vignes. Gene regulatory networks: a primer in biological processes and statistical modelling, in *Gene Regulatory Networks*, pp. 347-383. Humana Press, New York, NY, 2019. https://link.springer.com/protocol/10.1007/978-1-4939-8882-2_15

**Presented in Appendix C** (Supplementary File for Chapter 2)

Olivia Angelin-Bonnet, Patrick J Biggs, Samantha Baldwin, Susan Thomson, and Matthieu Vignes. "sismonr: simulation of *in silico* multi-omic networks with adjustable ploidy and post-transcriptional regulation in R." *Bioinformatics* 36, no. 9 (2020): 2938-2940. https://doi.org/10.1093/bioinformatics/btaa002

**Not presented in this thesis**

Olivia Angelin-Bonnet, Patrick J. Biggs, and Matthieu Vignes. "The sismonr Package: Simulation of In Silico Multi-Omic Networks in R." In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2729-2731. IEEE, 2018. https://doi.org/10.1109/BIBM.2018.8621131

# Introduction

One key challenge of modern biology is to understand the make-up, organisation and functioning of biological organisms. It was established in 1944 that DNA carries instructions about the development and functioning of organisms. Researchers have since been naturally interested in deciphering how this information is processed to give rise to the organisms' specific phenotypic (i.e. observable) characteristics, e.g. yield of a crop, colour of a plant's flowers, or muscle mass in animals. In particular, the question of what makes two individuals similar, and what drives their difference, has attracted a lot of attention. We now know that phenotypic traits are controlled by an ensemble of complex interactions occurring between molecular actors at different scales in cells. However, our understanding of biological systems is still incomplete, and a lot is yet to be learned about the specific mechanisms at play. **Precisely characterising the relationship between genotype and phenotype, that is, how the information flows from the former to affect the latter, is a crucial step in answering diverse biological challenges.** For example, understanding how variations at the genome level drive the development of diseases can help in diagnostics and prediction of disease outcome, and can lead to the development of cures. Similarly, animal or plant breeding programs benefit from knowledge of genome-level regulatory mechanisms of traits of interest, such as milk production in dairy cattle, or yield of a crop. Such mechanistic knowledge helps accelerate the selection process by selecting individuals with an advantageous genetic make-up for the concerned traits, rather than relying on phenotypic measurements that take a long time to obtain. A lot of progress has been made in the case of simple traits for model organisms, but more work is needed for more complex phenotypes and challenging organisms, in particular polyploids.

Research in this domain has been fuelled by the development of numerous technologies that have enabled scientists to directly observe the genomic content of cells (e.g. SNP genotyping, shotgun sequencing), and to measure the levels of expression of different molecular actors such as RNAs or proteins (for example with RNA-sequencing or liquid chromatography-mass spectrometry). The massive datasets produced by such technologies must then be interpreted and combed to extract useful information. This task necessitates the development of statistical and computational tools that can handle the unique nature of modern biological datasets. Recently, improvements in measurement capabilities have permitted research to shift from a molecule-centred paradigm, where a few molecular features were investigated, to a holistic approach: the goal is now to gain a comprehensive view of the systems at play. Integrating genome- and cell-wide measurements obtained at different levels in cells – termed “omics” datasets – is an area under active development. This Systems Biology approach needs to deal with a number of challenges: the high dimensionality of the datasets, in which several thousands of molecules are measured, the heterogeneity of the measurements, the presence of noise and technical biases arising from the experiments, and the existence of complex biological processes potentially unknown that shape the information contained in the datasets. Consequently, the computational tools used for analyses must address appropriately these challenges. Additionally, there is a need to move from association studies, which assess potential involvement of molecules to a given mechanism, to understanding the causal flow through the different molecular scales that link genotype and phenotype.

In this thesis, I approach the problem of understanding genotype-phenotype interactions from two different angles.

The first concerns consolidating existing statistical tools to the unique challenges posed by biological datasets. Indeed, these tools often rely on a set of assumptions that are often violated in experimental settings. Moreover, their performance is often evaluated on simulated data that overlooks the unique complexity of biological datasets. It is therefore crucial to assess their potential to extract information from experimental datasets. This can be done through the development of appropriate models that can be used to generate simulated data resembling the biological systems investigated. Plausible simulated datasets can then be used as benchmarks to objectively assess the performance of statistical tools developed, provide an (optimistic) estimate of their performance with real data and offer insight into potential areas of improvement. In this work, I focus on a set of mechanisms termed post-transcriptional regulation (defined in Chapter 1) which play a major role in shaping the genotype-phenotype relationship, yet have mostly been overlooked in existing modelling and simulations. I then focus on assessing the performance of statistical methods that are typically used to reconstruct causal relationships between molecular features. My goal is to evaluate their ability to extract information from biological datasets about the precise molecular mechanisms at play.

In the second part, I apply the theoretical framework just mentioned to an experimental dataset obtained from potato. Potatoes, along with a number of other crops and plants (such as wheat, cotton, blueberry, coffee, kiwifruit, rose), present unique challenges for genetic studies, as they are polyploids, i.e. they carry more than two copies of each chromosome. Polyploidy entails a number of specific genetic mechanisms that complicate both the acquisition of data and their analysis. Progress in sequencing capabilities renders the study of polyploid organisms feasible, but adequate analysis tools are still scarce. In this thesis, I focus on reconstructing the genotype-phenotype interactions in potato in the context of tuber bruising. Tuber bruising refers to the apparition of a brown area on the tuber flesh below the skin as the result of a mechanical impact, which affects the appearance and flavour of the tubers. It is therefore a trait of economic importance, and the elucidation of underlying mechanisms can inform breeding programs and lead to the selection of bruising resistant cultivars. Here, I combine a genotype-phenotype association study with the integration of different omics datasets in order to reconstruct the causal relationships between different molecular actors involved in the process of tuber bruising.

## 0.1 Aims of the thesis

The objective of the thesis are:

- To construct a simulator to generate benchmark datasets for the evaluation of statistical methods designed to reconstruct molecular regulatory networks. The simulator must take into account the complexity of biological systems, in particular post-transcriptional regulations of gene expression, and the impact of genetic variation on the system. The tool must also be applicable to polyploid organisms.
- To evaluate the performance of causal inference methods in the context of reconstructing gene regulatory networks. In particular, the comparison should focus on the performance of the methods in presence of post-transcriptional regulation, and highlight areas of improvement.
- To investigate genotype-phenotype relationships in a polyploid organism. The goal is to reconstruct the causal flow of information from the genotype through different omics layers that influences tuber bruising in autotetraploid potatoes. This will be achieved by combining association studies and reconstruction of multi-omics causal networks to shed light on the molecular mechanisms underlying tuber bruising in the potato.

## 0.2 Structure of the thesis

This thesis is organised as follows.

In Chapter 1, I start by presenting the biological processes that link genotype and phenotype. I then discuss the different methods that have been proposed to elucidate these processes, namely QTL mapping and association studies, and network inference methods. I introduce the topic of multi-omics data integration. I then detail the statistical approaches used to infer causality, and their application to a biological context. Lastly, I review existing tools for generating simulated datasets, and highlight gaps that must be addressed.

In Chapter 2, I present the development of the R package `sismonr`, a simulator of gene expression (including levels of mRNAs and proteins) that accounts for post-transcriptional regulation, genetic mutations and the ploidy of the system. The details of the algorithms are presented first. I explain the generation of synthetic networks of gene expression regulation, and discuss the sampling of parameter values in order to obtain realistic systems. Then, I detail how a set of *in silico* individuals are generated, in order to include the impact of genetic variations between the individuals on the system. Lastly, I present how the created systems are simulated, in order to obtain simulated mRNA and protein levels over time for each of the genes in the system, across the *in silico* individuals. Secondly, I compare `sismonr` with existing simulators of gene expression and highlight its novel features. The published Application Note presenting `sismonr` is replicated in Appendix C. Appendix D showcases the use of `sismonr` through illustrated examples, with the corresponding code in order to allow the reproduction of these examples. Note that a detailed tutorial showcasing the use of `sismonr` has been made available online at `https://oliviaab.github.io/sismonr/`.

In Chapter 3, I use `sismonr` to generate a set of benchmark datasets in order to assess the performance of different causal inference tools in reconstructing gene regulatory networks from observational data. The simulated datasets span different regulation scenarios, including different types of transcription and post-transcriptional regulation amongst the genes. I first detail the generation of the datasets with the help of the New Zealand eScience Infrastructure (NeSI) high-performance computers, and present the different statistical methods considered for this evaluation. I also explain the metrics used to assess their performance in reconstructing directed gene regulatory networks. I then contrast their performance across the different simulation scenarios, and discuss the use of protein measurements as opposed to mRNA levels only for the detection of post-transcriptional regulation.

In Chapter 4, I perform an association study on genomics and phenotypic data obtained from a population of half-sibling autotetraploid potatoes. I investigate the genomic regions associated with several traits, with a particular focus on tuber bruising. Moreover, I assess the impact of correcting for population structure in the case of a complex population make-up, and the use of different genetic models to represent the association of SNPs with a trait of interest. I combine the results of the association study with insights gained from transcriptomics data. I perform a differential analysis to assess the transcripts whose levels vary between tubers with low bruising response and tubers with a high bruising response. I compare them to the markers found associated with tuber bruising. I also reconstruct a co-expression network and investigate the distribution of differentially expressed transcripts into highly-coexpressed modules. This analysis provides a first insight into the genetic contribution of potato tuber bruising. It highlights the genomic regions in which association with the phenotype is observed both at the level of genetic variations and of gene expression, and thus provides clues about the molecular mechanisms linking genetic mutations with changes in the response to bruising.

In Chapter 5, I integrate genomics, transcriptomics, metabolomics and phenotypic measurements in order to reconstruct a causal multi-omics network, which depicts molecular mechanisms involved in tuber bruising. This builds on the work presented in Chapter 4, in which I focused on the genetic aspect of tuber bruising, by making use of genomics and gene expression datasets. In Chapter 5, I combine these results with metabolomics measurements to reconstruct biological mechanisms of tuber bruising across the omics layers. I use a multi-omics integration algorithm to select features from the genomics, transcriptomics and metabolomics datasets linked with tuber bruising and which vary together. This common variation is an important feature to detect as it arises from the interactions of different molecular actors across omics layers involved in common mechanisms. I compare the selected features to results obtained from single-omics analysis, namely GWAS and differential analyses. I then apply the causal inference methods presented in Chapter 3 to the set of selected features, in order to assess the causal relationships between them. The results of the feature selection step and causal reconstruction are compared to existing knowledge.

Finally in the General Discussion, I summarise and reflect upon the findings of this thesis. I also share possible directions for future research.

# Chapter 1

# Literature Review

## 1.1 Introduction

Understanding how the information encoded in an organism's DNA is related to its phenotype, i.e. its observable characteristics, is a crucial aspect of modern biology, with implications in many fields: from improving plant selection for pathogen or disease resistance, to personalised medicine by assessing optimal treatments for each patient, to name only two examples. Such knowledge offers exciting insights into functioning of cells, and allows us to decipher complex molecular mechanisms involved in a number of processes, from the development of a disease to the make-up of a trait of economic importance in a cultivated crop. It however requires an understanding of how cells process the information stored in the DNA. We postulate that this understanding can be obtained from statistical and computational methods that extract meaningful information from experimental datasets.

In this review of the literature, I provide first an overview of the biological context of this thesis. I notably explain the concept of gene expression regulation, the crosstalk between the different molecular layers within cells, and the impact of genetic variation on these molecular interactions. Secondly, I summarise two orthogonal statistical approaches to bridging genotype and phenotype: (i) studies of association between genetic and phenotypic variation, and (ii) reconstruction of regulatory networks from observational molecular measurements. I stress the importance of the integration of multi-omics datasets to reconstruct molecular networks spanning different cellular layers. I argue that the inference of causal relationships among molecular actors is key to understanding the flow of information within cells, and can be used to bring together these two complementary approaches. Hence, the mathematical concepts of causal inference are presented in a third section. I also showcase existing studies that have made use of causal inference in a multi-omics context. Finally, I approach the topic of gene regulatory network simulation. Such tools are necessary to assess the performance of methods aiming at reconstructing regulatory networks from experimental observations. I discuss existing simulators and their shortcomings.

## 1.2 A biological primer for biostatisticians: from genotype to phenotype

### Genes and gene expression

DNA, which stands for deoxyribonucleic acid, is the molecule carrying the genetic instructions essential to the functioning of cells. It is composed of two antiparallel chains of nucleotides, intertwined to form a double helix. Each nucleotide is composed of a sugar and a phosphate group, as well as one of four possible chemical bases: adenine, thymine, guanine and cytosine, abbreviated as A, G, C and T, respectively. The chemical bases that are opposite on the two chains pair up, in a specific order: adenine pairs up with thymine, while guanine pairs up with cytosine. Taken together, the sequence of chemical bases, and the pattern they generate, encode genetic information. This genetic information is passed between generations from parent to offspring, which is the basis of heredity.

Some portions of the DNA, termed genes, contain instructions for the synthesis of functional molecules. The complex multi-step process of decoding this information and using it to produce molecules is coined gene expression. Firstly, the information encoded by a given gene is copied by an enzyme termed RNA polymerase, to produce molecules of ribonucleic acid or RNAs. This is the transcription step. A molecule of RNA contains a copy of the sequence of chemical bases encoded in the gene, with the thymine bases replaced with uracil bases (abbreviated as U). In some cases, the resulting RNAs are templates used to produce proteins. In this case, we refer to them as messenger RNAs or mRNAs, and the corresponding genes are coined protein-coding genes. In other cases, the RNAs are not used as templates, but rather play a functional role in cells. The corresponding genes are then termed non-coding genes.

For protein-coding genes, the mRNAs synthesised during transcription are then processed and transported to ribosomes, that are cellular machineries via which mRNAs are translated into proteins. During this step, called translation, each consecutive triplet of chemical bases (referred to as a codon) on an mRNA is translated into a specific amino acid. The correspondence between each possible codon and a specific amino acid is called the genetic code. The resulting chain of amino acid constitutes the synthesised protein. Once synthesised, the proteins are dispatched into the cells or cellular compartments to perform a variety of function, according to a small signal contained within the proteins' sequence (Rapoport, 2007); from enzymes that catalyse specific metabolic reactions, to structural proteins that maintain the integrity of the cells or extracellular proteins that serve as signalling molecules. It must be noted that in addition to protein-coding genes, the genome also contains non-coding regions that serve other purposes, e.g. regulatory regions, or genes used to produce RNAs that will not be translated but rather perform diverse regulatory functions, as will be shown later.

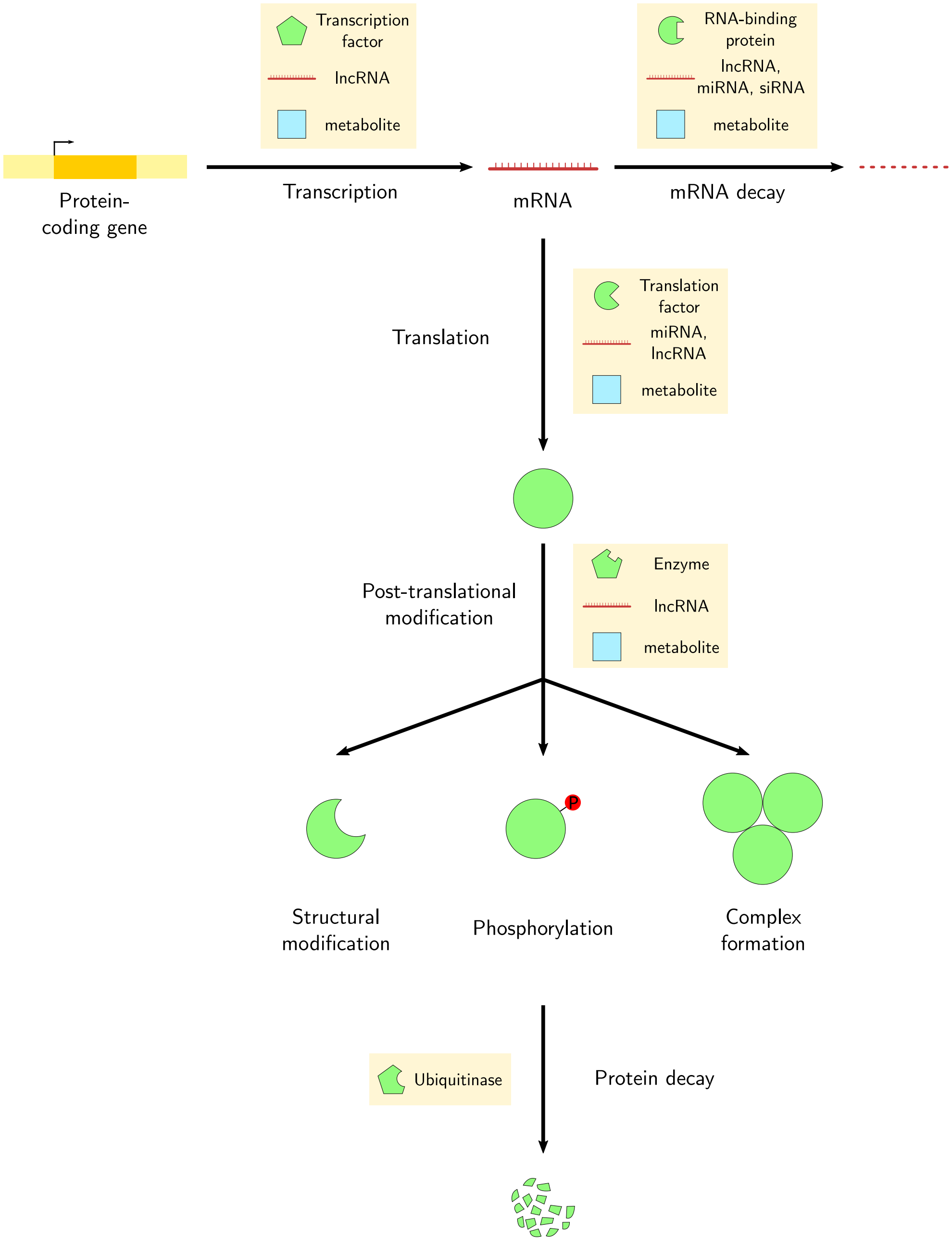


Figure 1.1: The different steps of the expression process of a protein-coding gene, and its possible regulatory molecules. The colors represent the different molecule types: yellow: DNA, red: RNA, green: protein, blue: metabolite. A gene is first transcribed into a mRNA, with the possible involvement of transcription factors, long noncoding RNAs (lncRNAs) or metabolites. The mRNA is then processed and translated into a protein; again this process can be affected by translation factors, microRNAs (miRNAs), lncRNAs, or small molecules. The degradation of transcripts is influenced by noncoding RNAs, RNA-binding proteins or metabolites. Once synthesised, a protein can undergo post-translational modifications, mediated by other proteins, lncRNAs or other small metabolites. Possible modifications include conformational change, modification of specific residues such as phosphorylation, or the formation of protein complexes. Proteins are tagged to degradation by specific enzymes, termed ubiquitinases.

Cells respond and adapt to environmental changes or intracellular cues by modulating the expression of their genes and hence the pool of available proteins and other non-translated RNAs. Notably, changes in the concentration of available enzymes affects the flux of metabolites (through metabolic reactions) and thus the concentrations of small compounds in the cell. To permit this cellular adaptability, the expression of each gene is a highly regulated process at each step (Figure 1.1). This regulation can target the transcription of a gene, to control the amount of produced mRNAs; its translation, in order to increase or decrease the pool of proteins produced; or even the decay of the gene's products (i.e. mRNAs and proteins) to quickly modulate their concentration in the cell. In addition, post-translational modifications of proteins can ensure the latter attain their functional state, by means of a physical modification of their sequence (e.g. cleavage of a peptidic chain or attachment of a specific chemical group), or a change in their conformation (i.e. their 3D structure). Such modifications can also inactivate the proteins. Additionally, proteins can assemble to form complexes, which will perform different roles in the cell. Regulation of gene expression can be performed by specialised proteins, non-coding RNAs, or small molecules termed metabolites. I detail some important mechanisms of gene expression regulation thereafter. Ultimately, the pool of gene products and metabolites or other small compounds resulting from these complex regulations give rise to the phenotypic traits of organisms. Importantly, these mechanisms of gene expression regulation are not always linear, but often relies on feedback loops to obtain specific patterns of expression (Alon, 2007).

### Regulation of gene expression and molecular networks

One major way for cells to control the expression of their genes is through the regulation of transcription by regulator proteins termed transcription factors (TFs) (Pai et al., 2015; Zlatanova & Van Holde, 2016). This phenomenon has been extensively studied. Transcription factors control the expression of their target selectively, by binding to specific motifs in the regulatory regions of the target genes. Once bound to the regulatory regions, they activate or increase the transcription of the target gene by recruiting the transcriptional apparatus. Alternatively, their presence can hinder the transcription of the target. In the former scenario, the TF is referred to as an activator, while in the latter scenario it is termed a repressor. One TF often controls the expression of numerous targets. Conversely, several TFs can regulate a same gene, either independently, cooperatively or in competition (Balaji et al., 2006). We organise our knowledge of such regulatory interactions with graphs, that are termed transcriptional regulatory networks (Blais & Dynlacht, 2005). In these graphs, nodes represent genes or gene products, and an edge is drawn from one node to another if there exists some evidence that the product of the first gene regulates the transcription of the second gene. Reconstructing such regulatory networks is key to bridging genotype and phenotype, and will be discussed in later sections. In addition to TFs, other molecules can play a role in regulating genes transcription. In the past decades, discoveries have hinted at the important role played by non-coding RNAs in regulating gene expression (see e.g. Castel & Martienssen, 2013; Geisler & Coller, 2013). For example, a class of small non-coding RNAs, called microRNAs or miRNAs, have been found to silence the transcription of their target (Catalanotto et al., 2016). Long non-coding RNAs (lncRNAs) are also thought to play a role in transcription regulation via interactions with the transcriptional machinery (Mercer et al., 2009).

Regulation also occurs at later stages of gene expression. RNA-binding proteins (RBPs) and non-coding RNAs (specifically miRNAs) can facilitate or prevent the translation of mRNAs, by binding to specific regions on the mRNAs (Gebauer & Hentze, 2004; Merchante et al., 2017). RBPs act by interacting with the translational apparatus, while miRNAs recruit molecular complexes to either silence the mRNAs translation or target them for degradation. Similar to the mode of action of some miRNAs, regulatory proteins can also trigger the degradation of target mRNAs by recruiting decay factors, which is yet an other indirect way to control the concentration of proteins available (Wang et al., 2002). Likewise, proteins can also be tagged for degradation by regulators (Lecker, 2006; Varshavsky, 2005). This is usually done by enzymes that add a specific chemical group to the targeted protein, which is then recognised and processed by the degradation machinery of the cell. Lastly, synthesised proteins can be modified in order to modulate their activity (Cooper, 2000; Walsh et al., 2005).

Such post-translational modifications provide the very building blocks of many signalling cascades that allow cells to react to the detection of extracellular compounds, for example (Hunter, 1995; Lizcano & Alessi, 2002). In a signalling cascade, the detection of a specific cue by a membrane protein or other detector triggers the modification of a first signalling protein, which in turn targets a second signalling protein for post-translational modification, *et cetera*. This permits to relay the signal from the cellular membrane to the location where a response is needed, e.g. in the nucleus for example to trigger changes in the expression of relevant genes. In some cases, a post-translational modification is necessary to allow newly synthesised proteins to become fully functional, e.g. through cleavage of a peptide bond to reveal the protein functional site. On the contrary, the modification can prevent the protein to perform its function. Post-translational modifications notably play an important role in modulating the activity of transcription factors, which provides an additional layer of regulation for the target genes expression. It is frequent that the endpoint of signalling cascades are TFs, and an activation of the cascade ultimately triggers a change in their activity. Taken together, these different pathways to gene expression regulation offer distinct dynamics by which the cell can respond efficiently and precisely to stimuli.

### The impact of genetic variation

Together with environmental factors, genetic variation between individuals, i.e. small differences in the information encoded in their respective genome, is a major cause of the difference between their phenotype (Albert & Kruglyak, 2015). Studying how genetic variations affect certain traits of interest allows us to gain a better understanding of how the information encoded in the genome is processed by cells to give rise to these traits. There are a number of ways by which a change in the genome can affect a given phenotype, and much is yet to be discovered on this topic. For example, changes in the regulatory sequences of a gene can affect the binding of regulators or of the transcriptional machinery, and thus impact the level of expression of the gene. On the other hand, variations in the coding region of a gene can affect the viability or properties of the resulting proteins. If the protein is a transcription factor for example, the change can impact its affinity for its target binding sites, thus affecting the strength of the regulation effected by the proteins. All these changes occurring at the genomic level are propagated through the complex networks of gene expression regulation, with consequences for higher cellular layers such as proteins or metabolites, and ultimately phenotypes. Therefore, understanding the role played by different regions of the genome, as well as the networks of regulation occurring between the different molecular layers of RNAs, proteins and metabolites, is the key to reconstructing the flow of information from genotype to phenotype. This is the very topic of this thesis. In particular, I focus on a type of organisms with a distinctive genetic organisation, polyploids.

### Genetics of polyploids

Polyploid organisms possess multiple (more than two) copies of each chromosome in their nucleus. This stands in contrast with bacteria that are haploids (one copy) or humans that are diploids (two copies). Polyploidy is common in plants, and also present in some fishes and amphibians (Woodhouse et al., 2009). In particular, a number of economically important crops exhibit diverse degrees of polyploidy, such as wheat, oat, cotton, potato, alfalfa or coffee. Polyploidy arises in evolution when an extraordinary genomic event leads to the formation of gametes with more than the required number of chromosomes. It confers some advantages to the concerned organisms. One example is a potentially higher level of heterozygosity (genetic variability) than their diploid counterpart (Osborn et al., 2003), as for each gene a polyploid can carry potentially more than two different versions, termed alleles. This can yield a greater potential for adaptation to changing environment (Z. J. Chen, 2007). In addition, the presence of several copies of a gene provides a measure of redundancy, thus protecting against the effects of deleterious mutations (i.e. having a negative impact on the organism). In addition, this redundancy provides the means for gene function diversification through evolution, as additional copies of a gene can be changed without affecting the organism (Woodhouse et al., 2009). However, because of their higher genetic complexity, polyploids have been less studied than traditional haploid or diploid model organisms. However, advances in genotyping technology as well bioinformatics and statistical tools are enabling researchers to study these complex organisms.

Polyploids can arise from the hybridisation of two different species; they are then termed allopolyploids (Renny-Byfield & Wendel, 2014). In such case, the chromosomes inherited from the two species – termed homoeologous – form two distinct groups, with chromosomes inherited from the same species being more similar between them (homologous chromosomes) than to the chromosomes inherited from the other species. Therefore, during meiosis, the more similar chromosomes will preferentially pair. This entails a pattern of allele inheritance – termed disomic inheritance – similar to what is observed in diploid species, as recombination (i.e. the exchange of genetic material between the chromosomes) occurs exclusively among these pairs. Conversely, polyploidy can emerge from a whole genome duplication event. The resulting organisms are named autopolyploids (Parisod et al., 2010). In this case, the different copies of a same chromosome are all homologous, i.e. all equally similar. Therefore, each possible homologue pair has the same probability of pairing during meiosis. This increases the number of possible allele combinations that will be transmitted to the gametes. In addition, multivalents can form during meiosis, meaning that more than two chromosomes simultaneously assemble. This can lead to a phenomenon termed double reduction (Bourke et al., 2015), in which both chromatids (arms of a chromosome) from the same chromosome are passed on to the gamete. Again, this phenomenon generates unique patterns of allele inheritance – referred to as polysomic inheritance – and must be accounted for during genomic data analysis.

## 1.3 Statistical tools to bridge genotype and phenotype

As mentioned in the previous section, precisely characterising the flow of information from genotype to phenotype implies reconstructing networks of molecular interactions and assessing through which parts of these networks genetic variations affect phenotypic traits. Experimentally, it would be extremely time consuming and expensive to test one by one all the interactions between gene products, or to conduct experiments to investigate the effect of a given genetic perturbation on a resulting phenotype of interest. A more effective approach is to make use of observational data. In this section, I present two main statistical methods to bridge genotype and phenotype: (i) QTL mapping and association studies, and (ii) molecular regulatory networks inference.

### 1.3.1 Linking genotype to phenotype with QTL mapping and association studies

Understanding which genes or genomic regions control phenotypic traits of interest would pave the way for the quest of precisely describing the genetic mechanisms at play. Traits can be binary, typically the presence/absence of a certain characteristic or of a disease. Traits can also be quantitative, as for example the fat percentage of milk in dairy cattle, or the yield of a certain crop. For this review, I focus on the latter; note that from a statistical perspective the difference between the two situations (i.e. binary vs quantitative trait) can be compared to going from a logistic regression to a traditional linear model. To assess which genomic regions impact a given phenotype, one approach is to capitalise on the genetic variability segregating in populations of interest, and use it to assess how genetic variation affects the investigated traits. This is complicated because a lot of phenotypic traits are controlled by multiple possibly interacting causal loci (i.e. genomic positions or regions) in the genome, each with a quantitative impact of varying magnitude on the trait. Two complementary methods have been explored: QTL mapping and association studies.

### QTL mapping

Historically, researchers made use of carefully designed mapping populations. These mapping populations, typically crosses between inbred lines, maximise the extent of linkage disequilibrium in the genome, i.e. the non-random association of alleles at different loci, mostly due to their physical proximity in the genome. In this case, it is possible to use the genotype of measurable molecular markers at known positions in the genome as proxy for the genotype of unobserved truly causal neighbouring loci (Collard et al., 2005). One common type of markers used are single nucleotide polymorphisms or SNPs, i.e. mutations affecting a single nucleotide in a DNA sequence, that can be found in a non-negligible fraction (i.e. more than one percent) of the population of interest. By comparing the allelic frequency of the molecular markers to expected patterns of allele segregation, it is possible to reconstruct a linkage map that gives the relative position and distance between the measured markers. Linkage maps differ from physical maps in the information they provide. While physical maps inform about the absolute genetic distance in base-pairs between genes for example, a linkage map informs about the recombination frequency between the genes, which is related to but not entirely defined by absolute genetic distance. Then, a statistical model is used to assess the difference in the mean phenotypic values recorded for groups of individuals with different genotypes at a given marker (Jansen, 2008). This association is generally tested with a t-test, an ANOVA model, a linear regression or a likelihood ratio test (Boopathi, 2013). More advanced models, which make use of multiple markers simultaneously, have also been proposed (Boopathi, 2013). They allow to handle missing data by using neighbouring markers, and to account for the effects of other potential QTLs in the regression.

One major drawback of QTL mapping is the large uncertainty in the position of the inferred QTLs, due to the high linkage between neighbouring markers. Detected QTL regions can span large genomic regions and contain many potential candidate causal genes (Mackay & Powell, 2007). Another problem is the lack of generalisation of the results as only a small fraction of all possible alleles are observed in a given mapping population (Bazakos et al., 2017). Lastly, in some cases, it is not possible to construct mapping populations to perform QTL mapping. This is for example impossible when studying human populations (for ethical reasons) or cattle (process prohibitively long and potentially not viable). Instead, the statistical models have been adjusted to accommodate the use of outbred populations for which a pedigree is available (Höschele, 2008). The pedigree information is used to estimate the patterns of allelic inheritance and recombination rates between measured individuals. The use of outbred populations also alleviates to some extent the problem of generalisation of the results, since more alleles are considered.

### Genome-wide association studies

QTL mapping methods leverage recent recombination events in closely related individuals obtained from designed crosses or families. Complementarily, association mapping relies on ancient recombination history over a large number of generations observed across unrelated individuals selected from a natural population of interest (Xu et al., 2017). In this case, the relationship between measured individuals is generally unknown. Due to the high number of recombination events throughout the population history, the extent of linkage disequilibrium is greatly reduced. This permits a more precise mapping of QTLs, as we can assume them linked to very closely located markers only. This advantage over QTL mapping is balanced by a loss of power, as more markers are required in order to detect QTLs, and QTLs with small effects might be missed. Association studies that test for the association of markers measured throughout the entire genome with a trait of interest are termed genome-wide association studies or GWAS (Bush & Moore, 2012).

GWAS studies suffer from a number of drawbacks (Tam et al., 2019). A major one is the impact of population structure amongst the individuals (Bazakos et al., 2017). This refers to relatedness arising from common ancestry of the individuals to different subpopulations within the population studied. Population structure may cause spurious associations between the phenotype and unrelated loci. This could be due to a difference in allelic frequency of the variants between, coupled with a difference in trait distribution between the subpopulations. Therefore population structure must be accounted for in GWAS models. This is done by extracting information about population structure from the genotypic data itself and by including it into the statistical model as a covariate. Population structure can be estimated using dimension reduction techniques such as PCA or DAPC (Jombart et al., 2010), that estimate the major axes of variation amongst the samples. Specialised tools such as the popular algorithm STRUCTURE (Pritchard et al., 2000) have been developed in order to assess the membership of each individual to different subpopulations, based on their genotype. STRUCTURE relies on a Bayesian framework with different models of individual's ancestry.

In consequence, a typical GWAS model consists of a general linear model as follows (see e.g. Aranzana et al., 2005):

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{S}\boldsymbol{\tau} + \mathbf{Q}\mathbf{v} + \epsilon \tag{1.1}$$

In this model, $\mathbf{y}$ is a $n \times 1$ vector of phenotypic values measured for $n$ individuals. The effect of $p$ covariates (such as environmental conditions) on the phenotype are modelled as fixed effects through the $p \times 1$ vector $\boldsymbol{\beta}$, with $\mathbf{X}$ the $n \times p$ covariance incidence matrix, which indicates the value of the different covariates for each observation. The effect of population structure is modelled separately with $q \times 1$ vector of fixed effects $\mathbf{v}$, where the $n \times q$ subpopulation incidence matrix $\mathbf{Q}$ indicates the membership of the individuals to each of the $q$ subpopulations. The $s \times 1$ vector of fixed effects $\boldsymbol{\tau}$

models the effect of the SNPs on the phenotype. The number $s$ of parameters depends on the model chosen to represent the effect of the SNPs (e.g. additive, dominant, etc). The $n \times s$ incidence matrix $\mathbf{S}$ maps the individuals to their genotype for the SNPs. Finally, the $n \times 1$ vector $\epsilon$ is the vector of residual effects, with $Var(\epsilon) = \mathbf{I}\sigma_r^2$ where $\sigma_r^2$ corresponds to the residual variance. This type of model is sometimes referred to as the Q model (Rosyara et al., 2016). The significance of the SNP effects can be assessed with a F-test or likelihood ratio test. It must be noted however that due to the large number of SNPs assessed, the resulting p-values must be corrected for multiple testing, which can significantly reduce the recall of the method.

A number of genetic models can be employed to model the effect of a given genomic variant on the phenotype (Bush & Moore, 2012). Typical GWAS studies focus on bi-allelic markers, i.e. loci for which two alleles only exist. We can denote these two alleles as $A$ and $B$. For a diploid organism (i.e. with two copies of each chromosome), possible genotypes for this marker are: $AA$, $AB$ and $BB$. One can choose to model the impact of each allele on the phenotype (allelic model), or to model the impact of each genotype (genotypic model). In the first case, the fixed effect vector will thus include a parameter for the effect of $A$ and one for the effect of $B$. On the other hand, a genotypic model considers the effect of each of the three possible genotypes. In such case, different scenarios can be considered. An additive model assumes that the effect of the SNP on the phenotype is linear with the number of alleles say $A$ in the genotype. In a dominant model, the presence of one copy of $A$ is enough to impact the phenotype; thus, the genotypes $AB$ and $AA$ have the same impact on the trait. On the contrary, for a recessive model, two copies of $A$ are needed to impact the trait, thus $AB$ and $BB$ are equivalent. Regardless of the genetic model used, the convention is to represent genotypic data for bi-allelic markers by their dosage, i.e. the number of alternate alleles present. In our example, if we consider $B$ as the alternate allele, then the dosage of genotypes $AA$, $AB$ and $BB$ is 0, 1 and 2 respectively.

Yu et al. (2006) showed that, depending on the population studied, correcting for population structure only was not sufficient. Indeed, the matrix $\mathbf{Q}$ reflects relationships between individuals arising from distant population effects. But in some cases, individuals are also related via more subtle and recent familial trends that are not reflected in the population structure. When ignored, this additional relatedness leads to bias in the GWAS results. Therefore, they proposed to account for these effects by integrating a random effect in the GWAS model. The relatedness between individuals due to recent history is captured in the kinship matrix denoted $\mathbf{K}$. A number of methods for estimating $\mathbf{K}$ has been proposed (Hardy & Vekemans, 2002; Oliehoek et al., 2006; VanRaden, 2008; Yu et al., 2006; Zhao et al., 2007). Notably, it has been argued that estimating $\mathbf{K}$ from genotypic data directly was better than relying on pedigree information, as the latter can be subject of errors or missing data (Zhang et al., 2010). The new mixed linear model, commonly named Q + K model, can be written as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{S}\boldsymbol{\tau} + \mathbf{Q}\mathbf{v} + \mathbf{Z}\mathbf{u} + \epsilon \tag{1.2}$$

The terms $\mathbf{y}$, $\boldsymbol{\beta}$, $\boldsymbol{\tau}$, $\boldsymbol{v}$, $\mathbf{X}$, $\mathbf{S}$, $\mathbf{Q}$, $\epsilon$ are identical to those in equation (1.1). In addition, the $n \times 1$ vector of random effects $\mathbf{u}$ is referred to as the vector of polygenic effects, with $n \times n$ incidence matrix $\mathbf{Z}$ (with $\mathbf{Z} = \mathbf{I}$ in the absence of replicates), and Var$(\mathbf{u}) = \sigma_g^2\mathbf{K}$. $\sigma^2$ corresponds to the genetic variance. Several methods and approximations have been proposed to estimate the model parameters (Kang et al., 2008; Listgarten et al., 2012; Yu et al., 2006; Zhang et al., 2010; Zhou & Stephens, 2012).

### Molecular QTLs

As the measurements of mRNA, protein or metabolites levels have become routine, the concepts of QTL mapping and association study have been extended to exploring the genetic variations associated with these molecular phenotypes. Markers found associated with the expression of a given gene are referred to as eQTLs (for expression QTLs) (Veyrieras et al., 2008). Similarly, pQTLs denote genetic variants associated with protein levels; mQTLs refers to variants associated with metabolites levels, and meQTLs to variants associated with methylation (modifications of the DNA via addition of a methyl group, used to repress gene expression on the long term). This however poses the challenge of working with hundreds or thousands of phenotypes, which much be accounted for when computing p-values to assess the statistical significance of such associations. Despite this, eQTLs offer an interesting insight into the mechanisms by which genetic variation affects the expression of genes. Indeed, they can be classified into cis-eQTLs, which are variants located near their associated gene, and trans-eQTLs, which are located away from the corresponding gene (Albert & Kruglyak, 2015; Gilad et al., 2008). Cis-eQTLs are assumed to impact the regulatory regions directly around the gene. On the other hand, trans-eQTLs affect the expression of the corresponding gene through changes that impact a regulator of the gene. For example, genetic variation that affects the expression of a transcription factor also impacts indirectly the expression of its targets. However, eQTL studies often detect more cis-eQTLs than trans-eQTLs as the impact of the latter category on the concerned gene expression is smaller and thus harder to detect (Pai et al., 2015).

### Challenges for polyploids

Initial efforts in bridging genotype and phenotype focused on diploid model organisms, whereas tools specialised to handle polyploid data have only started to be developed recently. This gap in development arises from the higher genetic complexity of polyploid organisms that complicate each step of any genetic analysis (Li et al., 2012). One example is the complications in reconstructing linkage maps from dosage data, due to additional molecular mechanisms (such as double reduction) that must be taken into account. Diploid tools can be used as an approximation by focusing on specific markers whose patterns of inheritance are similar to those of diploids, but this approach does not takes advantage of the full genotypic information. Similarly, genotypic data from allopolyploids can be simplified in order to use diploid tools with reasonable results, as their patterns of inheritance are similar. On the other hand, autopolyploids require dedicated algorithms to account for their polysomic inheritance.

In addition, one of the reasons preventing the use of diploid tools for analysis of polyploid data lies in the way genotypic data is encoded. In particular, considering a bi-allelic marker with alleles $A$ and $B$, the possible genotypes observed for a tetraploid individual (i.e. carrying four copies of each chromosome) are $AAAA$ (nulliplex), $AAAB$ (simplex), $AABB$ (duplex), $ABBB$ (triplex) and $BBBB$ (quadruplex). The corresponding dosages are 0, 1, 2, 3 and 4, respectively. Such data is not compatible with the genetic models used for markers with dosage between 0 and 2, in terms of data encoding as well as genetic model used: more scenarios (other than additive or dominant and recessive models) are possible. Therefore GWAS models must be extended for polyploid dosage.

In a recent review, Bourke et al. (2018) listed tools available to perform genomics analysis, and in particular QTL mapping and GWAS, for polyploid organisms. They pointed out the relatively small pool of software currently available, but highlight the increased developments in the past decade. To the best of my knowledge, there are currently only two pieces of software that allow to perform GWAS in a polyploid setting. One is the R package GWASpoly (Rosyara et al., 2016), which offers an implementation of the linear mixed model presented in Equation (1.2). It handles the computation of the $\mathbf{K}$ matrix, and offers five different genetic models that propose alternate explanations of the impact of SNPs on the quantitative trait. The other available tool is the web application SHEsisPlus (Shen et al., 2016). It is to note however that no details of the statistical model(s) used are provided. In particular, there is no mention of whether the model accounts for population structure and individual relatedness.

### 1.3.2 Reconstructing regulatory networks from observational data

In parallel to association studies, the question of deciphering molecular regulatory interactions mediating the association between genotype and phenotype has been actively pursued. By measuring the concentration or levels of molecular players within cells, we can gain an understanding of how they affect each other. It is possible to design experiments that will allow to answer this question for specific conditions, e.g. time-course measurements following a controlled perturbation (Sima et al., 2009), or molecular intervention targeting a specific molecule (e.g. gene knock-out), in order to observe the effects of the perturbation on other molecules (Gross et al., 2019). However, observational data, in which the concentration of molecular features are measured across different individuals, also contains information that can be used to reconstruct regulatory networks. This approach capitalises on the genetic variations between individuals that amounts to small random perturbations (Rockman, 2008), allowing us to assess how the different molecular concentrations co-vary in response to these small perturbations. A number of statistical tools have been consequently developed to extract molecular networks from observational data. Typically, they have been dedicated to the analysis of transcriptomics data, i.e. the measurement of genome-wide RNA levels, in order to reconstruct gene regulatory networks (GRNs).

### Network inference

A number of reviews of existing network inference methods are available, e.g. Li et al. (2015), Y. X. R. Wang & Huang (2014), Yan et al. (2017) or Van den Broeck et al. (2020). I focus here on tools developed to infer regulatory networks from transcriptomics data. However such tools could also be applied to other datasets measuring the levels of proteins or metabolites as well, to reconstruct protein or metabolites interaction network. One of the simplest approach to network inference relies on assessing the correlation between the genes' expression profiles. This approach is used notably by the popular algorithm WGCNA (Langfelder & Horvath, 2008). Correlation methods reconstruct association networks, in which an edge between two genes can arise from a regulator-target relationship, co-regulation, or membership to a similar biological process or pathway. More complex network inference methods have thus been implemented to uncover regulatory relationships amongst the genes. For example, the algorithm ARACNe (Margolin et al., 2006) is based on mutual information and the data processing inequality concept to remove edges arising from indirect interactions in the reconstructed networks. The concept of mutual information is also used by other network inference methods such as RELNET (Butte & Kohane, 2000), CLR (Faith et al., 2007) or MRNET (Meyer et al., 2007). Another popular method, GENIE3 (Huynh-Thu et al., 2010), decomposes the network inference problem into individual regression problems for each gene, that aim at recovering the set of regulators of the genes. It makes use of random forests to obtain a ranking of regulatory regulations between the genes. Other approaches combine linear regression (e.g. the TIGRESS algorithm, which requires information about which genes act as transcription factors – Haury et al., 2012) or differential equation modelling (Bonneau et al., 2006) and feature selection to reconstruct regulatory networks. Such methods employ a regularisation scheme to ensure the sparsity of the reconstructed graphs, such as LASSO (Tibshirani, 1996) or ridge regression (Hoerl & Kennard, 1970).

Interestingly, it has been shown that no one method performs better than the others in reconstructing GRNs across different biological scenarios, but combining the results from different methods provide better results (De Smet & Marchal, 2010; Marbach et al., 2012). However, gene regulatory networks reconstructed with these methods still lack information about the directionality of the regulation. To overcome this, Bayesian networks have been employed to reconstruct directed GRNs from transcriptomics data (Friedman et al., 2000). In a Bayesian setting, genes or gene products are represented as nodes in a directed acyclic graph (DAG). They are considered as random variables whose conditional distribution depends only on their parent nodes. The topology of the graph is learned by assessing the fit of different candidate graphs using a scoring criterion, which quantifies the fit of the candidate model to the data. The resulting graphs can provide information about causal relationships between pairs of genes. It is to be noted however that not all Bayesian networks can be interpreted as causal graphs.

Approaches to reconstructing biological networks, i.e. by focusing on inferring causal relationships rather than statistical associations in a classical sense, is the object of this thesis. Tools developed to learn causal relationships amongst a set of variables given observational data are detailed in the next section. However, the use of causal inference methodology in reconstructing molecular networks is not widespread. One of the reason is the dimension of biological datasets, which can exceed hundred of thousands of variables, rendering the task of causal structure learning very difficult. In addition, causal structure learning relies on a set of strong assumptions, which are often violated in biological systems, due to the presence of feedback loops, unobserved environmental confounders, etc. Nevertheless, adapting and applying these tools to biological datasets is a necessary step in order to gain a better understanding of biological systems. Indeed, information about causal relationships will help us decipher the flow of information from genotype to phenotype. Moreover, such knowledge can be used to predict the effect of interventions on specific targets (Shpitser & Pearl, 2006), and can help build precise models of biological systems. This could be of immense help to the plant and animal breeding community in answering questions such as the genes on which to focus the selection process in order to optimise a certain trait of interest, or to the biomedical community in the context of diseases to develop targets for a new vaccine or drug.

### Multi-omics data integration

Recently, improvements in technology made possible the cell-wide monitoring of different omics layers, e.g. the protein content (proteomics) or metabolite levels (metabolomics) of cells. Therefore, focus has shifted from analysing a single omics layer to integrating different omics datasets. Indeed, molecular networks reconstructed from a single omics layer, typically transcriptomics, only provide a partial view of the biological systems at play. For example, by using RNA levels as a proxy for gene expression, we ignore all the biological complexity arising from post-transcriptional regulation. Therefore, statistical tools are needed to integrate measurements of different omics molecules. Here, I focus on the case in which several omics measurements are obtained for a same set of individuals or samples – which cannot always be done (Hasin et al., 2017). This is akin to measuring different predictors in the same set of observations.

A number of statistical tools have been developed to integrate omics datasets and extract information that would otherwise be missed in single-omics analysis. Advances in this topic have been extensively reviewed (Eicher et al., 2020; Hawe et al., 2019; Meng, Zeleznik, et al., 2016; Misra et al., 2019; Zeng & Lumley, 2018), with some reviews focusing on specific fields such as human diseases

(Subramanian et al., 2020; Yan et al., 2017) or plant biology (Jamil et al., 2020), or on the integration of specific data types (Cavill et al., 2016). Existing algorithms make use of diverse statistical concepts (see Zeng & Lumley, 2018; Subramanian et al., 2020 for statistical-oriented reviews) and serve different objectives (Eicher et al., 2020; Subramanian et al., 2020). For example, several tools aim at clustering observations (samples, patients), e.g. similarity network fusion (SNF – B. Wang et al., 2014), iCluster (Shen et al., 2009), moCluster (Meng, Helm, et al., 2016), and multiple dataset integration (MDI – Kirk et al., 2012). This is especially useful for the analysis of disease-related data (e.g. cancer), in order to detect different disease subtypes or patients with similar outcomes. Other tools focus on clustering multi-omics features, such as LemonTree (Bonnet et al., 2015) or methods based on self-organised maps (SOM – e.g. Fatima & Rueda, 2020). In this case, the focus is on detecting groups of molecular features across the datasets involved in similar biological processes. A number of methods have been developed to preform dimension reduction, like nonnegative matrix factorisation (NMF – Yang & Michailidis, 2015), multiple factor analysis (MFA – Tayrac et al., 2009), multi-block PCA (Hassani et al., 2013), multiple co-inertia analysis (MCIA – Meng et al., 2014); and/or feature selection, e.g. DIABLO (Singh et al., 2016), sparse generalised canonical correlation analysis (sGCCA – Cai & Huo, 2020), multi-omics factor analysis (MOFA – Argelaguet et al., 2018), orthogonal projection to latent structure discriminant analysis (OPLS-DA – Bylesjö et al., 2006) or OnPLS (Löfstedt & Trygg, 2011). A number of these tools have very recently been evaluated on simulated and cancer datasets (Cantini et al., 2021). In the field of human diseases, methods have been developed for outcome prediction (Kim et al., 2015; Yang et al., 2020). Note that tools often don't fall exactly into one category; rather, they can serve more than one purpose, such as dimension reduction and clustering or feature selection and outcome prediction.

In the context of genotype-phenotype interactions, a few tools have been offered to reconstruct (undirected) molecular networks that link several omics layers. For example, Acharjee et al. (2016) used a combination of Random Forest regression and molecular QTL to select features from transcriptomics, proteomics and metabolomics datasets associated with a number of tuber quality traits in potato. They then used regularised partial correlations to reconstruct for each measured trait an undirected Gaussian graphical model including transcripts, proteins, metabolites and the phenotype of interest. Similarly, in the context of human data, Zierer et al. (2016) reconstructed a mixed graphical model from epigenomics, transcriptomics, glycomics, metabolomics and phenotypic data using the Graphical Random Forest algorithm (Fellinghauer et al., 2013) in the context of age-associated disease comorbidities. Recently, Shi et al. (2019) offered a new tool called sparse multiple canonical correlation network analysis (SmCCNet). SmCCNet uses the concept of canonical correlation analysis with a sparsity constraint to compute vectors of feature weights for each omics dataset that maximise the correlation between the datasets and the correlation between each dataset and the phenotype. Next, based on the sparse canonical weights attributed to each of the omics features, a similarity matrix is computed and a hierarchical clustering procedure is used to reconstruct a set of association networks between the selected features. While such networks provide invaluable information about the relationships between the different omics, they do not allow to assess the flow of causality between them.

In this thesis, I am interested in applying the concept of causal inference to reconstructing molecular regulatory networks, both within a single omics data type, or across several heterogeneous datasets. Therefore, in the next section, I present the concept of causal inference from a statistical perspective, and discuss its applications in the realm of biological datasets.

## 1.4 Causal inference

The concept of causality has been extensively discussed in the literature. Typical experiments to infer the causality between two variables involve interventions on the system to study the impact of changing one variable on the others; however, this is not always possible, and relying on observational data might be the only option in some cases (Glymour et al., 2019), due to ethics concerns, prohibitive cost of experiments, or destructive sampling methods for example. Therefore, reconstructing the causal structure amongst a set of variables for which we only possess observational data has generated a lot of interest. In the following section, I start by reviewing the mathematical notations necessary for handling the concept of causal systems and their representation in the form of graphs. I then expand on the assumptions required to be able to infer causal structures from observational data. This is followed by a review of existing algorithms that aim at uncovering the causal relationships between a set of observed data. Finally, I discuss the application of causal inference in the context of biological systems.

### 1.4.1 Graph terminology

Let $G$ be a graph with $G = (V, E)$, where $V$ is a set of $p$ variables and $E$ is a set of edges $\{(i, j) \mid i, j \in V\}$. A graph is directed if it contains only directed edges in the form $i \rightarrow j$ (Figure 1.2 a)), and undirected if it contains only undirected edges in the form $i - j$. A partially directed graph can contain both directed and undirected edges. The skeleton of a graph $G$ is the undirected graph obtained by rendering all edges in $G$ undirected (Figure 1.2 b)). We say that two variables $i$ and $j$ are adjacent in $G$ if there exists an edge between them. The cardinality of the set $Adj_G(i)$ of nodes adjacent to $i$ in $G$ corresponds to the degree of the node. If the edge $(i, j)$ is undirected, $i$ and $j$ are termed neighbours. If the edge is directed and in the form $j \rightarrow i$, $j$ is a called a parent of $i$, and $i$ is a child of $j$. The set of parents of $i$ in the graph $G$ is denoted as $Pa_G(i)$.

A triplet of variables $i$, $j$, $k$ for which the pairs $i$, $j$ and $j$, $k$ are adjacent but $i$ and $k$ are not adjacent is called an unshielded triple. If the edges are oriented as $i \rightarrow j \leftarrow k$, the triplet is called a v-structure (Figure 1.2 c)). The node $j$ in the v-structure is referred to as a collider. A path $\pi$ in $G$

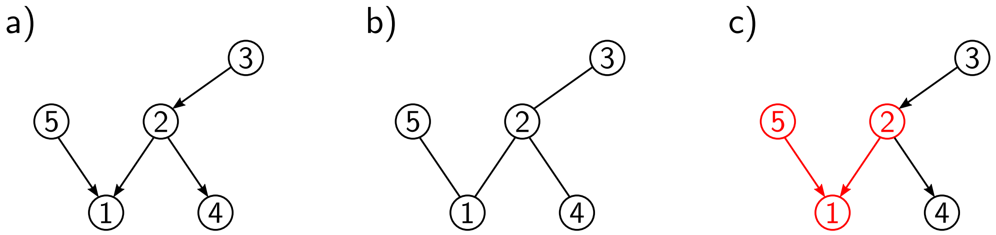


Figure 1.2: a) A Directed Acyclic Graph (DAG) with 5 nodes and b) corresponding skeleton (all edges have been rendered undirected. c) The nodes 1, 2 and 5 and the edges connecting them (in red) form a v-structure.

from $i$ to $j$ is a sequence of nodes $(i, ..., j)$ successively adjacent. $\pi$ is a directed path if all edges are oriented toward $j$. If there exists a directed path $\pi$ from $i$ to $j$ then $j$ is a descendant of $i$, and $i$ is an ancestor of $j$. By convention, a variable $i$ is considered as a descendant of itself, and therefore also an ancestor of itself. A path from $i$ to $i$ including at least one other variable is called a cycle. A directed path from $i$ to $i$ with at least one other node is a directed cycle. A directed graph without any cycles is termed a directed acyclic graph or DAG.

### 1.4.2 Causality – mathematical concepts

Let $\mathbf{X} = (X_1, ..., X_p)$ be a set of variables (discrete or continuous), and $P(\mathbf{X})$ the joint distribution over this set of variables. We say that two variables $X_i$ and $X_j$ are conditionally independent given a third variable $X_k$ if $P(X_i \mid X_j, X_k) = P(X_i \mid X_k)$, and we note $X_i \perp\!\!\!\perp X_j \mid X_k$. We can represent the causal relationships among the variables with a causal graph $G = (V, E)$, where $V$ is a set of vertices or nodes and $E$ a set of edges. In the graph, each node corresponds to a variable (node $i$ corresponding to variable $X_i$) and an edge from node $i$ to node $j$ indicates that $i$ has a direct causal effect on $j$, i.e. any intervention on the value of $X_i$ impacts the distribution of values of $X_j$, in a process that is not mediated by another variable. Note that in a causal graph, any causal effect of a variable $X_i$ on another variable $X_j$ mediated by other variables is termed an indirect causal effect. If variable $X_i$ causally affects $X_j$ both directly and indirectly, the total causal effect corresponds to the sum of the direct and indirect causal effects. A causal graph together with the conditional probability distribution of each node is termed a causal model (Lagnado & Sloman, 2002).

#### Conditional independencies and $d$-separation

I focus here on a specific family of graphs that are the directed acyclic graphs or DAGs. This particular class of graphs is chosen for some of their interesting properties. Indeed, the joint distribution of a set of variables corresponding to nodes in a DAG $G$ can be factorised using the rule of product decomposition:

$$P(X_1, ..., X_p) = \prod_i P\left(X_i \mid X_{Pa_G(i)}\right) \tag{1.3}$$

where $Pa_G(i)$ are the parents of node $i$ in the DAG $G$. This corresponds to the causal (Spirtes et al., 2001) or local (Drton & Maathuis, 2017) Markov condition, stating that in the graph each node is independent of its non-descendants conditionally on its parent nodes. The conditional dependence and (in)dependence relationships between the variables can be read from the graph using the concept of $d$-separation. Let $\pi$ be a path between a pair of variables $i$, $j$, and $S$ be a subset of variables not containing $i$ and $j$. We say that $S$ blocks the path $\pi$ if:

- $\pi$ contains a non-collider that is in $S$. That is, there is a chain $l \rightarrow m \rightarrow k$ or $l \leftarrow m \rightarrow k$ with $m \in S$, or

- $\pi$ contains a collider that has no descendant in $S$. That is, $\pi$ contains a v-structure $l \rightarrow m \leftarrow k$ such that no descendant of $m$ (including $m$) is in $S$.

If $S$ blocks every path from $i$ to $j$, then $i$ and $j$ are $d$-separated by $S$, and we write $i \perp_G j \mid S$ (see Figure 1.3 for an example). This definition can be extended to sets of nodes: the sets $A$ and $B$ are $d$-separated by $S$ if $S$ blocks every path from a node in $A$ to a node in $B$. We say that $\mathbf{X}$ satisfies the global Markov property with respect to the DAG $G$ if all $d$-separation occurrences in the graph imply conditional independencies in the distribution $P$:

$$A \perp_G B \mid C \Rightarrow \mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B \mid \mathbf{X}_C \tag{1.4}$$

with $\mathbf{X}_U = (X_u, u \in U)$. The global Markov property together with its reverse implication form the faithfulness property (Drton & Maathuis, 2017). It states that all conditional independencies in the

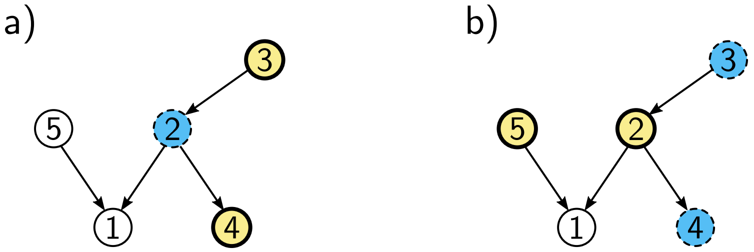


Figure 1.3: Examples of $d$-separation. The two yellow nodes with bold circles are $d$-separated by the set of blue nodes with dashed circles. a) Nodes 3 and 4 are $d$-separated by node 2, as node 2 is a non-collider in the path between 3 and 4. b) Nodes 3 and 4 $d$-separate nodes 2 and 5, as node 1 forms a v-structure with 2 and 5 and node 1 is not a descendant of nodes 3 and 4.

distribution $P$ are exactly the same as those implied by the $d$-separations in the graph:

$$X_A \perp\!\!\!\perp X_B \mid X_C \iff A \perp_G B \mid C \tag{1.5}$$

The DAG $G$ is then called a perfect map of $\mathbf{X}$ (Drton & Maathuis, 2017).

This relationship between conditional independence and $d$-separation provides the basis for causal structure learning. Indeed, if we do not know the causal DAG underlying a set of variables $\mathbf{X}$, but we possess $n$ independent and identically distributed (iid) observations of $\mathbf{X}$, we can use the information about conditional independencies amongst the data to try and reconstruct the causal DAG $G$. This inference relies on the assumption that there exists a causal DAG that is a perfect map of $\mathbf{X}$, i.e. that the Markov and faithfulness properties hold.

This inference problem is complicated by the fact that several distinct DAGs can encode the same set of conditional independencies (or $d$-separation relationships). In this case it is impossible from observational data alone to infer the exact topology of the causal graph. Graphs that entail the same set of conditional independencies are termed Markov equivalent; they share the same skeleton and the same v-structures. The set of all Markov equivalent graphs of a particular DAG is termed a Markov equivalence class, and can be represented by a completed partially directed DAG or CPDAG (see Figure 1.4 for an example). In the CPDAG, a directed edge $i \rightarrow j$ exists if this edge is present with the same orientation ($i \rightarrow j$) in each DAG of the equivalence class. An edge between node i and j is undirected ($i - j$) if the edge is present in one direction ($i \rightarrow j$) in some DAGs of the equivalence class and in the opposite direction ($i \leftarrow j$) in other DAGs. Therefore, causal structure learning usually aim at recovering a causal CPDAG. In some instances a DAG from the equivalent class can be returned, but this is not advisable as the inferred DAG may not be the true causal DAG underlying the data, and the uncertainty about edge orientation is not explicitly expressed.

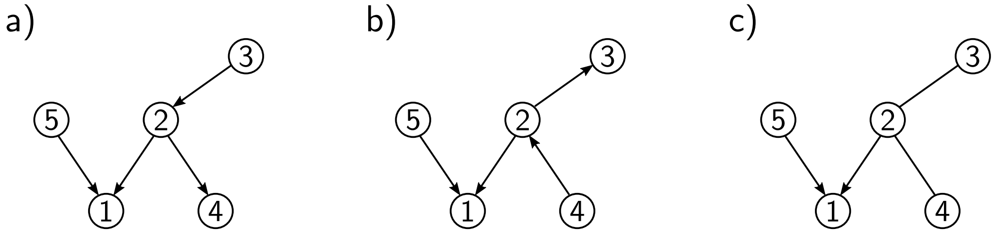


Figure 1.4: a) and b): two Markov equivalent DAGs. They share the same skeleton an v-structure (formed by the nodes 1, 2 and 5). c) The corresponding CPDAG that represents the Markov equivalence class of DAGs a) and b).

### Causal sufficiency

In causal inference, we generally assume that all the variables involved in a causal problem are observed in our data. This corresponds to the assumption of causal sufficiency. A number of existing causal structure learning algorithms rely on this assumption, thus assuming that no unmeasured confounder or common cause are at play in the investigated causal system. However, it can happen that this hypothesis does not hold, i.e. there exists an unmeasured variable that affect two or more observed variables. Such unmeasured variables are termed hidden variables or confounders. In this case, the observed conditional independence among the variables can lead to erroneous graph reconstruction as the causal Markov condition does not hold any more. We say that the space of DAGs is not closed under marginalisation (where marginalisation corresponds to removing some causal variables from the set of observed variables) (Colombo et al., 2012). This means that if there are some hidden variables affecting a set of observed variables there may not exist a DAG representing only the observed variables that is faithful to the distribution, i.e. that represents exactly the set of conditional independencies via $d$-separation. In this case, trying to learn the causal CPDAG between the observed variables would lead to adding additional edges in the graph that do not represent true direct causal effects, but rather are due to the impact of the latent variables.

In this case, it is possible to use a new class of graphs called maximal ancestral graph (MAG) in order to represent the causal relationships amongst the observed variables generated by a DAG with latent variables (Figure 1.5 a) and b)). In this class of graph, the interpretation of the edges differ from the DAGs. Contrary to edges in DAGs that inform about causal relationships, edges in MAGs inform about ancestral relationships (Heinze-Deml et al., 2018). In a MAG, an edge from $i$ to $j$ with a tail at $i$, $i \relbar\!\!\relbar\!\!\ast\, j$ (with the star indicating any type of mark, i.e. an arrow or a tail), indicates that $i$ is an ancestor of $j$ in the corresponding DAG. If the edge has an arrowhead pointing toward $i$, $i \leftarrow\!\!\relbar\!\!\ast\, j$, then

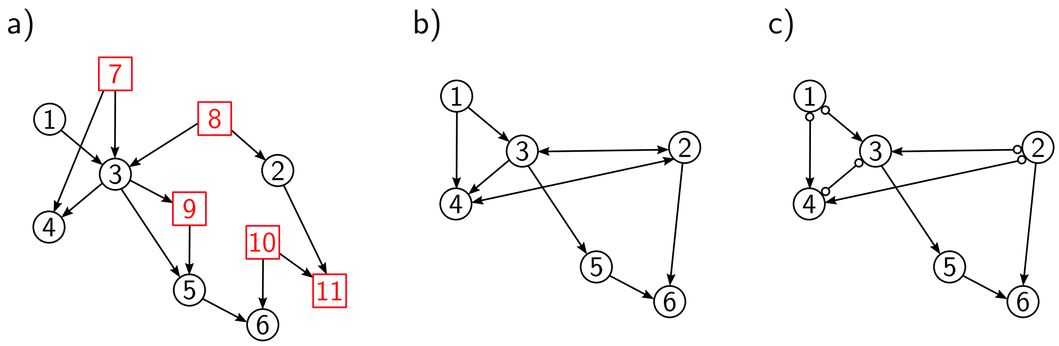


Figure 1.5: a) A DAG with 11 nodes. Nodes 7 to 11 (with red squares) are latent variables, i.e. are not observed. b) Corresponding MAG, representing the ancestral relationships between observed variables. c) Corresponding PAG, representing the Markov equivalence class of the MAG in b). Adapted from Claassen & Heskes (2012).

$i$ is not an ancestor of $j$ in the underlying DAG. A bidirected edge between $i$ and $j$, $i \longleftrightarrow j$, indicates the presence of a confounder affecting the two variables. MAGs also account for the presence of selection variables, which are unmeasured variables influencing the values that we can observe from of one or more variables. The presence of selection variables is represented by undirected edges in the MAG ($i$ —— $j$). The conditional independencies among observed variables can be extracted from the MAG using a generalisation of the $d$-separation principle termed $m$-separation. Similar to DAGs, several MAGs can encode the same set of conditional independencies, thus forming a Markov equivalence class. This equivalence class can be represented by a Partial Ancestral Graph (PAG – Figure 1.5 c)). The uncertainty about the mark at one end of an edge among the different equivalent MAGs is represented in the PAG with a circle; for example $i \circ\!\!\!-\!\!\ast\ j$ indicates that this edge is present with an arrowhead in some MAGs ($i \leftarrow\!\!\ast\ j$) and with a tail in others ($i \longrightarrow\!\!\!\!\ast\ j$). It is always possible to learn the PAG corresponding to a distribution under the faithfulness assumption.

### 1.4.3 Causal learning algorithms

A number of causal structure learning methods have been proposed to infer the topology of causal graphs from observational data. They can be classified into three categories, according to the strategy used for inferring the causal graph. The methods also differ in the assumption they make about the data.

#### Constraint-based algorithms

A first approach for causal structure learning is to learn the structure of the causal graph from the conditional independence information extracted from the data. By assuming that the faithfulness assumption holds, it is possible to assess $d$-separation relationships in the graph from conditional independencies tests. Information about $d$-separation, in turn, informs the addition or removal of edges in the graph. An intuitive constraint-based algorithm, such as the SGS algorithm (Spirtes et al., 2001) (for Spirtes-Glymour-Scheines), typically starts from a complete undirected graph (in which all possible pairs of variables are linked by undirected edges), and proceeds according to the following steps:

1. For each pair of nodes $i, j$, look for a subset of nodes $S$ that $d$-separates them in the data. If such a subset exists, remove the edge between the nodes $i$ and $j$, and record $S$ in $Sepset(i, j)$ and $Sepset(j, i)$.

2. Once all the tests have been performed, orient the v-structures (because v-structures share the same orientation in all Markov equivalent graphs) using the results of the conditional independence tests performed in step 1. More specifically, for a triplet of nodes $i$, $j$ and $k$ such that the pairs $i$, $j$ and $j$, $k$ are each adjacent, but $i$ and $k$ are not adjacent, if $j$ is not in $Sepset(i, k)$, orient $i - j - k$ as $i \rightarrow j \leftarrow k$.

3. Iteratively orient as many edges as possible, using the rule that no additional v-structures or cycles can be created (Spirtes et al., 2001).

The main problem of this approach is the computational cost of testing for $d$-separation. In particular, the SGS algorithm systematically tests every possible subset of variables as conditioning set for each pair of variables. It becomes practically impossible when the number of variables is large. The PC algorithm (Peter-Clark) (Spirtes & Glymour, 1990) alleviates this problem by iteratively testing for conditional dependencies with conditioning sets of increasing size. It starts by removing edges between variables based on zero order conditional independence. It then proceeds to test for conditional independencies with conditioning set of size one for all pairs of nodes still connected, then size two, etc. It stops when it has reached a size that is larger than the number of neighbours of any pair of nodes still connected. The output of the PC algorithm is a CPDAG representing the Markov equivalence class of the inferred causal graph. One of the limitations of the PC algorithm is that the order in which the variables are considered affects the result of the inference. Therefore, Colombo & Maathuis (2014) proposed an alternative version termed the PC-stable algorithm. Other versions of the PC algorithm include the conservative PC (Ramsey et al., 2006), which deals with a different orientation of the v-structures, or the parallel-PC (Le et al., 2019) that makes use of parallel computing.

To circumvent the difficulties in scaling up the PC algorithm for a large number of variables, local learning algorithms have been proposed. Rather than estimating the causal graph linking all observed variables, they seek to reconstruct the Markov Blanket of each variable, which is a minimal set of variables such that, when conditioned on, the variable of interest is independent of the remaining variables. The Markov Blanket of a variable contains its parents, its children and the children of its parent variables in the corresponding causal graph. One such algorithm, the IAMB (for Incremental Association Markov Blanket) was proposed by Tsamardinos et al. (2003a). The IAMB algorithm consists of two phases: in the first phase, for a given variable of interest $X_i$, iteratively include in the potential Markov Blanket set – denoted as $CMB$ – variables $X_j$ that maximise the mutual information score between $X_i$ and $X_j$ conditionally on the current $CMB$ set. In a second time, variables in the $CMB$ set are iteratively removed if they are independent of $X_i$ conditionally on the remaining $CMB$ set, according to the mutual information score. Other methods to estimate the Markov Blanket of a set of variables have been proposed (e.g. Peña et al., 2005; Nilsson et al., 2007).

In presence of unmeasured confounders, i.e. when the assumption of causal faithfulness is violated, the PC algorithm can produce an incorrect graph. Indeed, a hidden variable affecting two observed variables can induce statistical dependencies leading the algorithm to add a causal edge from one to the other (Spirtes et al., 2001). By performing additional independence tests conditioning on subsets of variables that are not adjacent to the considered variables, it is possible to remove such spurious edges. This is the principle of the FCI algorithm (Fast Causal Inference – Spirtes et al., 1999). The

FCI algorithm starts with a complete undirected graph over the set of observed variables and proceeds as follows:

1. Use the principle of the PC algorithm to compute an initial skeleton, i.e. by testing iteratively for increasing $i$-th order conditional independencies to detect $d$-separations in the graph. This step is often referred to as the PC-adjacency search.

2. Orients the v-structures: let $F$ be the resulting graph of the previous step. Orient all edges as $\circ\!\!-\!\!\circ$. For triplets of nodes $i$, $j$, $k$ such that $i$ and $j$ and $j$ and $k$ are adjacent in $F$ but $i$ and $k$ are not adjacent, if $j$ is not in $Sepset(i,k)$, orient $i \ast\!\!-\!\!\circ\, j \circ\!\!-\!\!\ast\, k$ as $i \ast\!\!\rightarrow j \leftarrow\!\!\ast\, k$.

3. Using the orientation, update the skeleton of the graph:

   - Compute for each pair of variables $i, j$ the set of variables $k$ satisfying the following conditions: there is a path $\pi$ between $i$ and $k$ such that for every subpath $< m, l, h >$ of $\pi$, $l$ is a collider on the subpath, or $< m, l, h >$ forms a triangle. This set is denoted by Poss-D-Sep$(i, j)$.
   - For each pair of nodes $i$, $j$, look for a subset $S$ of Poss-D-Sep$(i, j)$ or Poss-D-Sep$(j, i)$ that $d$-separates them. If such a subset exists, remove the edge between the nodes $i$ and $j$, and record $S$ in $Sepset(i, j)$ and $Sepset(j, i)$.

4. Orient all edges as $\circ\!\!-\!\!\circ$ and use the same process as in step 2 to orient the v-structures of the updated skeleton.

5. Use repeatedly a set of orientation rules to orient as many edges as possible. The set of orientation rules to be used was first introduced by Spirtes et al. (1999) and later extended by Zhang (2008).

Note that the output of the PC adjacency search is a superset of the final skeleton of the output graph, i.e. it contains too many edges, which are removed in the subsequent graph refinement step. The FCI algorithm returns a PAG representing the Markov equivalence class of the inferred MAG.

In practice, the FCI algorithm is inapplicable on large sets of variables. In consequence, several variants have been proposed to scale up the algorithm. For example, Colombo et al. (2012) proposed the RFCI algorithm (Really Fast Causal Inference). They removed the second step of graph refinement of the original algorithm, which is the most computationally heavy. Instead, additional conditional tests are performed before orienting the v-structures. This lower computational complexity comes at the cost that the output of RFCI can be less informative than these of FCI, as in some cases it can return too many edges. Moreover, the output graph cannot directly be interpreted as a PAG. Another variant, FCI+, has been proposed by Claassen et al. (2013). They also address the complexity of the second step of FCI by using an alternative method to compute the Poss-D-Sep sets of each pair of variables, directly based on the results of the PC-adjacency search. The FCI algorithm and the presented variants have been deemed too conservative (Frot et al., 2019) and with a poor performance on small sample sizes (Ogarrio et al., 2016).

### Score-based algorithms

The second category of causal discovery algorithms relies on a scoring scheme that evaluates the fit of a given causal network to the data, in order to select from a set of candidate graphs the one best explaining the data. Such algorithms rely on heuristics to explore the space of candidate causal graphs. A first greedy algorithm proposed by Meek (1997) explores the search space of causal DAGs in two phases, starting from an empty graph:

1. Forward phase: iteratively add an edge to the graph, selecting the edge addition that most improves the score, until a local maximum is reached.
2. Backward phase: iteratively remove an edge to the graph, selecting the edge deletion that most improves the score, until a maximum is reached.

It is important that the scoring criterion used to evaluate candidate moves is score equivalent (Chickering, 2003; Nandy et al., 2018), which means that two graphs from a same Markov equivalence graph will be assigned the same score. The score must also be decomposable into the sum of the contributions of each variable given its parents, and consistent, i.e. a graph not entailing the conditional independencies of the true graph will receive a lower score than a graph that does. If two graphs entail (a subset of) the conditional independencies of the true graph, the sparsest model, i.e. the graph with the lowest number of edges will be attributed a higher score. The $l_0$- or $l_1$-penalised log-likelihood score (Nandy et al., 2018) or the Bayesian Information criterion (BIC – see e.g. Chickering, 2003) are commonly used in score-based algorithms.

The GES (Greedy Equivalence Search) version of Chickering (2003) improves upon this first algorithm by applying the greedy search on the search space of equivalence classes (CPDAGs) instead of the search space of all DAGs. In practice, they defined new moves for the greedy search, that consists of a sequence of 1) one or more edge reversal (i.e. switch the orientation of the edge), 2) single edge addition, 3) one or more edge reversal again. An optimised implementation of the GES algorithm, termed FGS or FGES, was proposed by Ramsey et al. (2017). The mathematical concept is identical, but the algorithmic complexity is decreased notably by caching score information and parallelisation of some of the steps. Such score-based algorithms scale in general better than their constraint-based counterparts when the number of variables is large, but can become computationally expensive when the search space is densely connected.

### Hybrid algorithms

Methods have been proposed that combine the constraint- and score-based approaches. For example, the Sparse Candidate algorithm (Friedman et al., 1999), improves upon classical heuristics for Bayesian networks learning (score-based approaches) by restricting the search space with a constraint-based search for candidate parents of the variables. This first search is denoted as the Restrict phase.

In a second step, the Maximise phase, a heuristic is applied to find a Bayesian network that maximises the score while respecting the constraints inferred in the previous step. The output is used to update the set of potential parents for each variables. This cycle is repeated until convergence. This approach is modular, in that different methods can be chosen independently for the restrict and maximise phases. Friedman et al. (1999) chose to use the mutual information score to select the set of potential parents in the first phase, and a greedy hill-climbing algorithm for the second phase. The principal advantage of the method is that the Restrict phase decreases the complexity of the maximise phase by reducing the search space. An obvious drawback of the method is that the first step of the algorithm restricts the size of the set of potential parents of each variable to a constant $k$ to be defined by the user.

A similar framework is used in the MMHC (Max-Min Hill Climbing) algorithm by Tsamardinos et al. (2006). The restrict phase is performed by the MMPC algorithm (Max-Mix Parents and Children – Tsamardinos et al., 2003b), which outputs for each variable the list of its parents and children. More specifically, for a given variable $i$, the MMPC proceeds as follows:

1. Initialise the set of candidate parents and children of $i$, $CPC(i)$, as an empty set.
2. For each other variable $f$, look for the subset of $CPC(i)$ that minimises the association between $i$ and $f$. The association between any two variables is computed from a conditional independence test. Retain the associated minimum score, denoted $assoc_f$.
3. Select the variable $f$ with the highest minimum score $assoc_f$. If the selected $assoc_f$ is non-null, add $f$ to $CPC(i)$.
4. Repeat until no more variables can be added to $CPC(i)$.

The output of this first step is used to construct an undirected graph in which each variable is linked to each of its potential parents and children (variables in $CPC(i)$). A greedy hill-climbing search is then performed to find the graph optimising a scoring criterion. The possible moves of this greedy search include edge addition, edge removal and edge reversal, with the constraint that only edges present in the undirected graph from step 1 can be considered. Nandy et al. (2018) have showed that the MMHC algorithm is inconsistent, i.e. the probability that the inferred graph is the true graph does not tend to one when the number of observations becomes really large. They proposed a related algorithm, ARGES (Adaptive Restricted Greedy Equivalence Search). The algorithm also starts by finding a first graph skeleton by applying either the MMPC algorithm as shown above, or the adaptive LASSO (Zou, 2006), a feature selection algorithm that builds on the LASSO (Tibshirani, 1996). Then, they apply a restricted version of the GES algorithm, in which the possible moves performed in each step of the greedy search are restricted according to the skeleton inferred in the first step and the current graph. This modification restricts the search space of the GES algorithm and allows to prove the consistency of ARGES in certain high dimensional settings.

Tsirlis et al. (2018) extended the concept of MMHC by relaxing the condition of causal sufficiency. They showed that if the assumption of causal sufficiency is violated, the MMPC algorithm returns for a given variable a superset of the true parents and children of the variables. To circumvent this, their $M^3HC$ algorithm (MAG Max-Min Hill Climbing) starts by applying the MMPC algorithm to each variable, as in the MMHC algorithm. A greedy search is then applied, starting from an empty graph. The difference with the MMHC algorithm is that the moves considered during the greedy search allow to search the MAG space, therefore allowing for the presence of confounders. In addition, the scoring criterion used, here the BIC, is a valid criterion for comparing MAGs. In a similar spirit, Frot et al. (2019) extend the GES-based hybrid algorithm to the case of latent variables. However, contrary to the $M^3HC$ algorithm, they restrict themselves to cases where a few unobserved variables impact many of the observed variables, and in which the underlying DAG among the observed variables is sparse. They apply the low-rank plus sparse algorithm (Chandrasekaran et al., 2012) to the data to estimate the inverse covariance matrix between the observed variables conditional on the hidden variables. The low-rank plus sparse matrix estimates a sparse graphical model between the observed variables conditionally on the latent variables, under the hypothesis that the number of latent variables is smaller than the number of observed variables. If the assumption holds, then the matrix summarising the effect of the latent variables has a low rank. This covariance matrix is then used for the GES algorithm, allowing to reconstruct the causal graph among the observed variables after correcting for the effect of the confounders.

All the above algorithms use the strategy of reducing the search space for score-based methods by applying in a first step a constraint-based algorithm. Ogarrio et al. (2016) take the reversed approach. Their goal is to combine GES, asymptotically correct but which requires the assumption of no hidden confounders, with FCI, that allow for hidden variables but performs poorly on small samples. The optimised version of GES, FGS, is applied in a first step to obtain a CPDAG. In the presence of hidden variables, the skeleton of the output CPDAG is a hypergraph of the correct causal skeleton, and the orientations of the edges are potentially incorrect. The FCI algorithm is then applied to the skeleton of the output CPDAG. This phase is identical to the FCI algorithm with the exception of the v-structure orientation steps that are performed using information about the output of the GES phase.

#### Additional assumptions to break the Markov equivalence

Recently, a number of models have been proposed in order to distinguish between several DAGs from a same Markov Equivalence class (Glymour et al., 2019; Zhang et al., 2018). Such models have to make additional assumptions on the distribution of the observed data, as the conditional independencies alone cannot separate two Markov equivalent DAGs. They form a subgroup of the class of Functional Causal Models, i.e, models that model a response variable $Y$ as a deterministic function of its cause(s) $X$ and an unmeasured noise or disturbance term $\epsilon$:

$$Y = f(X, \epsilon)$$

In particular, Shimizu et al. (2006) makes the assumption that the response depends linearly on its cause, and the noise term is non-Gaussian:

$$Y = bX + \epsilon$$

This model is called a Linear Non-Gaussian Acyclic Model (LiNGAM). If two variables follow such a generative model, it becomes possible to test the directionality of the causal relationship by assessing the independence of the error terms with each of the two variables. Indeed, the residuals of a regression of $Y$ on $X$ will be independent of $X$, while the residuals of a regression of $X$ on $Y$ will not be independent from $Y$. Other similar models have been proposed, such as the nonlinear additive model (NAM – Hoyer et al., 2008):

$$Y = f(X) + \epsilon$$

where $f$ is a non-linear function, or the post-nonlinear model (PNL – Zhang & Chan, 2006), which assumes:

$$Y = f_2(f_1(X) + \epsilon)$$

These methods are mentioned for the sake of completeness, but are outside the scope of this thesis, as they rely on a deterministic representation of the variables, and due to the additional assumptions they make about the distribution of the data.

### 1.4.4 Structural Equation Modelling

The algorithms presented above rely on probabilistic graph models to model causal relationships amongst a set of variables. An alternative approach is to use Structural Equation Modelling (SEM – Hox & Bechger, 1998; Schumacker & Lomax, 2016). Structural equation models also use a graph to represent the causal relationships between the variables. Variables whose values depend on other variables are termed endogenous variables. Variables whose values do not depend on other variables are termed exogenous variables. The difference between Structural Equation Modelling and methods presented above, that rely on probabilistic graphs, lies in the mathematical model used to encode the causal relationships between variables. In a structural equation model, structural linear equations depict each variable as a linear combination of the values of its parent nodes, with the addition of an error term called disturbance. The error term is sometimes considered as an exogenous variable that we do not observe. This gives, in a matrix notation:

$$\mathbf{Y} = \mathbf{B}\mathbf{Y} + \boldsymbol{\Gamma}\mathbf{X} + \boldsymbol{\zeta}$$

where $\mathbf{X}$ and $\mathbf{Y}$ represent the set of exogenous and endogenous variables, respectively, and $\boldsymbol{\zeta}$ corresponds to the disturbance term. $\boldsymbol{\Gamma}$ is the matrix of fixed effects of the covariates, and $\mathbf{B}$ is the matrix of structural coefficients that represent the effects of a parent node on its children. For a given causal graph, it is possible to estimate the parameters of the models and assess the fit of the model to the data, i.e. with a maximum likelihood approach. Therefore, structural equation models are generally used to test the validity of a causal model defined *a priori*, rather than performing causal structure learning.

### 1.4.5 Causal inference in biological systems

The concept of causal inference has been applied to biological systems as early as the 2000's, although its use in this context has been far from wide-spread. Studies aiming at linking genotype and phenotype have made use of the unidirectionality of genotype-to-phenotype relationships. Indeed, the genomic make-up of individuals inform their phenotypes, including molecular phenotypes such as gene expression. On the other hand, a phenotypic trait or the expression of a gene does not influence the individual's genomic composition. Therefore, genomic variants can be seen as causal anchors, i.e. source variables from which causality flows (unless if one considers evolutionary feedback, which is outside the scope of this thesis). This concept has been used in two ways: Mendelian randomisation and causal inference tests for triplets of variables.

Mendelian randomisation makes the parallel between the random allocation of alleles during gamete formation and the allocation of a treatment to each individual at random in a randomised controlled trial (Davey Smith & Ebrahim, 2003). Therefore, genetic variations between individuals in observational studies can be considered as treatments allocated at random. Because of this, genetic variants can be used as instrumental variables to test for the causal relationships between a risk factor (say, the expression of a gene) and an outcome (e.g. a phenotypic trait of interest) – see for example Burgess et al. (2017). The use of the instrumental variable allows to test for the effect of an unobserved confounder on the relationships between risk factor and outcome. This concept has been applied to epidemiology, but it has also been used to study the directionality of causal relationships between pairs of (molecular) phenotypes. For example, Aten et al. (2008) proposed the NEO software, which relies on non-common markers for a pair of traits in order to score the possible causal orientations between them. This information can then be used to reconstruct a directed network between a set of traits. Interestingly, Mendelian randomisation has also been used to link different omics layers. For example, Shin et al. (2014) and Bartel et al. (2015) used eQTLs and mQTLs as instrumental variables to reconstruct the causal relationships between transcripts and metabolites in human blood. Recently, Qiu et al. (2020) used a combination of molecular QTL mapping, multi-omics integration tools and Mendelian randomization to identify biomarkers of osteoporosis.

In a similar fashion, a number of studies have made use of co-localised genetic markers associated with pairs of traits, phenotypic traits, or levels of a molecule (e.g. mRNA or protein). The co-localisation of markers associated with each of two traits $T_1$ and $T_2$ (phenotypes or molecular levels) indicates a common genotypic cause. One can then test the causal structure linking the variant and the traits. This is made possible by the fact that only a number of causal scenarios are possible, since no causal edge can target the genetic variant. For example, the genetic variant can control the trait $T_2$ only through its impact on trait $T_1$ (causal model); conversely, $T_2$ can be the mediator of the effect of the variant on $T_1$ (reactive model). In another scenario, the association of the variant with the traits can arise from independent processes (independent model) (Schadt et al., 2005). Schadt et al. (2005) derived a likelihood-based causality model selection (LCMS) to distinguish between these three causal scenarios by comparing the goodness of fit of each model. Later, Lin Chen et al. (2007) proposed the Trigger algorithm, based on a causality equivalence theorem which assess the probability of the causal model $L \rightarrow T_1 \rightarrow T_2$ between a given locus $L$ and a pair of trait $T_1$ and $T_2$. The causality equivalence theorem rules that a causal relationship in the form of the causal model exists if the variables satisfy: (i) $L \rightarrow T_i$, (ii) $L \rightarrow T_j | L \rightarrow T_i$; and (iii) $L \perp\!\!\!\perp T_j | T_i$. All three conditions are tested using appropriate likelihood ratio tests. Resulting orientations between genotype-traits triplets are assembled to obtain a directed transcription regulatory network. In the same vein, Millstein et al. (2009) developed a Causal Inference Test to test the causal model between a genetic variant and a pair of phenotypes, by fitting and comparing linear regressions. More recently, the FINDER algorithm (Wang & Michoel, 2017) has been used to reconstruct GRNs using genetic variants as causal anchors. Alternatively, genomic variants have been used in parallel with Structural Equations modelling to reconstruct GRNs (Liu et al., 2008) or phenotype networks (Li et al., 2006; Peñagaricano et al., 2015b).

Both Mendelian randomisation-based approaches and causal inference tests between triplets of variables have been compared in the context of molecular networks in recent studies (Ainsworth et al., 2017; Auerbach et al., 2018). However, these approaches are limited by the fact that they focus on triplets of variables independently (a genetic variant and two traits, either molecular measurements of phenotypes). A few other studies have applied genomic variants to causal inference in a more general setting. For example, Zhu et al. (2004) used common eQTLs to assess the set of potential parents for each gene from an expression dataset. This information was used as prior knowledge for the reconstruction of a Bayesian network between the transcripts. Later, they added transcription factor binding sites as a prior to help in the reconstruction of the network (Zhu et al., 2008). Furthermore, they also used this approach to reconstruct a directed network comprising both transcripts and metabolites (Zhu et al., 2012). Li et al. (2006) and Liu et al. (2008) also used results from QTL mapping to help in reconstructing Structural Equation models for a set of genes. Nevertheless, the application of causal learning algorithms for the reconstruction of regulatory networks using observational studies has been limited (Glymour et al., 2019). One example of such application is by Neto et al. (2008), which combines the PC algorithm with a generalisation of the LCMS test to reconstruct a causal GRN.

Peñagaricano et al. (2015a) uses the IAMB algorithm (Tsamardinos et al., 2003a), a local structure learning algorithm, to reconstruct directed gene-phenotype networks. Montastier et al. (2015) used Mixed Graphical Models and canonical correlation analysis to reconstruct causal networks between mRNA levels, fatty acids and a number of bio-clinical traits related to weight change. Finally, it is to note that Sedgewick et al. (2019) developed a web-based algorithm CausalMGM to reconstruct directed graphs linking both continuous and discrete variables. This is a promising tool for the reconstruction of causal networks for across different omics layers.

Moreover, while a number of causal inference algorithms presented in this section have been evaluated and compared in recent benchmarking studies (Constantinou et al., 2020; Heinze-Deml et al., 2018), there is a lack of assessment of such tools in the context of reconstructing molecular networks. This is an important point, as biological measurements are typically noisy and susceptible to violate a number of assumptions necessary to causal inference. It is therefore important to assess whether such tools can be applied to the reconstruction of biological processes, and to pinpoint possible areas of improvements. This will pave the way for the development of improved algorithms more suited to the unique nature of biological datasets.

## 1.5 Simulation of Gene Regulatory Networks

When developing statistical tools and algorithms to extract information from biological data, it is crucial to assess their performance. This allows to evaluate the methods' strengths and weaknesses, to compare it to other methods and to highlight potential areas of improvement. I focus here on the specific problem of inferring regulatory networks from biological measurements (either association or causal networks), typically transcriptomics data. A first way to do so is to apply the given tool to an existing experimental dataset, and compare the results of the inference to known regulations and pathways (e.g. Simoes et al., 2013). However, such an approach presents a number of drawbacks. First, the comparison is limited to our existing knowledge about the biological system at play which can be biased, incomplete or even erroneous. Also, additional experimental measurements needed to validate new results can be very expensive and time-consuming, thus hindering the capacity to assess the potential for new discoveries. Lastly, existing biological gold-standards are biased towards well-known model organisms and might not be representative of all biological datasets on which the tool will be applied.

Therefore, a second approach to assess the performance of network inference algorithms is to apply them to simulated datasets for which the underlying generating regulatory network is known (for example Mendes et al., 2003). Simulated datasets offers several advantages. First, because the model used to generate the data is known, it is possible to objectively compare the inference results to the true network, and thus produce performance metrics such as precision or recall. One can also generate several datasets by changing some aspect of the generating model, and compare the changes in performance of the tool across these different simulation scenarios (e.g. inclusion of technical noise). Also, one has complete control over the model used to simulate the data, and can therefore tune different aspects of the simulation, which can be used for example to represent non-model organisms, or include technical noise. Of course, simulations are a crude, partial, simplified, idealised representations of biological systems and are also biased by our incomplete knowledge about those systems. Nevertheless, simulated datasets have been widely used to evaluate and compare gene regulatory networks, independently or through competitions such as the DREAM challenges.

There are many aspects of designing a simulator for assessing the performance of network inference methods. In the following paragraphs, I review the main aspects of constructing a simulator, and discuss existing tools to generate simulated gene expression datasets. A list of existing simulators that simulate gene expression for the purpose of evaluating network inference algorithms is presented in Supplementary Table A.1.

### 1.5.1 Topology of the simulated GRNs

As our knowledge about regulation between gene products increased and was summarised into GRNs, it has been found that these networks exhibit a number of properties. For example, most regulatory genes only control a small number of target genes, while a few genes termed hubs regulate a large number of targets. Consequently, the distribution of in- and out-degree of the genes (i.e. the number of incoming and outgoing edges from the genes) can be described by a power-law distribution (Albert & Barabási, 2000; Barabási & Albert, 1999; Emmert-Streib et al., 2014; Neal et al., 2021; Ouma et al., 2018), or in some cases by a exponential distribution (Balaji et al., 2006; Guelzim et al., 2002). Networks obeying such property are termed scale-free networks. Also, biological networks have a tendency to form groups of highly interconnected nodes or modules (Wagner & Fell, 2001; Watts & Strogatz, 1998). Genes from one module are less likely to interact with other genes from a different modules. Moreover, the hub genes mentioned previously form the link between different modules (Ravasz et al., 2002). These two characteristics are referred to as the modularity and hierarchical organisation properties. Biological networks are also small-world networks, meaning that any component in the network can easily be reached from any other component (Jeong et al., 2000).

Different approaches have been employed to account for these properties when simulating gene expression regulation. Simulators such as the ones implemented in (Di Camillo et al., 2009; Mendes et al., 2003; Pinna et al., 2011; Roy et al., 2008) use algorithms that generate networks with some of these properties. For example, Di Camillo et al. (2009) proposed an algorithm to generate hierarchical and modular networks, and used it in their simulator. Generation of networks has been criticised for the fact that it often does not account for all the properties of biological networks. Therefore, other simulators rely on sampling sub-networks from known GRNs (Bulcke et al., 2006; Schaffter et al., 2011). In their simulator GeneNetWeaver, Schaffter et al. (2011) use known GRNs from *Escherichia coli* and *Saccharomyces cerevisiae* as templates. It then generates a GRN with $N$ nodes by sampling $N$ nodes and the corresponding interactions from one of these templates. Again, this has the drawback of biasing the resulting GRNs towards known topologies and model organisms, resulting in networks that are possibly not representative of some biological systems. In addition, subsampling from a template network might alter the topological properties of the resulting subnetwork.

### 1.5.2 Mathematical framework

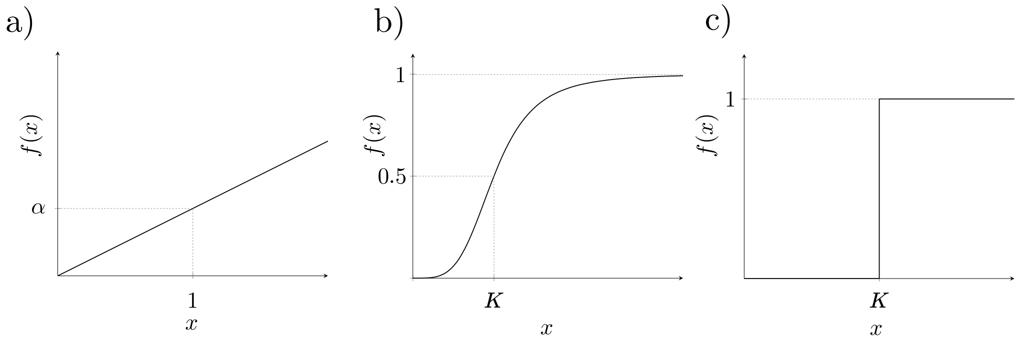
Along with a graph representing the regulations between genes, a simulator must include a model of gene expression. There are a number of mathematical frameworks that can be used to define such a model. A simple approach is to use a Boolean model, i.e. to consider each gene as a binary variable, with two possible states: activated or repressed (Bornholdt, 2008). In their activated state, the genes produce mRNAs and proteins; in their repressed state they do not. A more detailed model can be obtained under a deterministic framework (de Jong, 2002). Variables are used to represent the concentration of mRNAs or other gene products in the system. A set of differential equations model the evolution over time of these concentrations. The differential equations account for the production and decay of the different gene products, as well as the impact of other molecules' concentrations on these processes. A third approach is to employ an agent-based or stochastic model (Higham, 2008). In this case, we model the absolute abundance of each molecular species present in the system of interest. A list of biochemical reactions describes the interactions between the different species and allows to simulate the evolution of the system's state over time.

In order to evaluate network inference methods, simulated data must take the form of quantitative measurements of gene expression across a set of observations. Therefore, approaches that output numerical output, i.e. differential equation systems or agent-based models, are preferentially employed by simulators. Deterministic models have the advantage of being fast and straightforward to simulate, while stochastic models can become computationally expensive for moderate-size systems. On the other hand, the assumptions underlying deterministic models break down when some molecules are present in low numbers in the cell, which is typically the case for some transcription factors (Zlatanova & Van Holde, 2016). In this case, differential equations may not be appropriate to represent the state of the system, as biological noise plays an important role in guiding the state of the system, and one might turn to stochastic models instead. Indeed the latter intrinsically accounts for the variation arising from biological noise. Simulators such as those presented in Mendes et al. (2003), Haynes & Brent (2009), or Hache, Wierling, et al. (2009) rely on systems of ODEs for the simulations. On the contrary, Ribeiro & Lloyd-Price (2007) and Tripathi et al. (2017) offer stochastic simulators. GeneNetWeaver (Schaffter et al., 2011) also use deterministic models, but also offer an option to include biological noise through the use of Chemical Langevin Equations, which bridge the gap between stochastic and deterministic models.

### 1.5.3 Mathematical representation of gene regulation

An important part of the models of gene expression concerns the representation of expression regulation by regulatory molecules. This amounts to transcribing the GRNs into mathematical models. For example, in a Boolean model, regulators propagate their state to their targets through logic gates such as AND and OR. In the deterministic case, the expression of a target gene is modelled as a function of the concentration of its regulators. Such regulation can be represented as first order process or more complex non-linear functions (see Figure 1.6). A popular choice is to use a Hill function to express the expression of the target gene as a function of its regulation concentration (e.g Ackers et al., 1982; Bintu, Buchler, Garcia, Gerland, Hwa, Kondev, et al., 2005b). Hills functions account for the effect of promoter saturation, i.e. once the regulator is present in high concentration, a further increase of its concentration does not increase the expression of the target gene. In the case of stochastic models, the biochemical reactions represent the biological processes of regulation. They can be used to model each step of the transcription process, e.g. the binding and unbinding of regulatory molecules to their binding site. On the contrary, several such steps can be summarised in one reaction, for example the presence of the regulatory molecule triggering the creation of a target's mRNA.

Another key aspect of models of gene expression regulation is the type of molecules performing the regulation. Early models focused on simulating the production of mRNAs from genes and used



$$f(x) = \alpha x$$

$$f(x) = \frac{x^n}{K^n + x^n}$$

$$f(x) = \begin{cases} 0 & \text{if } x < K \\ 1 & \text{if } x \geq K \end{cases}$$

Figure 1.6: Possible transcription rate law functions. a) The rate law accounts for a linear dependence between the concentration of the regulator ($x$) and the transcription rate of the target gene ($f(x)$). b) The Hill function accounts for the saturation of the regulation: when $x$ is high, the variation of the transcription rate tends to zero. The parameter $K$ corresponds to the concentration at which the regulatory molecules induce a transcription rate equal to half its maximum value. c) With a step function, the target gene is only transcribed when the concentration of the regulator exceeds a certain threshold, here $K$.

mRNA levels as a proxy for the levels of the corresponding transcription factors (Bulcke et al., 2006; Di Camillo et al., 2009; Mendes et al., 2003). This has been justified by the difficulty at the time to record protein measurements experimentally, and was therefore regarded as a valid approximation. We now know that post-transcriptional regulation and other biological processes decouple the levels of mRNAs and proteins. Therefore, it is necessary to explicitly represent proteins in the models, which act as regulators of gene expression. It is to note however that none of the existing simulators account for post-transcriptional regulations. This results in simulated levels of proteins that are very close of corresponding mRNA levels. It is an important weakness of existing simulators, as in the resulting simulations the correlation observed between mRNAs and proteins is higher than what is observed experimentally. This results in a possibly biased assessment of the performance of network inference methods, since the complexity arising from post-transcriptional modification is ignored.

### 1.5.4 Model simplifications

As mentioned previously, simulations arising from mathematical models do not grasp the full complexity of experimental datasets. This arises from the simplifications made to the representations of the biological processes at play. One example is the use of mRNA concentrations as a proxy for protein levels, which amounts to ignoring translation and related processes. Another example is the focus on transcription regulation, without accounting for post-transcriptional events. Both points are however crucial, as post-transcription regulation is assumed to be one of the main reason driving the lack of correlation between RNA and corresponding protein levels (e.g. estimated to 0.41 in yeast – Vogel & Marcotte, 2012). Similarly, regulation acted by small non-coding RNAs is often left outside of simulators. They have been however found to be pervasive in biological systems and are thus likely to play a role in the complexity of experimental datasets.

Additionally, it is important to account for the impact of genetic variations on gene expression and regulatory interactions. This has been included in some simulators. For example, Pinna et al. (2011) models the effect of *cis*-acting genomic variants, i.e. modifying the expression of a gene nearby (promoter-like function), as well as *trans*-variants, which affects the strength of the regulation between a regulator and its target (likely a sequence modification of the regulator, impacting its affinity for its targets). In contrast, Schaffter et al. (2011) represents genetic variations between different individuals or observation as a multifactorial perturbation that affects the basal expression of all the genes simultaneously with a small effect. However, other simulators such as those from Tripathi et al. (2017) or Bulcke et al. (2006) do not account for the impact of genetic variation on the regulatory networks. Lastly, the inclusion of experimental noise can offer an additional layer of complexity to the resulting simulations and is important when assessing the performance of network inference methods as it clouds the patterns of regulation.

More generally, there is still a lack of tools approaching the simulation of gene expression regulation from a Systems Biology perspective, i.e. moving away from the traditional dogma of biology by accounting for the different molecular layers involved as well as the plethora of biological mechanisms via which cells can control the expression of their genes. Only with such models, will we be able to explain the gap in performance of inference tools between simulated and experimental datasets. This would also enable to researchers to quantify the loss of information related to independent analyses on single omics layers. For example, one question that has yet to be answered is how much we can learn from reconstructing GRNs from transcriptomics data alone – which is still a very popular approach to GRN reconstruction – when alternative mechanisms of expression regulation are at play, such as post-transcriptional regulation, or regulation via non-protein regulators.

## 1.6 Conclusion

The problem of understanding the relationship between genotype and phenotype has been intensively investigated in the past decades. Two orthogonal approaches have been applied to try and answer this arduous question. QTL mapping and association studies have focused on detecting genomic regions responsible for variations in phenotypic traits of interest. In parallel, network inference methods have aimed at reconstructing from observational measurements the interactions between molecular features. While network inference has firstly been applied to reconstructing gene regulatory networks from transcriptomics datasets, the focus has now shifted to integrating measurements obtained at different molecular scales. In particular, it would be very interesting to be able to decipher regulation across the different cellular layers. By gaining an understanding of these across-omics regulations, we can decipher the mechanisms by which genetic variation affects the phenotype, thus drawing a link between association studies and molecular networks. However, this is complicated by the extreme complexity of biological systems, and the pervasivity of post-transcriptional regulation events that cannot be detected from single omics studies alone. Moreover, it is necessary to start looking at biological systems in terms of causality. Statistical tools have been developed in order to infer the causal relationships between a set of variables using observational data, but their use in the context of biological systems has been limited so far. In particular, proper benchmarking of these tools with simulations that mimic biological systems is missing. Specifically, performance evaluation must account for the diversity of regulatory mechanisms through which cells control the expression of their genes. Also, the use of causal inference tools for reconstruction of regulatory networks across different omics datasets must be investigated. These directions will be explored in the present thesis.

# Chapter 2

# Simulating gene regulatory networks and gene expression with the R package sismonr

## 2.1 Introduction

A first essential step in inferring from biological datasets molecular interactions, or deciphering causal relationships among genes and gene products, is to assess the performance of the statistical methods used for such inference. In particular, it is interesting to see in which conditions or for which type of data the methods perform best or worst, and what kind of information can be extracted or is systematically missing from the data. Due to the highly complex nature of biological datasets and our incomplete knowledge of biological causal networks, the performance of network reconstruction methods, including causal inference methods, is traditionally assessed using simulated data (e.g. Ainsworth et al., 2017; Ahmed et al., 2018; Muldoon et al., 2019), although experimental datasets can also be used (Auerbach et al., 2018; Mooij et al., 2016). This consists in generating gene expression data from an underlying causal network, applying the causal inference method on the simulated data, and comparing the inferred causal graph to the one used to generate the data. Such comparison allows us to produce performance estimators such as precision, recall, etc, that inform on the method's strengths and weaknesses.

However, simulating suitable data that will provide useful insights into the statistical methods' performance is no trivial task. Each step of the simulation process must be subject to careful design. For example, the first stage of simulating data to evaluate causal inference methods is to decide on a network that describes the causal relationships between the variables to be simulated (here, genes, RNAs or proteins). A first approach is to extract a sub-network from existing biological pathways in public databases, as it is done for example in Marbach et al. (2009). Another option is to generate a simulated network with characteristic resembling those of biological networks (e.g. Di Camillo et al., 2009). Both approaches have their limitations (Bulcke et al., 2006): sampled networks from existing datasets are biased due to our incomplete knowledge of biological networks, while generated networks might be a simplification of biological networks as they might not possess all the characteristics of true biological pathways. The next step in the simulation process is to generate values for the variables (e.g. abundance or concentration of RNAs or proteins). A statistical or mathematical model is required to sample values for the different variables obeying the causal relationships dictated by the causal graph. A simple approach is to choose a distribution for each node in the graph, whose parameters depends on the values of the parent nodes, and use the sampled values as a proxy for the genes' expression. However, in order to obtain simulations that better resembles biological data, one can design a mathematical model mimicking the biological process of gene expression, describing the expression of the genes as a function of their regulators' levels. I will focus on the latter type of data simulation.

There are a number of modelling choices to be made when developing a model for gene expression: deterministic (better suited to represent the concentration of abundant molecules) versus stochastic modelling (absolute abundance of molecules), type of function describing the impact of a regulator's level on the expression of its targets (e.g. linear, Hill's function), parametrisation of the model, etc. One particular point of interest is the biological mechanisms of gene expression that are modelled. Early simulators focused on simulating the production of RNAs in order to model gene expression. They assumed that the RNA abundance of regulator genes can be used as a proxy for their activity on their targets' expression (e.g. Mendes et al., 2003). Later models included the translation of RNAs into proteins, in an effort to account for the lack of correlation found in biological datasets between RNA and protein expression (Roy et al., 2008). However, simulators focus on a certain aspect of gene expression that concerns the regulation of transcription. While this is a critical aspect of gene expression, it is not the only means by which a cell can control the expression of its genes. Post-transcriptional regulation of gene expression has been found pervasive in biological systems (Buccitelli & Selbach, 2020; Liu et al., 2016; Merchante et al., 2017). This mechanism is crucial as it impacts the observed patterns in gene expression measurements, and contributes to the complexity of biological datasets. When simulating such expression measurements, overlooking post-transcriptional relationships between genes can lead to over-simplification of the data and thus to optimistic evaluation of the methods' performance. However, there is currently a lack of simulation tools that account for post-transcriptional regulation. The reader is referred to Chapter 1 for a review of existing simulators of gene expression.

In this chapter, I am interested in developing a simulator of gene expression that models transcriptional as well as post-transcriptional regulations among genes. This new simulator, `sismonr`, is available as an R package on the CRAN at `https://CRAN.R-project.org/package=sismonr`;

the code is publicly available at `https://github.com/oliviaAB/sismonr`. `sismonr` can be used to generate realistic expression data to evaluate network or causal inference methods, or even multi-omics integration methods, as the simulator models both RNA and protein abundance for the simulated genes. It also includes the presence of non-coding genes in regulatory networks and explicitly models the ploidy of the simulated system, i.e. the number of copies of each gene present in the biological system, two points that have not yet been incorporated into gene expression simulators. Non-coding regulatory RNAs have been found pervasive in biological systems, and their inclusion in synthetic GRNs is an important factor. Indeed, regulation by RNA entails a different dynamics in biological systems than regulation by proteins, as the regulatory genes do not need to be translated, and thus the regulation occurs on a shorter timescale. In addition, transcriptomics measurements often exclude non-coding RNAs, and thus the latter act as unobserved confounders in the networks. Therefore, benchmarking network inference methods when unobserved regulatory RNAs are at play allows us to assess the impact of these hidden variables in the network reconstruction performance. Moreover, `sismonr` accounts for genetic mutations impacting the properties of the system, which is a fundamental aspect for causal inference assessment. Indeed, causal inference relies on small random perturbations between observations of a set of variables' values in order to estimate causal relationships among them. In a biological setting, and more specifically for inference of relationships between gene products, it has been argued that genetic mutations or differences among individuals in a population of interest, also termed Mendelian randomisation (Davey Smith & Ebrahim, 2003) emulate these random changes and allows researchers to extract causal information from observational expression data. In the present simulator, I accounted for different types of genetic mutations, each affecting the expression of a gene in a different way, that mimics genotype-gene expression relationships uncovered in biological experiments.

A schema of the sismonr pipeline is presented in Figure 2.1. This work has been published as an Application Note in the Bioinformatics journal, in which I provide an overview of the simulator and its implementation (Appendix C). A review of existing simulators and a comparison of `sismonr` with previous simulators have been made available as a Supplementary file for the Application Note (Supplementary file 1 available online at `https://academic.oup.com/bioinformatics/article/36/9/2938/5711287`). I repeat this information, specifically the comparison of `sismonr` with existing tools, in Section 2.5 of the present chapter. In a second Supplementary file for the Application Note, I provided more details about the implementation of `sismonr` and the mathematical model used to simulate gene expression data. This work is presented in the present chapter (Sections 2.2 to 2.4). Finally, examples of the use of `sismonr` along with the corresponding R code have been made available as a third Supplementary file, and can be found in Appendix D. In addition, a comprehensive tutorial presenting how to use `sismonr` is available at `https://oliviaab.github.io/sismonr/`. Table 2.1 displays the abbreviations used throughout this document and in `sismonr`.

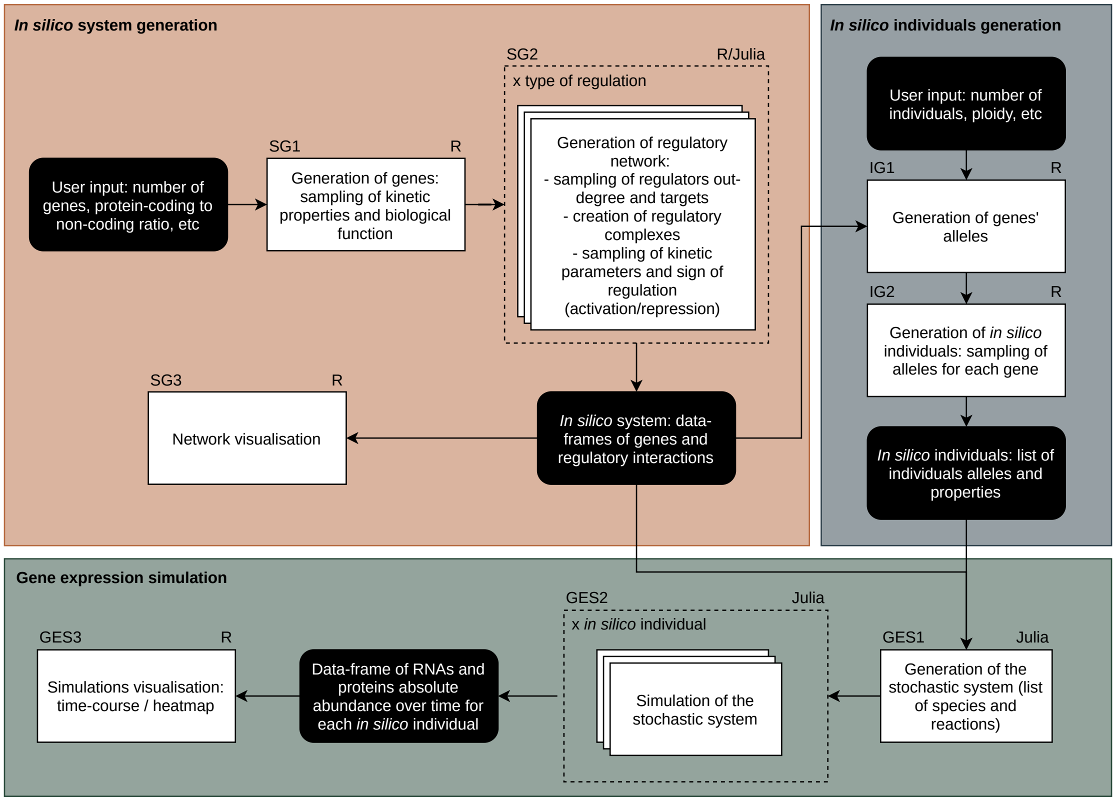

*In silico* system generation
User input: number of genes, protein-coding to non-coding ratio, etc
SG1
R
Generation of genes: sampling of kinetic properties and biological function
SG2
R/Julia
x type of regulation
Generation of regulatory network:
- sampling of regulators out-degree and targets
- creation of regulatory complexes
- sampling of kinetic parameters and sign of regulation (activation/repression)
SG3
R
Network visualisation
*In silico* system: data-frames of genes and regulatory interactions
*In silico* individuals generation
User input: number of individuals, ploidy, etc
IG1
R
Generation of genes' alleles
IG2
R
Generation of *in silico* individuals: sampling of alleles for each gene
*In silico* individuals: list of individuals alleles and properties
Gene expression simulation
GES3
R
Simulations visualisation: time-course / heatmap
Data-frame of RNAs and proteins absolute abundance over time for each *in silico* individual
GES2
Julia
x *in silico* individual
Simulation of the stochastic system
GES1
Julia
Generation of the stochastic system (list of species and reactions)

Figure 2.1: Schema of the `sismonr` package pipeline. Input/output datasets and endpoints are presented in black rounded boxes and algorithm steps are shown in white rectangle boxes. The coloured rectangles outline different steps in the algorithm. The programming language used for each step is indicated above the top-right corner of each algorithm step.

## 2.2 Creating the *in silico* system

In `sismonr`, an *in silico* system is characterised by the list of protein-coding and non-coding genes and regulatory complexes existing in the system, together with the regulatory interactions among them, summarised in a graph representing the gene regulatory network (GRN) of the system. In `sismonr`, I define a gene as a DNA sequence that is transcribed into mRNAs (for protein-coding genes) or regulatory non-coding RNAs (for non-coding genes). The mRNAs of a protein-coding gene are then translated into proteins. I detail below how the genes are represented in this model. `sismonr` generates *in silico* systems in two main steps: it first creates a list of *in silico* genes constituting the system (box SG1 in Figure 2.1), and secondly it generates the GRN linking the genes (box SG2 in Figure 2.1). In addition, the system is characterised by a ploidy level (parameter `ploidy`), which describes the number of copies of each gene carried by the different *in silico* individuals.

Table 2.1: List of abbreviations used for `sismonr`.

| Abbreviation | Meaning |
|---|---|
| TC | Transcription |
| TL | Translation |
| RD | RNA decay |
| PD | Protein decay |
| PTM | Post-translational modification |
| GRN | Gene regulatory network |
| PC | Protein-coding |
| NC | Non-coding |
| Pr | Promoter binding site |
| RBS | RNA binding site |
| P | Protein |
| Pm | Modified protein |
| C | Regulatory complex |

### 2.2.1 Creating the genes

The number of genes $G$ in the system is defined by the user. The **coding status** of each gene (i.e. protein-coding or non-coding) is randomly sampled according to the parameter `PC.p`, giving the probability of a gene to be protein-coding. This parameter can be specified by the user; the default value is (arbitrarily) set to 0.7. The coding status of a gene determines what its **active products** will be, that is which of its products (RNAs or proteins) will perform the regulatory actions on the gene's targets. The active products of a protein-coding gene will be the protein molecules encoded by the gene, and its RNA molecules for a non-coding gene. If in the GRN a protein-coding gene is targeted for post-translational modification, only the modified version of its protein will be active (i.e. will be able to carry the gene's regulatory role).

Each gene is assigned a *biological function*, describing the type of regulation its active products will exert on its targets in the GRN. The biological function is randomly sampled for each gene depending on its coding status and according to the default (and arbitrary) probabilities given in Table 2.2. It must be noted that the biological function "metabolic enzyme" for protein-coding genes indicates that the genes cannot perform any regulatory action in the GRN. This biological function has been included to account for genes that are only target in the GRN and not regulators, and to allow for a future extension of `sismonr` where metabolic reactions will be simulated as well.

The user can customise the composition of the system by specifying values for some or all of these probabilities. Taking the example of the protein-coding genes, if the user specify values for only some of the biological function probabilities, and the sum of these values is equal or greater than one (say `PC.TC.p` $= 0.9$, and `PC.TL.p` $= 0.6$), then the specified probabilities are normalised so that

Table 2.2: Possible biological functions of protein-coding and non-coding genes and associated default probabilities, with the name of the parameters representing these probabilities in the `sismonr` package.

| Coding status | Biological function | Default probability | Parameter name |
|---|---|---|---|
| Protein-coding | Regulator of transcription | 0.4 | `PC.TC.p` |
| | Regulator of translation | 0.3 | `PC.TL.p` |
| | Regulator of RNA decay | 0.1 | `PC.RD.p` |
| | Regulator of protein decay | 0.1 | `PC.PD.p` |
| | Regulator of protein post-translational modification | 0.05 | `PC.PTM.p` |
| | Metabolic enzyme | 0.05 | `PC.MR.p` |
| Non-coding | Regulator of transcription | 0.3 | `NC.TC.p` |
| | Regulator of translation | 0.3 | `NC.TL.p` |
| | Regulator of RNA decay | 0.3 | `NC.RD.p` |
| | Regulator of protein decay | 0.05 | `NC.PD.p` |
| | Regulator of protein post-translational modification | 0.05 | `NC.PTM.p` |

their sum equals one (here `PC.TC.p` $= 0.6$ and `PC.TL.p` $= 0.4$) and the other probabilities are set to zero. Consequently, there will only be regulators of transcription and translation in the system. On the other hand, if the specified values don't sum up to one (e.g. `PC.TC.p` $= 0.3$ and `PC.TL.p` $= 0.3$), the non-specified probabilities are assigned equal non-null values such that the sum of all probabilities is one (i.e. `PC.RD.p` = `PC.PD.p` = `PC.PTM.p` = `PC.MR.p` $= 0.1$).

Table 2.3: Kinetic parameters assigned to each gene and their default sampling distribution with the name of the parameters representing these distributions in the `sismonr` package. The notation of each parameter for a gene $G_i$ that will be used in the following sections as well as its unit are indicated below the parameter.

| Kinetic parameter | Default sampling distribution | Parameter name |
|---|---|---|
| Transcription rate<br>$\mathrm{TCr}_i^{basal}$<br>(RNA/sec) | $10^x/60$<br>with $x \sim \mathcal{N}(\mu = -0.92, \sigma = 0.35)$ | `basal_transcription_rate_samplingfct` |
| Translation rate<br>$\mathrm{TLr}_i^{basal}$<br>(protein/RNA/sec) | $10^x/3600$<br>with $x \sim \mathcal{N}(\mu = 2.146, \sigma = 0.7)$ | `basal_translation_rate_samplingfct` |
| RNA lifetime<br>$1/\mathrm{RDr}_i^{basal}$<br>(sec) | $10^x \times 60$<br>with $x \sim \mathcal{N}(\mu = 1.36, \sigma = 0.2)$ | `basal_RNAlifetime_samplingfct` |
| Protein lifetime<br>$1/\mathrm{PDr}_i^{basal}$<br>(sec) | $2^x \times 60$<br>with $x \sim \mathcal{N}(\mu = 5.43, \sigma = 1)$ | `basal_protlifetime_samplingfct` |

Lastly, each gene is assigned a number of kinetic parameters describing its transcription, translation (if applicable) and decay rates. The values for these parameters are sampled from functions that can be customised by the user. The default functions are shown in Table 2.3 overleaf. These sampling distributions have been designed according to experimental results found in the literature. For the sake of consistency, I tried to retrieve when possible values measured in the organism *Saccharomyces cerevisiae* (budding yeast). If these were not available, I used experiments performed on other eukaryotes, and the organism studied will be indicated in brackets. The advantage of using *S. cerevisiae* is two-fold: first, it is used as a model organism for eukaryotes and hence is extensively studied. Second, the relative high RNA and protein turnovers allows us to reduce the simulation time required to observe the effects of gene regulation on expression levels. In accordance with the results of (Pelechano et al., 2010), I designed the sampling distribution of transcription rates such that sampled values range from $10^{-4}$ to $10^{-2}$ RNA/sec. The translation rate sampling distribution has been constructed using results from (Siwiak & Zielenkiewicz, 2010) and (Schwanhäusser et al., 2011) (using work on mouse fibroblasts), and provide values ranging from $10^{-4}$ to $1$ protein/RNA/sec. The RNA and protein lifetime sampling distributions are based on (Wang et al., 2002) and (Belle et al., 2006), respectively, and give values that approximately range from nine minutes to one hour for the RNAs and from 11 minutes to 2.7 hours for the proteins. Once sampled, the molecule lifetimes are transformed into decay rates according to the formula $decay\ rate = \frac{1}{lifetime}$.

Note that the different kinetic rates associated with the expression of genes can vary significantly between different classes of organism (e.g. bacteria vs yeast or human). For example, while the half-life of RNAs ranges from one to 15 minutes in *Escherichia coli*, it ranges from 10 minutes to a few hours for yeast, and from five hours to a day in humans. This impacts the amount of RNA and proteins of the genes in the system.

### 2.2.2 Creating the regulatory network

The next step of the algorithm is to construct the GRN that specifies the regulations among the genes. First, a regulatory network is created separately for each type of regulation (i.e. regulation of transcription, translation, RNA decay, protein decay, and post-translational modification). To create one of the aforementioned regulatory networks, the genes that will act as regulators in the network are selected, i.e. the genes with the corresponding biological function (e.g. only regulators of transcription will be selected as regulators for the transcription regulatory network). The set of potential target genes is selected according to the type of regulation considered: all genes can be targeted for transcription or RNA decay regulation, but regulation impacting translation, protein decay or protein post-translational modification can only affect protein-coding genes.

For a given type of regulation, a regulatory network for protein-coding regulators and one for non-coding regulators are created separately (both networks share the same set of potential target genes). This is to allow the user to give different properties to protein-coding and non-coding regulators in a same regulatory network. For both protein-coding and non-coding regulated networks, the different steps of the generation process are:

**1. Sampling of the out-degree $k_{out}$ (i.e. number of targets) of each regulator**, according to either a power-law or an exponential distribution (parameter `[Regulation type].[Coding status].outdeg.distr`[1] set to either "powerlaw" or "exponential"), with the following densities:

$$\begin{aligned} &\text{Power-law distribution:} && P(k_{out}) \propto k_{out}^{-\gamma}, \quad \text{or} \\ &\text{Exponential distribution:} && P(k_{out}) \propto \frac{1}{\gamma} \cdot e^{-\frac{k_{out}}{\gamma}} \end{aligned}$$

By default, the out-degree distribution for both protein-coding and non-coding regulators in all five regulatory networks is chosen to be a power-law distribution. The value of the $\gamma$ parameter in the out-degree distribution is determined by the parameter `[Regulation type].[Coding status].outdeg.exp` (set in the transcription regulatory network to three for protein-coding regulators – according to (Albert & Barabási, 2002) – and to five for non-coding regulators, and in all other regulatory networks to four and six for protein-coding and non-coding regulators, respectively, to ensure sparsity). Please note that a higher $\gamma$ value implies a sparser network.

**2. For each regulator (starting with the regulators with the highest out-degrees), sampling of its targets** from the set of potential target genes. The probability of each potential target being regulated by the considered regulator is computed following one of two possible **preferential attachment schemes** (parameter `[Regulation type].[Coding status].outdeg.distr` set to either "powerlaw" or "exponential") defining the probability of a gene being chosen as target of a new regulator given its current **in-degree** (i.e. the number of regulators already targeting it) during the graph construction process. The first scheme, proposed by (Barabási & Albert, 1999) stipulates that the probability of a node with in-degree $k_{in}$ being chosen as target is:

$$P(k_{in}) \propto k_{in} + 1$$

The $+1$ term ensures that genes with an in-degree of zero still have a non-null probability to be chosen. Under this scheme, the resulting in-degree distribution is supposed to follow a power-law distribution (given a sufficiently large number of nodes). The second preferential attachment scheme has been proposed by (Lachgar & Achahbar, 2016). They set:

$$P(k_{in}) \propto 1 - \frac{k_{in}}{\sum_{\text{genes}} k_{in}}$$

---

[1] E.g. the out-degree distribution of protein-coding regulators (`Coding status` = PC) in the transcription regulatory network (`Regulation type` = TC) is dictated by the parameter `TC.PC.outdeg.distr`.

In this scenario, genes that have a lower in-degree are more likely to be chosen as target for the current considered regulator. This preferential attachment is supposed to lead to an in-degree distribution that is approximately exponential. However, this does not appear to be the case for graphs with a high ratio of regulators to target genes. Note that the probabilities are computed by first computing the right-hand side of the preferential attachment formula for each possible target, then dividing by the sum of these values across all possible targets, such that the sum of the $P(k_{in})$ across all possible targets equals one. According to these probabilities, a number of targets corresponding to the out-degree of the considered regulator are sampled. An edge is created in the regulatory network between the regulator and each of its targets. The parameter `[Regulation type].[Coding status].autoregproba` defines the probability of each regulator to perform autoregulation (i.e. to select itself as its target). Setting the parameter to zero ensures that there will not be any autoregulation in the network. The boolean parameter `[Regulation type ID].[Coding status].twonodesloop` dictates whether or not the target of a regulator can regulate the regulator in the same regulatory network (only valid if the target is also a regulator in this regulatory network).

Once generated, the protein-coding-regulated network and the non-coding-regulated network (for the considered type of regulation) are merged into one network, i.e. the edges from the protein-coding regulators network and those from the non-coding regulators network are added to create a single network. There won't be any overlapping edges as the set of source nodes (or regulators) in the two networks are distinct.

**3. (Optional) Creation of regulatory complexes among regulators targeting a common gene**. The parameter `regcomplexes` dictates the creation of these complexes. If set to "none", this step will be overlooked and no regulatory complexes will be created. If set to "prot", only protein-coding regulators of a common target can form a regulatory complex. If set to "both", all regulators of a common target can form a complex. The size of the complexes is determined by the parameter `regcomplexes.size`, with a default value of two (for computational efficiency).

I now describe the process that allows `sismonr` to generate regulatory complexes. Let $n_{\text{reg}}$ be the number of regulators targeting a given gene ($n_{\text{reg}} > 1$). I will define the number of "complex trials" as the result of the integer division $n_{\text{reg}}/$`regcomplexes.size` (i.e. the maximum number of complexes of size `recomplexes.size` that can be simultaneously created from the $n_{\text{reg}}$ regulators). For each complex trial, there is a probability `regcomplexes.p` (arbitrary default value of 0.3) to create a regulatory complex. If successful, then `recomplexes.size` regulators are sampled with replacement from the list of the gene's regulators. In the regulatory graph, the edges between the selected regulators and the target are removed. Instead, an edge is added between the created regulatory complex and the target. It must be noted that some components of a regulatory complex can be present in more than one copy in the complex (e.g. a complex of size two can be composed of two copies of a same gene product, that is can be a homodimer).

The composition of each created regulatory complex is stored, and the complexes are each assigned a formation and dissociation rate, defining the rates at which its components assemble to form a complex or dissociate from each other. By default, the rates are sampled from the distributions specified in Table 2.4. I based these sampling distributions on experimental values (Kastritis & Bonvin, 2012; Schreiber et al., 2009). It has been shown that association rates of protein complexes range from $10^5$ to $10^9$ $\text{M}^{-1}.\text{s}^{-1}$. Assuming that the cellular volume is $\sim 5 \times 10^{-14}$ L (*S. cerevisiae* cell, see Milo & Phillips (2016)), we can convert these values from $\text{M}^{-1}.\text{s}.^{-1}$ to $\text{molecule}^{-1}.\text{s}.^{-1}$ with 1 M $= N_{av} \times V_{cellular} \sim 6 \times 10^{23} \times 5 \times 10^{-14} \sim 10^{10}$ molecules. From the same sources, dissociation rates range from $10^{-2}$ to $10^6$ $\text{s}^{-1}$.

Table 2.4: Kinetic parameters assigned to each regulatory complex and their default sampling distributions with the name of the parameters representing these distributions in the `sismonr` package. The notation of each parameter for a gene $G_i$ that will be used in the following sections as well as its unit are indicated below the parameter.

| Kinetic parameter | Default sampling distribution | Parameter name |
|---|---|---|
| Formation rate<br>$\text{formr}_i$<br>(/molecule/s) | $10^x$, $x \sim \mathcal{N}(-3, 0.7)$ | `complexesformationrate_samplingfct` |
| Dissociation rate<br>$\text{dissr}_i$ (/s) | $10^x$, $x \sim \mathcal{N}(2, 1.2)$ | `complexesdissociationrate_samplingfct` |

**4. Sampling of the sign of the edges in the regulatory network**, a plus sign (encoded as “1”) indicating a positive regulation or **activation** and a minus sign (encoded as “-1”) a negative regulation or **repression**. The meaning of activating/repressing regulation varies according to the type of regulation considered:

- For regulation of transcription, a positive edge implies that the regulator increases the transcription rate of its target; while a negative sign on the edge means that the regulator suppresses the transcription of the target. The same definition of activation or inhibition applies to translation;

- For RNA and protein decay, there can only be positive regulations, i.e. the regulator increases the decay rate of its target. In biological systems there exist molecules protecting their target from degradation, but for the sake of computational simplicity I simplified the model to not include such a type of regulation;

- For protein post-translational regulation, a positive edge means that the regulator transforms the “original” or non-modified protein form of its target (similar to the action of a protein kinase on its target(s)), while a negative regulation implies that the regulator transforms back the modified version of its target into its original form (similar to the action of a phosphatase on its targets).

The probability of each edge being positive is defined separately for the different types of regulation with the parameter `[Regulation type].pos.p` set by default to 0.5. As previously mentioned, `RD.pos.p` and `PD.pos.p` are automatically set to one as RNA or protein decay regulation can only be positive.

**5. Sampling of kinetic parameters for each edge in the regulatory network.** The sampled kinetic rates depend on the type of regulation considered. Table 2.6 displays the parameters for each type of regulation as well as the default distributions from which they are sampled. These parameters arise from the way the different types of regulation are modelled (see Section 2.4.1 and in particular Figure 2.2).

The sampling distributions for the values of these parameters have been chosen according to experimental evidence. For the parameters relating to regulation of transcription, I made use of experimental evidence from (Nalefski et al., 2006), (Biggin, 2011) and (Milo & Phillips, 2016). More specifically, the unbinding rates of regulators from their target are directly sampled from the distribution specified in Table 2.6. As the dissociation constant $K_d = \frac{\text{unbinding rate}}{\text{binding rate}}$ is often close to the steady-state concentration of the regulators (Milo & Phillips, 2016), I compute the binding rate of a transcription regulatory edge $i \rightarrow j$ as follows:

- Sample an unbinding rate value $\text{unbindr}_{ij}^{TC}$.
- Compute the steady-state abundance of regulator $i$'s active products (i.e. RNAs for non-coding regulators and proteins for protein-coding regulators), according to the balance formula $SS_i = \frac{\text{production rate}}{\text{decay rate}}$. If the regulator is a regulatory complex, the minimum value among the steady-state abundances of its components is selected.
- Compute the value $\mu = \frac{\text{unbindr}_{ij}^{TC}}{SS_i}$.
- Sample a value for the binding rate $\text{bindr}_{ij}^{TC}$ according to:

$$\text{bindr}_{ij}^{TC} = 10^x, \;\; x \sim \mathcal{TN}(\log_{10}(\mu), 0.1, \log_{10}(\mu), \infty) \tag{2.1}$$

where $\mathcal{TN}(\mu, \sigma, a, b)$ is a truncated Gaussian distribution with mean $\mu$ and s.d. $\sigma$, with $a$ and $b$ as lower and upper boundaries, respectively. Indeed, we want to sample values for the binding and unbinding rates such that the regulator steady-state abundance ensures that the occupancy rate of its target binding site is at least 0.5. As the dissociation constant $K_d$ corresponds to the abundance of regulators at which the target binding site has an occupancy probability of 0.5, hence we want $K_d^{ij} = \frac{\text{unbindr}_{ij}^{TC}}{\text{bindr}_{ij}^{TC}} < SS_i \Rightarrow \text{bindr}_{ij}^{TC} > \frac{\text{unbindr}_{ij}^{TC}}{SS_i}$.

I use the same reasoning and values for kinetic parameters corresponding to translation regulatory edges. It must be noted that each edge corresponding to a transcription or translation regulation is assigned a fold-change, corresponding to the change in the rate of transcription or translation of the target gene when the regulator is bound. If such an edge is assigned a negative sign in step 4, i.e. represents a repression, the associated fold-change is automatically set to zero. This means that when the regulator is bound the transcription or translation of the target is blocked. In case of an activation, I chose a minimal fold-change value of $1.5$ to represent moderate to strong regulation. In absence of experimental estimates for the rates of regulated RNA degradation, protein degradation or post-translational modification, I chose a sampling distribution spanning several orders of magnitude (based on experimental values of $k_{cat}/K_M$ of enzymes) to allow for a wide range of regulatory behaviours.

### 2.2.3 Customising the system

The `sismonr` package gives the opportunity to the user to customise the system, by adding genes, or adding/removing edges or regulatory complexes. More details are available in the online tutorial. Note that when adding regulatory complexes in the system, the user can create complexes of complexes: for example the products of genes $G_1$ and $G_2$ can form a first complex $C_1$. The products of gene $G_3$ can bind to the complex $C_1$ to form a second regulatory complex $C_2$. This allows for example to model the formation of a multimer as a multi-step process. In addition, the complexes are not required to have the same size, i.e. in a same system complexes composed of a different number of constituents can coexist.

## 2.3 Creating the *in silico* population

Similarly to different individuals of a same species carrying different alleles of the same set of genes that characterise this species, I want to simulate the system's behaviour for different *in silico* individuals that carry different alleles of the genes present in the *in silico* system. The genetic variations between the different alleles of a gene impact the kinetic properties of the gene or of the regulatory interactions. The same regulatory interactions described by the GRN in the *in silico* system apply to the genes of each *in silico* individual, but their properties can be impacted by the genetic variations simulated.

### 2.3.1 Modelling genetic variation

In a biological system, a genetic mutation in the sequence[2] of a gene can have different consequences (Pai et al., 2015) on the gene's expression or activity. In `sismonr`, similarly to (Pinna et al., 2011), I consider two categories of mutations: *cis*-mutations, i.e. that directly affect the expression of a

---

[2] Here the "sequence" of a gene comprises coding as well as regulatory regions of the DNA.

Table 2.5: The different modelled mutations, their associated QTL coefficient effect variables and the effect of these mutations on the different gene kinetic parameters in `sismonr`.

| Kinetic parameter affected | QTL effect coefficient name | Effect of the mutation |
|---|---|---|
| Transcription rate | `qtlTCrate` | Mutation that affects the transcription rate of the gene |
| Translation rate | `qtlTLrate` | Mutation that affects the translation rate of the gene<br>*(only applicable to protein-coding genes)* |
| RNA decay rate | `qtlRDrate` | Mutation that affects the RNA decay rate of the gene |
| Protein decay rate | `qtlPDrate` | Mutation that affects the protein decay rate of the gene<br>*(only applicable to protein-coding genes)* |
| Binding rate of gene's transcription regulators | `qtlTCregbind` | Mutation that affects the binding rate of transcription regulators to the gene's binding sites |
| Binding rate of gene's translation regulators | `qtlTLregbind` | Mutation that affects the binding rate of translation regulators to the gene's RNA binding sites<br>*(only applicable to protein-coding genes)* |
| RNA decay rate triggered by RNA decay regulators | `qtlRDregrate` | Mutation that affects the rate at which RNA decay regulators trigger the decay of the gene's RNAs |
| Protein decay rate triggered by protein decay regulators | `qtlPDregrate` | Mutation that affects the rate at which protein decay regulators trigger the decay of the gene's proteins<br>*(only applicable to protein-coding genes)* |
| Protein post-translational modification rate triggered by post-translational regulators | `qtlPTMregrate` | Mutation that affects the rate at which protein post-translational regulators trigger the gene's proteins modification<br>*(only applicable to protein-coding genes)* |
| Gene's active products activity rate | `qtlactivity` | Mutation that affects the activity of a gene's active products, i.e. the rate at which its active products bind their target's binding sites or trigger the modification or degradation of its targets |

Table 2.6: Kinetic parameters assigned to each edge (interaction) in the GRN according to the type of regulation and their default sampling distributions with the name of the parameters representing these distributions in the `sismonr` package. The notation of each parameter for the edge $i \rightarrow j$ that will be used in the following sections as well as its unit are indicated below the parameter. The distribution $\mathcal{TN}(\mu, \sigma, a, b)$ is a truncated Gaussian distribution with mean $\mu$ and s.d. $\sigma$, with $a$ and $b$ as lower and upper boundaries, respectively.

| Regulation type | Kinetic parameter | Parameter name | Default sampling distribution |
|---|---|---|---|
| Transcription regulation | Binding rate of the regulator on the target promoter – $\text{bindr}_{ij}^{TC}$ (/molecule/s) | `TCbindingrate_samplingfct` | see in text |
| | Unbinding rate of the regulator from the target promoter – $\text{unbindr}_{ij}^{TC}$ (/s) | `TCunbindingrate_samplingfct` | $10^x, x \sim \mathcal{N}(-3, 0.2)$ |
| | Multiplying factor of the target's transcription rate when the regulator is bound – $\text{FC}_{ij}^{TC}$ (1) | `TCfoldchange_samplingfct` | $\mathcal{TN}(3, 10, 1.5, \infty)$ |
| Translation regulation | Binding rate of the regulator on the target RNA binding site – $\text{bindr}_{ij}^{TL}$ (/molecule/s) | `TLbindingrate_samplingfct` | see in text |
| | Unbinding rate of the regulator from the target RNA binding site – $\text{unbindr}_{ij}^{TL}$ (/s) | `TLunbindingrate_samplingfct` | $10^x, x \sim \mathcal{N}(-3, 0.2)$ |
| | Multiplying factor of the target's translation rate when the regulator is bound – $\text{FC}_{ij}^{TL}$ (1) | `TLfoldchange_samplingfct` | $\mathcal{TN}(3, 10, 1.5, \infty)$ |
| RNA decay regulation | Rate at which a regulator molecule encountering its target RNA triggers the target's degradation – $\text{regrate}_{ij}^{RD}$ (/molecule/s) | `RDregrate_samplingfct` | $10^x, x \sim \mathcal{N}(-4, 1.1)$ |
| Protein decay regulation | Rate at which a regulator molecule encountering its target protein triggers the target's degradation – $\text{regrate}_{ij}^{PD}$ (/molecule/s) | `PDregrate_samplingfct` | $10^x, x \sim \mathcal{N}(-4, 1.1)$ |
| Post-translational modification | Rate at which a regulator molecule encountering its target protein triggers its modification – $\text{regrate}_{ij}^{PTM}$ (/molecule/s) | `PTMregrate_samplingfct` | $10^x, x \sim \mathcal{N}(-4, 1.1)$ |

gene (e.g. its basal transcription or translation rate, or the affinity of its binding sites for regulatory molecules) and *trans*-mutations, which do not impact the expression of a gene but only the activity of its active products, e.g. the affinity of its proteins for the regulatory sequences of its targets. Instead of modelling a gene allele as a genetic sequence directly, I represent the allele by the quantitative effects of its different mutations, termed **QTL effect coefficients**.[3] More specifically, I model different types of mutations, each impacting a specific kinetic parameter of the gene, and for each type of mutation the associated QTL effect coefficient will be applied (i.e. multiplied) to the corresponding kinetic parameter in the stochastic system (see Section 2.4.1). The different types of mutations and associated QTL coefficients are shown in Table 2.5. QTL effect coefficients can only take real positive values. As they are multiplicative coefficients, a gene allele with a QTL effect coefficient of one for a given kinetic parameter indicates that the allele carries no mutation affecting this kinetic parameter. A coefficient larger than one corresponds to mutations that increase the concerned kinetic parameter, whereas a coefficient smaller than one corresponds to mutations decreasing the kinetic parameter. For example, the following gene allele:

$$\{\texttt{qtlTCrate} = 1;\ \texttt{qtlTLrate} = 1;\ \texttt{qtlRDrate} = 1;\ \texttt{qtlPDrate} = 1;\ \texttt{qtlTCregbind} = 1;\\ \texttt{qtlTLregbind} = 1;\ \texttt{qtlRDregrate} = 1;\ \texttt{qtlPDregrate} = 1;\ \texttt{qtlPTMregrate} = 1\}$$

corresponds to the reference version of the considered gene, as it has no mutation affecting any of its kinetic parameters. The allele:

$$\{\texttt{qtlTCrate} = 0.5;\ \texttt{qtlTLrate} = 1;\ \texttt{qtlRDrate} = 1;\ \texttt{qtlPDrate} = 1;\ \texttt{qtlTCregbind} = 1;\\ \texttt{qtlTLregbind} = 1;\ \texttt{qtlRDregrate} = 1;\ \texttt{qtlPDregrate} = 1;\ \texttt{qtlPTMregrate} = 1\}$$

has a mutation affecting its transcription rate, more specifically its transcription rate is divided by two compared to the original version of the gene.

### 2.3.2 Creating the segregating gene alleles

In order to create a population of *in silico* individuals, the first step is to construct for each gene of the *in silico* system the list of alleles of this gene existing in the *in silico* population (box IG1 in Figure 2.1). The number of alleles existing for each gene is controlled by the parameter `ngenevariants` (with an arbitrary default value of five). The generation of the list of alleles for any given gene follows the steps:

**1. Sampling the number of mutations for each allele**. The alleles of protein-coding genes can possess at most 10 different types of mutations, whereas those of non-coding genes can only possess

---

[3] Note however that simulating genomic data for each individual can easily be achieved with tools such as `sim1000G` (Dimitromanolakis et al., 2019) or `PedigreeSim` (Voorrips & Maliepaard, 2012), for example. These genomic data can in turn be transformed into QTL effect coefficients by choosing causal variants and assigning them an effect on the genes or GRN properties.

five different types of mutations (see Table 2.5). The number of mutations carried by each allele is sampled from a discrete uniform distribution $\mathcal{DU}(1:10)$ for protein-coding genes and $\mathcal{DU}(1:5)$ for non-coding genes.

**2. Defining the types of mutations that each allele possesses**. For each allele, the types of mutations they carry is sampled from the list of possible mutations (defined in Table 2.5), according to the number of mutations sampled in step 1.

**3. Sampling the QTL effect coefficient of each mutation**. For each allele, the QTL effect coefficients corresponding to the mutations selected in step 2 are assigned a value, sampled from the default distribution shown in Table 2.7. Values of the QTL effect coefficients for the mutations that the alleles do not possess (i.e. not sampled in step 2) are set to one.

Table 2.7: Default sampling distribution of the QTL effect coefficients with the name of the variable representing this distribution in the `sismonr` package. The distribution $\mathcal{TN}(\mu, \sigma, a, b)$ is a truncated Gaussian distribution with mean $\mu$ and s.d. $\sigma$, with $a$ and $b$ as lower and upper boundaries, respectively.

| Default sampling distribution | Parameter name |
|---|---|
| $\mathcal{TN}(1, 0.1, 0, \infty)$ | `qtleffect_samplingfct` |

### 2.3.3 Creating the *in silico* individuals

An *in silico* individual is characterised by the list of alleles that it carries for each gene of the *in silico* system. The *in silico* individual is created by sampling (with replacement) for each gene a number of alleles from the list of existing alleles (box IG2 in Figure 2.1). The number of homologs of each gene that the individuals carry is defined by the ploidy of the simulated system. By default, the different alleles of the genes have the same probability of being chosen, but the user can define these probabilities for each allele of each gene. As the sampling is performed with replacement, an individual can carry more than one copy of the same allele for a given gene (if the ploidy is at least two). An individual can have at most $p$ different versions of each gene, or carry at most $p$ copies of the same allele, if $p$ is the ploidy of the system.

During the generation of each *in silico* individual, the initial abundance of each molecule (i.e. RNAs and proteins for each allele of each gene) is computed as its steady-state abundance in absence of any regulation. The parameter `initialNoise` controls whether or not noise is added to these values. If `initialNoise` is set to `TRUE` (default behaviour), the initial abundance of a given molecule is instead sampled from a normal distribution with a mean equal to this no-regulation steady state abundance, and a standard deviation equal to the square root of the aforementioned steady-state abundance. Note that the square root of the mean is used as standard deviation to reduce the variation for molecules with low abundance.

The user can decide if the initial abundance of the different molecules in the system are the same for all the *in silico* individuals (parameter `sameInit` set to `TRUE` or `FALSE`). If not, the *in silico* individuals are assigned for the RNA and protein versions of each gene an *initial abundance variation coefficient* that will be multiplied to the default initial abundance of the corresponding molecules (see Section 2.4.1 for the computation of the initial abundance of each molecule).

## 2.4 Simulating the system

In order to simulate the expression profile of the genes (i.e. their RNA and protein abundance over time) for each *in silico* individual, a stochastic system is generated (box GES1 in Figure 2.1). The stochastic system is defined by the list of all molecular entities in the system as well as the list of biochemical reactions and associated rates occurring in the system. A Stochastic Simulation Algorithm (Gillespie, 2007) is then used to simulate the evolution of the abundance of the different entities over time (box GES2 in Figure 2.1). I preferred a stochastic approach over a deterministic one, as a deterministic model is only valid if the abundance of each molecule in the system is high enough for any variation in this abundance to be modelled as a continuous change. This is not true for molecules present in only a few copies, as it can be the case for some low-expressed genes, for example. In such a microscopic setting, in which the molecules are present in very small quantity, the deterministic assumption is challenged by the fluctuation in the timing of the different reactions. For these reasons, I prefer to run stochastic simulations to model the behaviour of a system, the major drawback being a commensurate computational burden.

### 2.4.1 Generating the stochastic system

A stochastic system is generated from the *in silico* system, according to the kinetic properties of the genes and the regulations specified by the GRN. It must be noted that while the list of entities and the list of reactions in the system is identical for all the *in silico* individuals, the rates of the reactions can differ because of the genetic mutations affecting the kinetic properties of the system. For the sake of clarity, I first present the different steps of the construction of the *in silico* system assuming that the ploidy of the *in silico* system is one (only one copy of each gene). Next, I explain how to introduce a higher ploidy in the model. A schematic representation of the model of expression regulation used, detailed in the following section, is presented in Figure 2.2.

#### DNA and transcription

I first construct the different transcription reactions occurring in the system and the associated entities. In the simple case of a gene $G_i$ not targeted by any transcription regulator, its transcription

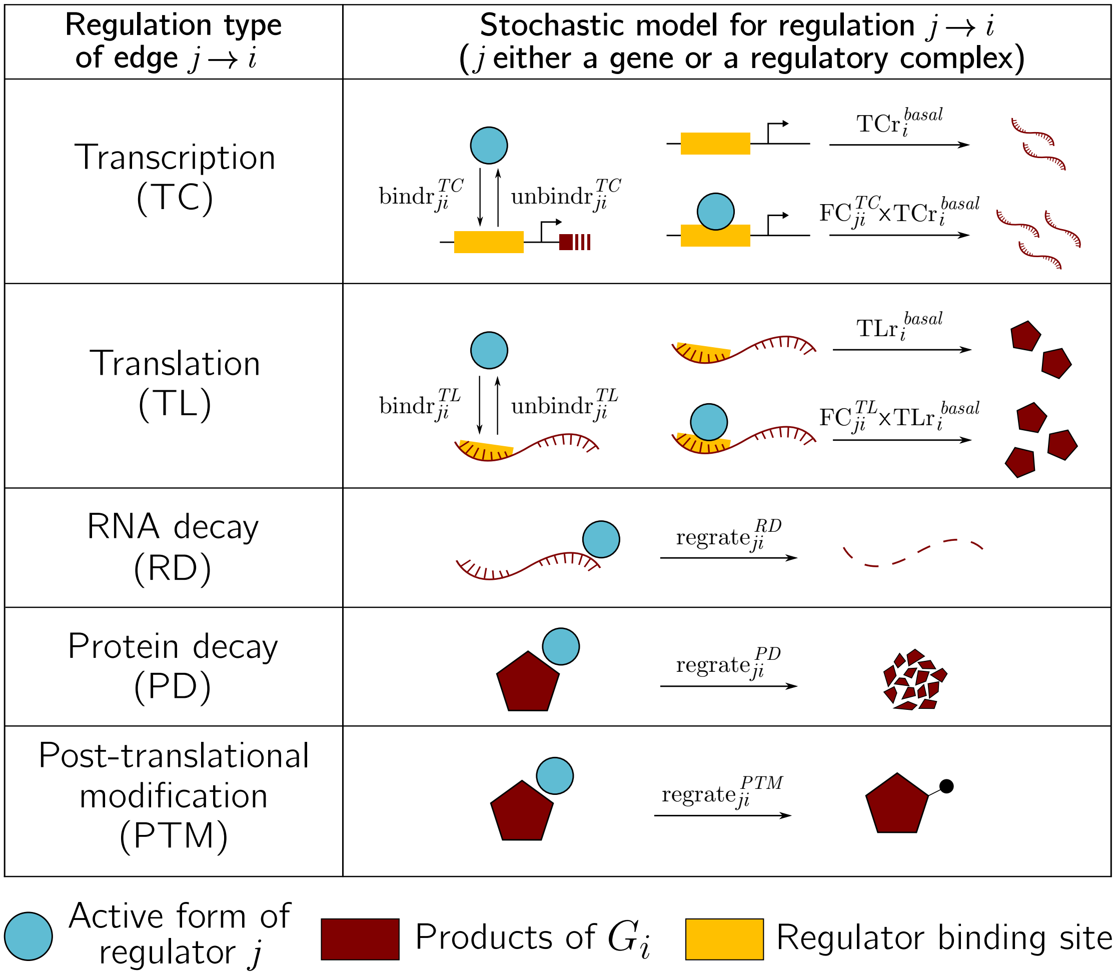

Regulation type of edge $j \rightarrow i$
Stochastic model for regulation $j \rightarrow i$ ($j$ either a gene or a regulatory complex)
Transcription (TC)
$\text{bindr}_{ji}^{TC}$
$\text{unbindr}_{ji}^{TC}$
$\text{TCr}_i^{basal}$
$\text{FC}_{ji}^{TC} \times \text{TCr}_i^{basal}$
Translation (TL)
$\text{bindr}_{ji}^{TL}$
$\text{unbindr}_{ji}^{TL}$
$\text{TLr}_i^{basal}$
$\text{FC}_{ji}^{TL} \times \text{TLr}_i^{basal}$
RNA decay (RD)
$\text{regrate}_{ji}^{RD}$
Protein decay (PD)
$\text{regrate}_{ji}^{PD}$
Post-translational modification (PTM)
$\text{regrate}_{ji}^{PTM}$
Active form of regulator $j$
Products of $G_i$
Regulator binding site

Figure 2.2: Modelling of the different types of expression regulation in the `sismonr` package. Each edge $j \rightarrow i$ in the GRN is transformed into a set of biochemical reactions with associated rates, as presented.

reaction can simply be written:

$$\emptyset \quad \rightarrow \quad [\text{RNA form of gene } i] \tag{2.2}$$

The RNA form of the gene will be discussed later. For an individual carrying the allele $v$ of the gene, the rate of the reaction is:

$$\text{TCr}_i \quad = \quad \text{TCr}_i^{basal} \times \text{qtlTCrate}_i^v \tag{2.3}$$

where $\text{TCr}_i^{basal}$ represents the basal transcription rate of gene $G_i$ as described in the *in silico* system, and $\text{qtlTCrate}_i^v$ represents the QTL effect coefficient of the allele $v$ affecting the gene transcription rate. For this reaction, we do not need to explicitly represent the DNA form of gene $G_i$, as it is present in only one copy throughout the simulation and does not intervene in any other reaction in the stochastic model. Now, let us assume that gene $G_i$ is targeted by a regulator $R_j$ (edge $j \to i$ in the regulatory network), with $R_j$ either a single gene or a regulatory complex. Throughout this section, I use the notation $a_j$ to represent the molecular entity performing the regulation $j \to i$. The actual entity performing the regulation depends on the type of $R_j$ (protein-coding or non-coding gene, or regulatory complex), as presented in Table 2.8. Let us assume that the regulator possesses exactly one binding site in the promoter of gene $G_i$, denoted $\text{Pr}_i\text{\_reg}_j$ with an indication of its free or bound status. The binding and unbinding reactions of the regulator to and from its binding site are:

Table 2.8: Different possible active forms $a_i$ of a regulator $R_j$. It must be noted that if there exists a modified form of the protein encoded by a regulator gene $R_j$, only its modified form $\text{Prot}_j^m$ is active (and not the original form $\text{Prot}_j$).

| Nature of regulator $R_j$ | Form of $a_j$ |
|---|---|
| Non-coding gene | $\text{RNA}_j$ |
| Protein-coding gene | $\text{Prot}_j$ |
| Protein-coding gene targeted by post-translational modification | $\text{Prot}_j^m$ |
| Regulatory complex | $\text{C}_j$ |

$$\text{Pr}_i\text{\_reg}_j\text{\_free} + a_j \rightarrow \text{Pr}_i\text{\_reg}_j\text{\_bound} \tag{2.4}$$

$$\text{Pr}_i\text{\_reg}_j\text{\_bound} \rightarrow \text{Pr}_i\text{\_reg}_j\text{\_free} + a_j \tag{2.5}$$

with rates, for an individual carrying the allele $v$ of gene $G_i$ and the allele $w$ of the regulator $R_j$:

$$\text{bindr}_{ji}^{TC} \times \text{qtlTCregbind}_i^v \times \text{qtlactivity}_j^w \tag{2.6}$$

$$\text{unbindr}_{ji}^{TC} \tag{2.7}$$

with Equation (2.6) corresponding to the rate of Equation (2.4) and Equation (2.7) corresponding to the rate of Equation (2.5). $\text{bindr}_{ji}^{TC}$ and $\text{unbindr}_{ji}^{TC}$ are the binding and unbinding rates associated with the edge $j \to i$ in the GRN. The binding rate of the regulator on its binding site is influenced by $\text{qtlTCregbind}_i^v$, the effect of a mutation on allele $v$ of gene $G_i$ affecting the affinity of its regulatory regions for regulators of transcription, and by $\text{qtlactivity}_j^w$ representing the effect of a mutation on allele $w$ of regulator $R_j$ affecting the activity of its active products. If regulator $R_j$ is a regulatory complex, $\text{qtlactivity}_j$ is obtained by multiplying the QTL effect coefficient `qtlactivity` of each of its components. The unbinding rate of the regulator from its binding site is not affected by genetic mutations. This allows the mutations to affect the occupancy time of the binding site by the regulator, and hence the efficiency of the latter.

The transcription rate of gene $G_i$ depends on whether or not its binding site is occupied by a regulator. If the binding site is free, the transcription occurs at rate $\text{TCr}_i$ (as defined by Equation (2.3)), and follows:

$$\text{Pr}_i\text{\_reg}_j\text{\_free} \quad \rightarrow \quad \text{Pr}_i\text{\_reg}_j\text{\_free} + [\text{RNA form of gene } i] \tag{2.8}$$

If on the contrary a regulator is bound, the following transcription reaction:

$$\text{Pr}_i\text{\_reg}_j\text{\_bound} \quad \rightarrow \quad \text{Pr}_i\text{\_reg}_j\text{\_bound} + [\text{RNA form of gene } i] \tag{2.9}$$

occurs at a rate $\text{TCr}_i \times \text{FC}_{ji}^{TC}$. If regulator $R_j$ is a repressor of gene $G_i$ transcription, then $\text{FC}_{ji}^{TC} = 0$ and the resulting transcription rate when the regulator is bound is zero.

This formalism can be generalised to the case where gene $G_i$ is targeted by a set of $s$ regulators $\{R_1, ..., R_s\}$, again either as single genes or regulatory complexes. In this case the DNA form of the gene is represented by the sum of its binding sites for each regulator, or:

$$\sum_{j=1}^{s} \text{Pr}_i\text{\_reg}_j \tag{2.10}$$

For example if the gene $G_i$ is targeted by regulators $R_1$, $R_2$ and $R_3$, its DNA form is $\text{Pr}_i\text{\_reg}_1 + \text{Pr}_i\text{\_reg}_2 + \text{Pr}_i\text{\_reg}_3$. This representation of the DNA region encoding a gene by the sum of its regulator binding sites is inspired by (Tripathi et al., 2017), who use the same approach in their model. Each binding site can either be in a free or a bound state, so we must consider every combination of free/bound binding sites for the transcription. The general form of the transcription reaction is now:

$$\begin{gathered} \sum_{\substack{\text{unbound} \\ \text{regulators } j}} \text{Pr}_i\text{\_reg}_j\text{\_free} + \sum_{\substack{\text{bound} \\ \text{regulators } k}} \text{Pr}_i\text{\_reg}_k\text{\_bound} \rightarrow \\ \sum_{\substack{\text{unbound} \\ \text{regulators } j}} \text{Pr}_i\text{\_reg}_j\text{\_free} + \sum_{\substack{\text{bound} \\ \text{regulators } k}} \text{Pr}_i\text{\_reg}_k\text{\_bound} + [\text{RNA form of gene } i] \end{gathered} \tag{2.11}$$

with rate $\text{TCr}_i \times \prod_{\substack{\text{bound} \\ \text{regulators } k}} \text{FC}_{ki}^{TC}$. This type of combinatorial control implies that as soon as a repressor is bound to its binding site the transcription rate becomes null.

### RNA and translation

The RNA form of the genes and the translation reactions are modelled similarly to the DNA form and transcription rates presented above. If gene $G_i$ is not targeted by any regulators of translation, then its RNA form is simply $\text{RNA}_i$ and its transcription reaction is:

$$\text{RNA}_i \quad \rightarrow \quad \text{RNA}_i + \text{Prot}_i \tag{2.12}$$

with rate, for an individual carrying the allele $v$ of $G_i$:

$$\text{TLr}_i \;=\; \text{TLr}_i^{basal} \times \text{qtlTLrate}_i^v \tag{2.13}$$

If gene $G_i$ is targeted by a set of $s$ regulators of translation $\{R_1, ..., R_s\}$ (either single genes or regulatory complexes), then the RNA form of the gene is modelled as the sum of its binding sites for the regulators:

$$\sum_{j=1}^{s} \text{RBS}_i\text{\_reg}_j \tag{2.14}$$

It must be noted than the transcription reaction will create each of the binding sites in a free state, so in Equations (2.2) and (2.11) the term [RNA form of gene $i$] can be replaced by $\sum_{l=1}^{u} \text{RBS}_i\text{\_reg}_l\text{\_free}$ if the gene is targeted by a set $\{R_1, ..., R_u\}$ of $u$ regulators of translation, or by RNA$_i$ if the gene is not controlled at the translational level.

The binding and unbinding reactions of each regulator are:

$$\text{RBS}_i\text{\_reg}_j\text{\_free} + a_j \;\rightarrow\; \text{RBS}_i\text{\_reg}_j\text{\_bound} \tag{2.15}$$

$$\text{RBS}_i\text{\_reg}_j\text{\_bound} \;\rightarrow\; \text{RBS}_i\text{\_reg}_j\text{\_free} + a_j \tag{2.16}$$

with rates, for an individual carrying the allele $v$ of gene $G_i$ and the allele $w$ of regulator $R_j$:

$$\text{bindr}_{ji}^{TL} \times \text{qtlTLregbind}_i^v \times \text{qtlactivity}_j^w \tag{2.17}$$

$$\text{unbindr}_{ji}^{TL} \tag{2.18}$$

with Equation (2.17) corresponding to the rate of Equation (2.15) and Equation (2.18) corresponding to the rate of Equation (2.16). The general form of the translation reaction is:

$$\sum_{\substack{\text{unbound}\\ \text{regulators } j}} \text{RBS}_i\text{\_reg}_j\text{\_free} + \sum_{\substack{\text{bound}\\ \text{regulators } k}} \text{RBS}_i\text{\_reg}_k\text{\_bound} \;\rightarrow \tag{2.19}$$

$$\sum_{\substack{\text{unbound}\\ \text{regulators } j}} \text{RBS}_i\text{\_reg}_j\text{\_free} + \sum_{\substack{\text{bound}\\ \text{regulators } k}} \text{RBS}_i\text{\_reg}_k\text{\_bound} + \text{Prot}_i \tag{2.20}$$

with associated rate $\text{TLr}_i \times \prod_{\substack{\text{bound}\\ \text{regulators } k}} \text{FC}_{ki}^{TL}$, TLr$_i$ being defined in Equation (2.13).

### Protein post-translational modification

In the GRN, a protein-coding gene $G_i$ can be targeted by a set of regulators of post-translational modification; each activator regulator $R_j$ for $G_i$ (i.e. for which the edge $j \rightarrow i$ is assigned a plus sign) triggers the modification of the protein according to:

$$a_j \ + \text{Prot}_i \ \ \rightarrow \ \ a_j \ + \ \text{Prot}_i^m \tag{2.21}$$

with rate $\text{regrate}_{ji}^{PTM} \times \text{qtlPTMregrate}_i^v \times \text{qtlactivity}_j^w$. Accordingly, regulators $R_j$ exerting a negative regulation (i.e. for which the edge $j \rightarrow i$ is assigned a minus sign) will trigger the transformation of the protein back into its original form:

$$a_j \ + \text{Prot}_i^m \ \ \rightarrow \ \ a_j \ + \ \text{Prot}_i \tag{2.22}$$

with the same rate $\text{regrate}_{ji}^{PTM} \times \text{qtlPTMregrate}_i^v \times \text{qtlactivity}_j^w$. It must be noted that during the initial translation of gene $G_i$ the created protein is in its original form, $\text{Prot}_i$, and never in its modified form.

### RNA and protein decay

With the exception of DNA, each molecule in the system has a finite lifetime and can therefore decay naturally. The natural decay of RNAs can be written:

$$\text{RNA}_i \ \ \rightarrow \ \ \emptyset \tag{2.23}$$

or:

$$\sum_{j=1}^{s} \text{RBS}_i\text{\_reg}_j \ \ \rightarrow \ \ \emptyset \tag{2.24}$$

depending on the RNA form of the gene, with rate:

$$\text{RDr}_i^{basal} \ \times \ \text{qtlRDrate}_i^v \tag{2.25}$$

Note that in Equation (2.24), the different RNA binding sites can be in different states, i.e. either free or bound by a regulator. `sismonr` hence generates a decay reaction for each possible combination of the different binding sites in all possible states, all with the same rate, as described in Equation (2.25).

The same modelling applies for the natural decay of proteins. For proteins targeted by post-translational modifications, I consider that the original and modified forms of the protein have the same lifetime, hence:

$$\text{Prot}_i \rightarrow \emptyset \tag{2.26}$$

$$\text{Prot}_i^m \rightarrow \emptyset \tag{2.27}$$

both occur at the same rate:

$$\text{PDr}_i^{basal} \times \text{qtlPDrate}_i^v \tag{2.28}$$

The decay of RNAs and proteins can also be regulated. If gene $G_i$ is targeted by a set $\{R_1, ..., R_s\}$ of regulators of RNA decay, then for $1 \leq j \leq s$ the reaction:

$$a_j + [\text{RNA form of gene } i] \rightarrow a_j \tag{2.29}$$

with [RNA form of gene $i$] equal to either RNA$_i$ or $\sum_j$ RBS$_i$_reg$_j$_free, occurs at rate:

$$\text{regrate}_{ji}^{RD} \times \text{qtlRDregrate}_i^v \times \text{qtlactivity}_j^w \tag{2.30}$$

If the gene is targeted by a set $\{R_1, ..., R_s\}$ of regulators of protein decay, then for $1 \leq j \leq s$ the reaction:

$$a_j + \text{Prot}_i \rightarrow a_j \tag{2.31}$$

$$a_j + \text{Prot}_i^m \rightarrow a_j \tag{2.32}$$

occurs both at rate:

$$\text{regrate}_{ji}^{PD} \times \text{qtlPDregrate}_i^v \times \text{qtlactivity}_j^w \tag{2.33}$$

It must be noted that the natural and regulated decay of molecules occur even if the latter is bound to a binding site or in complex. In the first case, the molecule decays and the regulator returns to its free form. In the second case, the other components of the complex are released.

#### Regulatory complex formation

As mentioned previously, the products of some genes can assemble into a regulatory complex that controls the expression of its target genes. Let us consider a complex $C_i$ composed of the products of

$u$ genes $\{G_1, ..., G_u\}$. Its formation and dissociation reactions are:

$$\sum_{j=1}^{u} [\text{active product of gene } j] \quad \rightarrow \quad C_i \tag{2.34}$$

$$C_i \quad \rightarrow \quad \sum_{j=1}^{u} [\text{active product of gene } j] \tag{2.35}$$

with rates formr$_i$ and dissr$_i$, respectively. The form of the active product of each gene can be found in Table 2.8.

### Initial abundances

An initial abundance is assigned to each entity in the stochastic system that corresponds to the number of copies of these entities at the beginning of the simulation. I will use the notation $[X]_0$ to represent the initial abundance of entity $X$. The following rules are applied:

- The different promoter binding sites of the genes are present in one copy in the system and are in a free state, i.e.:

$$\forall i, j \quad \left[\text{Pr}_i\text{\_reg}_j\text{\_free}\right]_0 = 1, \quad \left[\text{Pr}_i\text{\_reg}_j\text{\_bound}\right]_0 = 0 \tag{2.36}$$

- For each gene, the biosynthesis of RNAs has reached a steady state (ignoring any regulation), so:

$$\forall i, \quad [\text{RNA}_i]_0 = \frac{\text{TCr}_i^{basal}}{\text{RDr}_i^{basal}} \tag{2.37}$$

or, if the RNA of gene $i$ is on the form $\sum\limits_j \text{RBS}_i\text{\_reg}_j$, assume that all regulator binding sites are in a free state:

$$\forall j, \quad \left[\text{RBS}_i\text{\_reg}_j\text{\_free}\right]_0 = \frac{\text{TCr}_i^{basal}}{\text{RDr}_i^{basal}}, \quad \left[\text{RBS}_i\text{\_reg}_j\text{\_bound}\right]_0 = 0 \tag{2.38}$$

- For each gene, the biosynthesis of proteins has reached a steady state (ignoring any regulation), so:

$$\forall i, \quad [\text{Prot}_i]_0 = [\text{RNA}_i]_0 \times \frac{\text{TLr}_i^{basal}}{\text{PDr}_i^{basal}} \tag{2.39}$$

and all existing proteins are in their original form, so for all genes $i$ for which Prot$_i^m$ exists:

$$[\text{Prot}_i^m]_0 = 0 \tag{2.40}$$

- No regulatory complex has been formed, so for all complexes $j$:

$$[C_j]_0 = 0 \tag{2.41}$$

The fact that the RNAs and proteins of all genes are present at a steady-state level ignoring any regulation implies that the simulation mimics the sudden activation of the regulatory network as a response to a stimulus.

### Modelling ploidy

If the ploidy of the system is set to $P$, each gene is present in $P$ copies or homologs in the system. I denote the $k$-th homolog with the suffix GCN$_k$, and use this notation to differentiate the homolog of origin of the different gene products. The products of the $k$-th homolog of gene $G_i$ will thus be:

$$\text{DNA:} \quad \sum_j \text{Pr}_i\text{\_GCN}_k\text{\_reg}_j \tag{2.42}$$

$$\text{RNA:} \quad \text{RNA}_i\text{\_GCN}_k \quad \text{or} \quad \sum_j \text{RBS}_i\text{\_GCN}_k\text{\_reg}_j \tag{2.43}$$

$$\text{Protein:} \quad \text{Prot}_i\text{\_GCN}_k \quad \text{and if applicable} \quad \text{Prot}_i^m\text{\_GCN}_k \tag{2.44}$$

This is essential as the different homologs can be different alleles of the gene and hence possess different mutations that affect their kinetic properties.

Let the protein-coding gene $G_j$ control the transcription of gene $G_i$. There is still only one binding site for the regulator's active products on the promoter of each homolog of gene $G_i$, but the different versions of the regulator (arising from the different homologs of the regulator gene) can all bind to these binding sites. The entity representing the bound binding site must carry the information about which version of the regulator is bound, in order to correctly model unbinding reactions. The binding and unbinding reactions of the protein originating from the $l$-th homolog of gene $G_j$ ($1 \leq l \leq P$) to and from its binding site on the $k$-th homolog of gene $G_i$ ($1 \leq k \leq P$) are:

$$\text{Pr}_i\text{\_GCN}_k\text{\_reg}_j\text{\_free} + \text{Prot}_j\text{\_GCN}_l \rightarrow \text{Pr}_i\text{\_GCN}_k\text{\_reg}_j\text{\_GCN}_l\text{\_bound} \tag{2.45}$$

$$\text{Pr}_i\text{\_GCN}_k\text{\_reg}_j\text{\_GCN}_l\text{\_bound} \rightarrow \text{Pr}_i\text{\_GCN}_k\text{\_reg}_j\text{\_free} + \text{Prot}_j\text{\_GCN}_l \tag{2.46}$$

Hence, for a ploidy of $P$, the number of binding/unbinding reactions sets to generate for the edge $i \rightarrow j$ in the GRN is $P^2$ ($P$ versions of the regulator can bind to each of the $P$ homologs of the target). The same reasoning can be applied to regulatory complex formation, where each of the versions of the components can form a complex. The number of formation/dissociation reactions sets for a complex of size $c$ is $P^c$.

### 2.4.2 Stochastic simulation

The stochastic system generated for each *in silico* individual is used to simulate the evolution over time of the abundance of the RNAs and proteins encoded by each gene. To do so, I use an existing Julia implementation of the Stochastic Simulation Algorithm (SSA, Gillespie (2007)) available in the module `BioSimulator` (Landeros et al., 2018). A more detailed description of the principle of stochastic simulation is available in (Gillespie, 2007), but I briefly present here the main steps of the algorithm.

For each reaction of the system, it is possible to compute its propensity, that is the probability of the reaction to occur in the next time step. Its value depends on the rate of the reaction as well as the abundance of the different reactants. It must be noted that the SSA makes the assumption that the system is homogeneous, that is, there is no specific spatial distribution of the molecules in the system. At a given time-point $t$, the SSA samples from an exponential distribution the interval of time $\tau$ before the next reaction occurs. The next reaction to be "fired" is also randomly sampled, according to the propensities of the reactions. The system state (i.e. the abundance of the different molecules) is updated according to the stoichiometry of the chosen reaction, and the time is incremented by $\tau$. This so-called direct method simulates each individual reaction firing in the system. Other exact implementations of the SSA (simulating each individual reaction) have been proposed. In addition, approximate methods have been implemented. They trade accuracy of the simulation for efficiency, by means of simplifying assumptions. The `BioSimulator` package implements several versions of the SSA, including exact and approximate approaches. The current methods implemented are the Direct method (Gillespie, 1977), the First Reaction method (Gillespie, 1976), the Next Reaction method (Gibson & Bruck, 2000), the Optimised Direct method (Cao et al., 2004), the Tau-Leaping method (Gillespie & Petzold, 2003) and the Step Anticipation Tau-Leaping method (Sehl et al., 2009). A description of each method is provided in (Landeros et al., 2018). Each of these methods can be called by `sismonr` for the system simulation.

## 2.5 Comparison of `sismonr` with existing tools

I aim at comparing `sismonr` with existing tools with similar objectives. Consequently, I review here existing algorithms that generate artificial *in silico* gene regulatory networks (GRNs) and simulate the expression profiles of their genes using a mathematical model of the system. Tools that only simulate (deterministically or stochastically) a given mathematical model, e.g. different implementations of the Stochastic Simulation Algorithm, are thus not considered here.

### 2.5.1 Existing GRN simulators

#### SynTReN

`SynTReN` (Bulcke et al., 2006) is a Java application that simulates steady state RNA abundances for genes in subnetworks sampled from a source network. The transcriptional networks of *Saccharomyces cerevisiae* and *Escherichia coli* are available as source networks; users can also provide their own. The user can set the desired number of genes in the extracted subnetwork, and can also add a number of background genes, i.e. genes not involved in the network. Two-regulators interactions are possible in the network. There is no visualisation feature in the application, but the extracted network is saved both as a *sif* and an *xml* file, the latter describing the value of the different kinetic parameters used for the simulation.

The translation of genes is not modelled; instead the mRNA level of a given regulator gene is used a a proxy for the abundance of its proteins. Gene expression is modelled deterministically, and Michaelis-Menten and Hill equations are used to model gene transcription regulation. It is possible to simulate different experiments representing different external conditions: a set of source genes (i.e. genes without regulatory input) is randomly chosen, and their expression levels is set to a different value for each experiment. Several simulations or samples can be performed for each experiment. Normalised RNA concentrations are simulated at the steady state, and it is not possible to obtain time series data.

#### SGNSim

`SGNSim` (Ribeiro & Lloyd-Price, 2007) is a command-line tool composed of two algorithms: `sgne`, a generator of systems of reactions describing GRNs, and `sgns`, a Stochastic Simulation Algorithm implementation optimised for incorporating time delays in the simulated reactions. The `sgne` program is used to generate regulatory networks of desired size. Users can choose between two possible topology models when generating a system, or alternatively can provide their own topology as a file indicating the target-regulator pairs. Combinatorial regulation is allowed, as regulators can form homodimers or heterodimers that control the target's transcription. Other parameters, such as the sampling distributions of the different rates (transcription, translation) or initial concentrations, can be specified in the form of an input file. There is no default value defined for the parameters but the software manual includes several examples to guide the user. The output of this step is a file containing the necessary information to simulate the system: a list of species in the system with their initial abundance, and a list of reactions with associated rates and time delays. In a second step, the `sgns` program takes as input the output of `sgne`, and runs a stochastic simulator algorithm to simulate the abundance of the different molecules over time. The result of the simulation is saved in a tab-delimited file.

One of the key strengths of `SGNSim` is its ability to simulate time delays in the transcription and translation reactions. This notably permits to simulate different cellular compartments. Also, the sampling distributions for the different parameters necessary to the generation of the system can be defined by the user, in the limit of implemented distributions. It is to note however that the generated network representing the regulation among genes is not available to the user, which impairs the user's ability to compare any network inference result to the true GRN.

### GeneNetWeaver

`GeneNetWeaver` (Schaffter et al., 2011) is a Java-implemented software for GRN generation and simulation. Networks are generated by sampling subnetworks of desired size from source networks, i.e. known transcriptional regulation networks from model organisms. The transcriptional networks of *Escherichia coli* and *Saccharomyces cerevisiae* are available as source networks. Alternatively, users can provide their own network structure. The software includes a network visualisation feature.Once the network is defined, a kinetic model is generated. To the best of my knowledge, the user has no control over the generation of this kinetic model and hence cannot specify the distributions from which values for the different kinetic parameters are sampled. It is not clear from the user manual, nor from the associated publication what distributions are used for the different reaction rates, what their biological justification is or which mathematical model is been used. Combinatorial regulation is possible, but again the exact formalism is not presented. Then, the user can simulate different experiments from the network:

- wild-type data, which corresponds to simulating the system as is;
- knockouts or knockdowns, which correspond to partial and complete gene transcription reduction, respectively;
- Multifactorial perturbation, which amounts to adding small perturbations to the transcription rates of the different genes, in order to mimic genotypic variability.

It is possible to obtain time series or steady state data for each of these experiments. The system can be simulated deterministically, through the use of differential equations, or semi-stochastically, through the use of chemical Langevin equations. The latter are equations representing the evolution of the abundance of the different molecules over time. The first part of such equation is identical to their deterministic counterpart, but a noise term is added, proportional to the square root of the deterministic term. The user can control the level of noise in this case. The use of deterministic or semi-deterministic model for the simulation enables the software to generate data for large networks in only a few seconds. After the simulation, it is possible to add noise to the results of the simulation in order to mimic experimental noise. In particular, `GeneNetWeaver` integrates a model of microarrays-like noise.

### sgnesR

`sgnesR` (Tripathi et al., 2017) is an R package providing an R interface for the `SGNSim` simulator. It aims at simulating stochastic time series RNA and protein profiles for genes interacting through a transcriptional regulatory network, accounting for delays in the transcription and translation reactions. In a first step, the user has to provide a graph structure representing the transcriptional regulation among genes. The package does not provide a synthetic GRN generator. The user then sets the different kinetic parameters of the system (transcription, translation, RNA and protein decay, protein binding and unbinding rates). It must be noted that, for each of these parameters, a unique value is chosen and applied to all genes. There is hence no possibility to simulate genes with different transcription rates, for example. Reaction rates functions can also be specified, i.e. allowing the rate of a reaction to vary non-linearly as a function of the abundance of some other species. The user can also specify the delays of the different reactions. No values are provided by default. Alternatively, the user can construct the list of biochemical reactions defining the system by adding the reactants and products of each reaction one by one to the system. This method of constructing *in silico* systems is outside the scope of this review, since it amounts to directly simulating a system of reactions rather than using a generator of GRN. In a second step, the system is simulated stochastically for a defined amount of (simulated) time, and molecules abundance are read at regular intervals (specified by the user) during the simulation. This is achieved using the stochastic simulator algorithm of `SGNSim` that accounts for delays in the reactions.

A major limitation of the package is the lack of flexibility in terms of kinetic parameters. As mentioned above, all genes are assigned the same transcription rate, translation rate, etc., which limits the biological relevance of the simulated system. Furthermore, `sgnesR` performs only one simulation of a system at a time. There is no way to generate time series with environmental or genotypic perturbations, which makes it impractical to generate benchmarks for gene network reconstruction.

### Other tools not considered for the comparison

Other tools with similar objectives have been proposed, but are no longer available/supported, and consequently were not included in the comparison:

- `A-BIOCHEM` (Mendes et al., 2003): this simulator uses a deterministic representation of the RNAs' concentration, and transcription regulation is represented by Hill functions. The translation of genes is not modelled. This software is not available publicly, and thus cannot be used for the comparison.

- `RENCO` (Roy et al., 2008): `RENCO` is a C++ command line program for generating differential equations to simulate protein-protein and transcriptional regulation networks, allowing to model combinatorial regulation of transcription. An important point is that `RENCO` does not perform the simulation of the system; instead, the system of differential equations representing the system is saved as a SBML format, which can be used by other software to simulate the system.

- `NETSim` (Di Camillo et al., 2009): this R package generates *in silico* gene regulatory networks using a hierarchical modular topology model, and simulates deterministic RNA and protein time series. Combinatorial regulation of transcription is modelled via fuzzy logic. However, this R package is not maintained and could not be installed.
- `GRENDEL` (Haynes & Brent, 2009): `GRENDEL` is a command line tool for generating GRNs incorporating time-dependent experimental stimuli impacting genes transcription. The kinetic parameters for the deterministic model are sampled from values measured experimentally in yeast. It is to note that the software is not maintained and could not be compiled.
- `GeNGe` (Hache, Wierling, et al., 2009): `GeNGe` is a web application for generating and simulating synthetic GRNS, using deterministic equations. Transcription regulation is modelled via non-linear kinetics, that allow for combinatorial regulation of a common target by different regulators. The simulator allows the user to model local or global perturbations of the system. This web application is however not available any more.
- `SysGenSIM` (Pinna et al., 2011): this GRN simulator allows to generate synthetic systems genetic data, as it explicitly models eQTLs, that are genetic mutations affecting the different properties of the genes. The tool is implemented in Matlab, a proprietary software. Moreover, it has a dependency towards Matlab's Bioinformatics toolbox. In consequence, I was not able to use it for comparison with `sismonr`.

### 2.5.2 Comparison of `sismonr` and `sgnesR` Stochastic Simulation Algorithm implementations

I show that the results obtained with `sismonr` are reproducible with any other implementation of the Stochastic Simulation Algorithm. As the `sgnesR` package allows to simulate a predefined set of biochemical reactions with the `SGNSim` stochastic simulator, I used it to simulate the behaviour of an *in silico* system generated with `sismonr` (see Figure 2.3, the different rates of production and decay used for each gene are presented in Supplementary Table B.1). I thus input the same set of biochemical reactions with identical rates and initial abundances in both simulators. This does not compare the GRN and system generator of the two packages but only their stochastic simulator algorithms: the Julia BioSimulator module (Landeros et al., 2018) for `sismonr` and the `SGNSim` simulator for `sgnesR`. As the simulations are stochastic, identical results are not expected, but the profiles of molecules abundance should be similar. I confirm that for an identical set of biochemical reactions, both packages produce very similar results, as can be seen in Figure 2.4 and Figure 2.5. The simulated expression of the genes over 100 simulation runs for `sismonr` can be found in Supplementary Figure B.1, and one of the simulation runs is depicted in Supplementary Figure B.2.

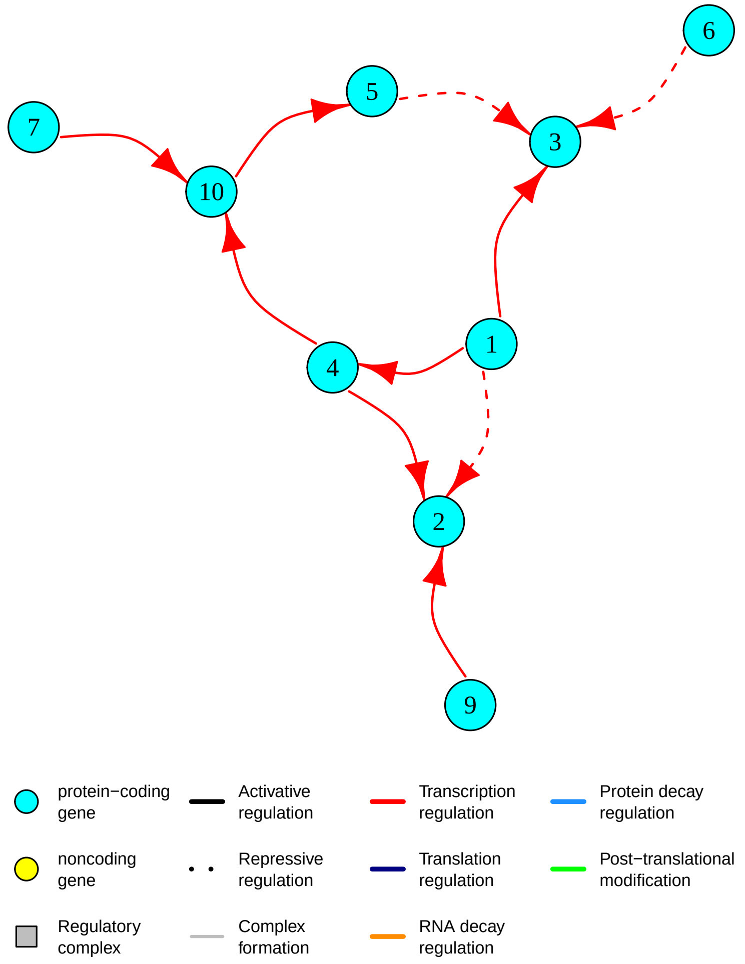


Figure 2.3: Transcription regulatory network used for the comparison of `sismonr` and `sgnesR` simulations. The system has 10 protein-coding genes, linked through transcription regulations as depicted. Solid lines represent activating regulations, dashed lines represent repressing regulations. The plot is produced with the `plotGRN()` function of `sismonr`.

### 2.5.3 Comparison of the simulations of a same regulatory network

In this section, I compare different aspects of the genes profiles simulated by `sismonr` and different existing simulators using their default settings. Namely, I compare `sismonr` to the Java application `SynTren` (Bulcke et al., 2006), the command-line tool `SGNSim` (Ribeiro & Lloyd-Price, 2007), the software `GeneNetWeaver` (Schaffter et al., 2011) and the R package `sgnesR` (Tripathi et al., 2017). To facilitate the comparison, I used the same network topology across the simulators (i.e. same edges and types of regulatory interaction, that is activation or repression, see Figure 2.6). The other parameters are kept to their default values as provided for each simulator. It must be noted that the R package `sgnesR` does not provide default values for the different kinetic parameters. I instead use the same values as in the example presented in their publication. Similarly, `SGNSim` does not come with a default parameter file; I used instead the values provided as example in the `sgne` manual. For

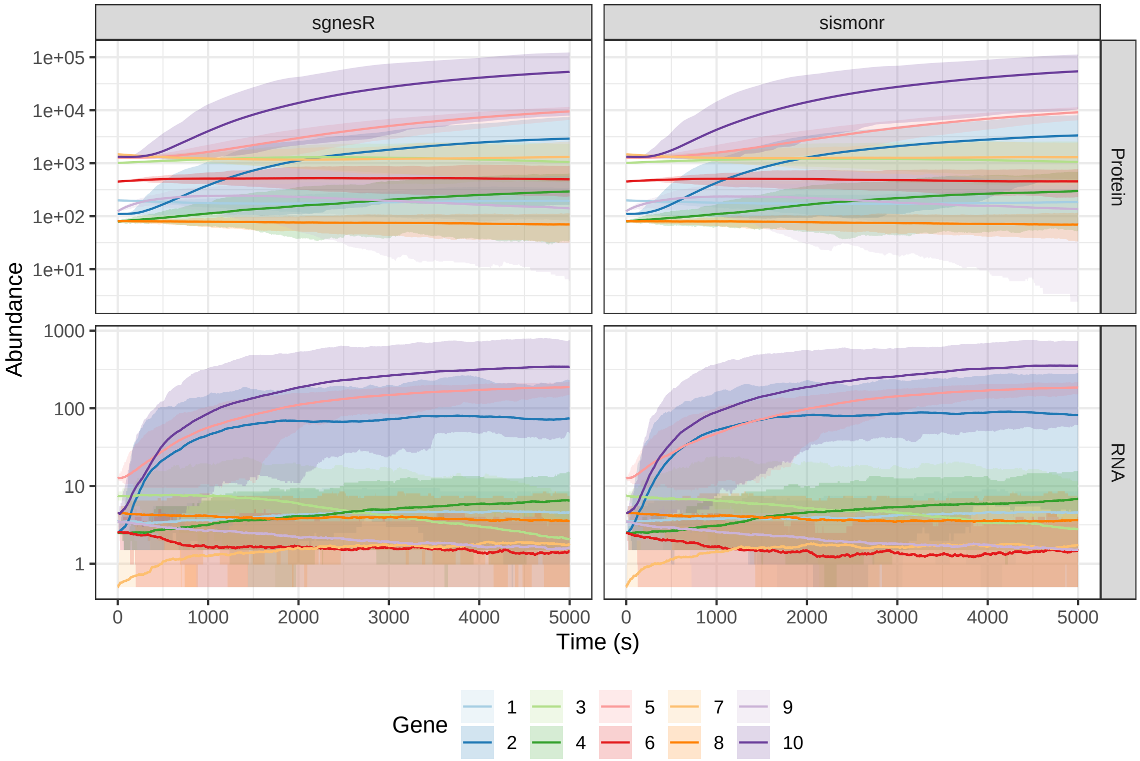


Figure 2.4: RNA (bottom panels) and protein (top panels) abundance of the genes (one colour per gene) over time as generated by the stochastic simulation algorithm of `sgnesR` (left panels) and `sismonr` (right pannels). The simulation was reproduced 100 times for each simulator. The solid lines represent the mean molecules abundance over the 100 simulations at each time-point. The coloured areas represent the 2.5% and 97.5% quantiles of the molecules abundance over the 100 simulations at each time-point. For visualisation purpose, an offset of 0.5 was added to the abundance of each molecule at each time-point for the 100 simulations of each simulator before plotting.

`sismonr`, I set the ploidy of the system to one (haploid situation) to make the results comparable with those of other simulators. I note that, for the sake of fairness, I can only compare features of `sismonr` that are also present with the other simulators. Novel features such as post-transcriptional regulation or ploidy of the system cannot directly be compared with existing simulators, and will instead be discussed in Section 2.6.

I produced 100 simulations of the same network with each simulator. For `SGNSim` and `sgnesR`, it amounts to repeating the simulation of the same set of equations, as they do not allow to generate perturbations of the system. For `SynTReN`, I generated 100 experiments, with one sample per experiment. Each experiment corresponds to setting the initial abundance of a few source genes (i.e. without regulatory input) to a different random value. With `GeneNetWeaver`, I simulated 100 multifactorial perturbations. Each multifactorial perturbation corresponds to modifying the transcription rate of all

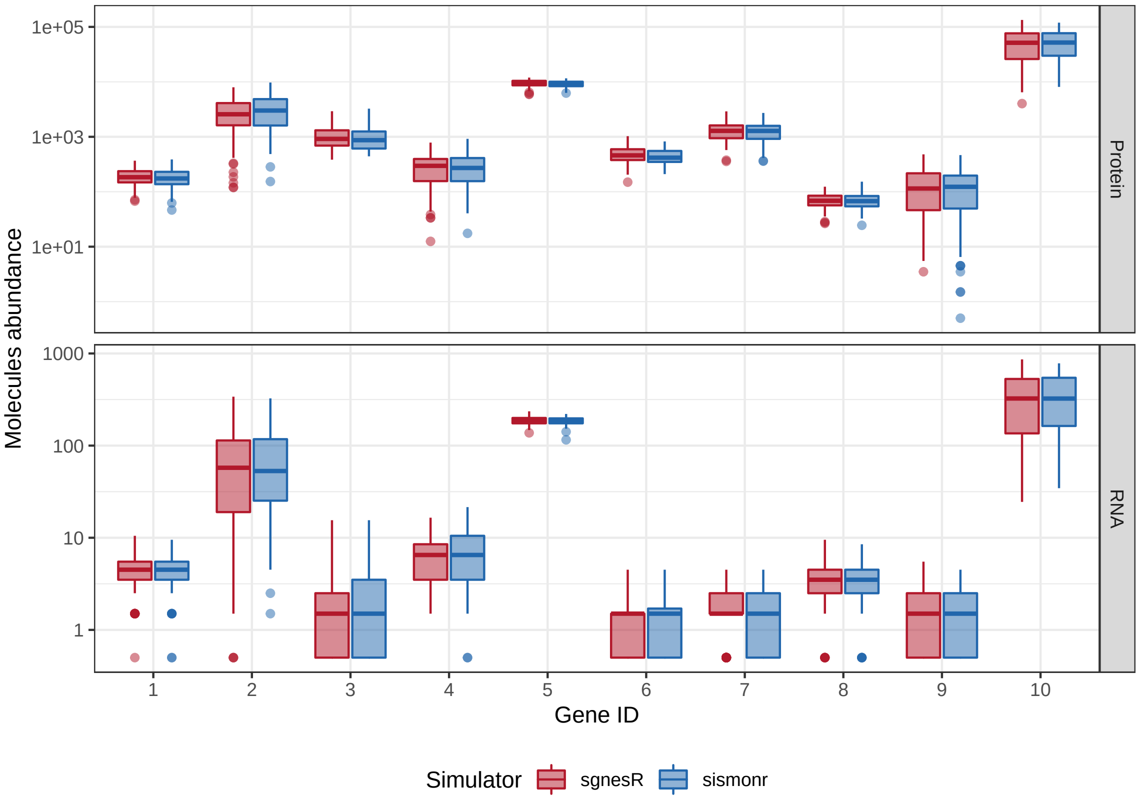


Figure 2.5: Distribution of the RNA (bottom pannel) and protein (top pannel) abundance of the genes at t = 5000s of the simulation over 100 simulations of the system with the `sgnesR` simulator (in red) and `sismonr` simulator (in blue). For visualisation purpose, an offset of 0.5 was added to the abundance of each molecule at each time-point for the 100 simulations of each simulator before plotting.

genes by a small random amount. With `sismonr`, I generated a population of 100 individuals, each carrying one of five possible allele of each gene.

### RNA and protein abundances

Figure 2.7 depicts the RNA and protein levels of each gene at the last time-point of the simulations for each simulator. For stochastic simulators, namely `sismonr`, `sgnesR` and `SGNSim`, the values obtained correspond to the absolute abundance of the RNA and proteins. For both deterministic simulators, `GeneNetWeaver` and `SynTReN`, the values obtained correspond to normalised concentrations instead. As `SynTReN` does not model the translation of genes, protein concentrations are not available.

First, `sismonr`, `GeneNetWeaver` and `SynTReN` simulate genes with varied RNA and protein levels, contrary to `sgnesR` and `SGNSim`. `sismonr` is the only simulator that uncouples transcription and translation of the genes. For other simulators, genes with similar RNA levels also exhibit similar protein values. Second, protein levels simulated with `sismonr` are a few orders of magnitude higher

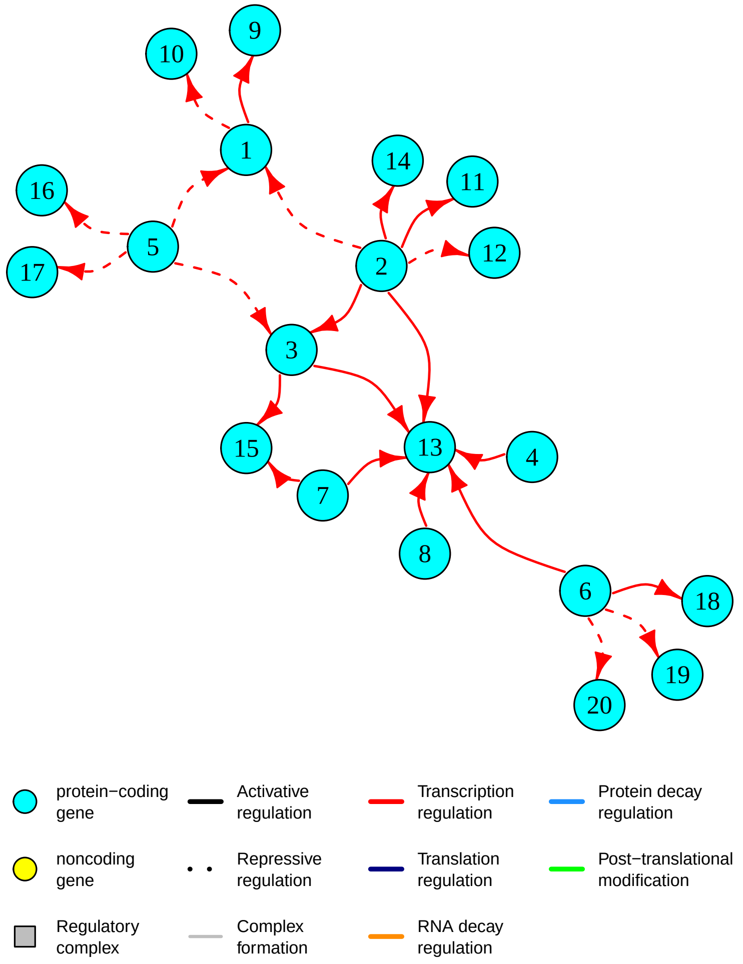


Figure 2.6: Transcription regulatory network used for the comparison of the different simulators. The system has 20 protein-coding genes, linked through transcription regulations as depicted. Solid lines represent activating regulations, dashed lines represent repressing regulations. The plot is produced with the `plotGRN()` function of `sismonr`.

than corresponding RNA levels. This is in better agreement with experimental values observed in yeast (Milo & Phillips, 2016) than the similar levels of RNA and proteins displayed by stochastic simulators (`sgnesR` and `SGNSim`). These two points are also illustrated in the previous simulation example in Section 2.5.2 (see notably Supplementary Figure B.1). Deterministic simulators (`GeneNetWeaver` and `SynTReN`) only provide relative concentrations.

### RNA and protein correlation

This paragraph is concerned with the correlation between the RNA and protein levels of each gene. I exclude from this comparison `SynTReN`, as it does not model proteins. In Figure 2.8, I represent the distribution of RNA-protein Pearson correlation coefficients across the 20 genes for each simulator. The mean RNA-protein correlation for `GeneNetWeaver`, `sismonr`, `sgnesR` and `SGNSim` are 0.68,

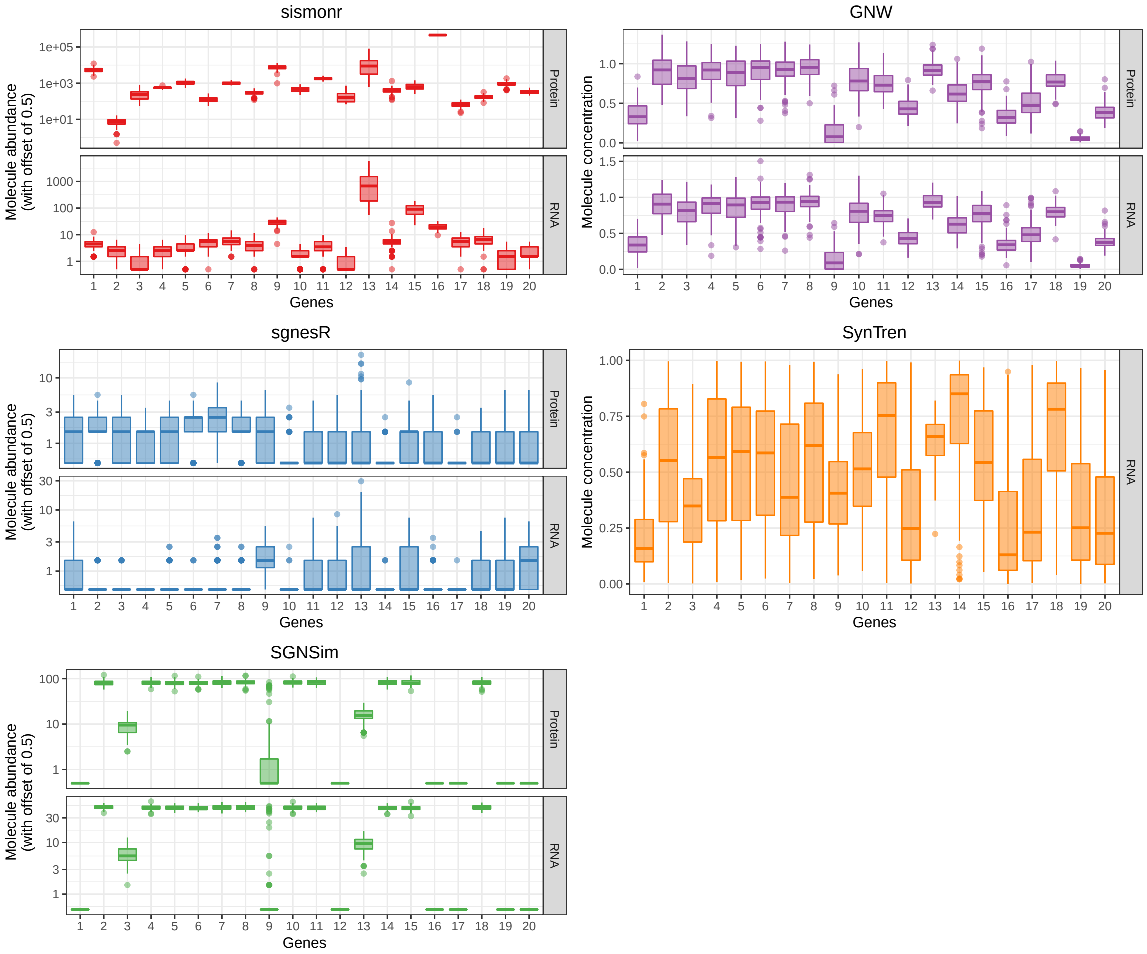


Figure 2.7: Distribution of the RNA and protein abundance (left column, stochastic simulators) or relative concentration (right column, deteriministic simulators) of the different genes at t = 2000s of the 100 simulations of the system with the different simulators. Molecules absolute abundance (left column) are plotted on a log10 scale, while normalised concentrations are plotted on a non-transformed scale. An offset of 0.5 was added to all abundance values in the results of all stochastic simulators (`sismonr`, `sgnesR` and `SGNSim`) for better visualisation.

0.63, 0.32 and 0.33, respectively. These results can be explained. For `sgnesR` and `SGNSim`, the observed variation in the data stems from the stochastic noise representing biological variability, which reduces the correlation between mRNA and the corresponding protein abundance across the simulations. On the contrary, `sismonr` and `GeneNetWeaver` include biological noise and genetic variability between the different simulations. The genetically induced variation in kinetic parameters extends the range of RNA simulated abundance. The amplitude of the biological noise does not depend on the simulated genetic mutations. In other words, within a genetic background, the RNA-protein correlation is the same for `sismonr` as for `sgnesR` and `SGNSim`. For `GeneNetWeaver`, all simulated individuals are genetically unique and the population does not reflect any structure. I illustrate this for one gene in Figure 2.9. Note that the RNA-protein correlations obtained are optimistic compared to what can be observed experimentally. This is due to the absence of any post-transcriptional regulation in these simulations. Contrary to other simulators, `sismonr` can model such regulations and thus provide more realistic RNA-protein correlations. Lastly, it can be observed in the simulations that, as expected, the level of correlation between a regulator abundance and its targets' level is higher with the fully deterministic simulator `SynTReN` (mean correlation overall regulations of 0.72) and smaller for fully stochastic simulators (mean correlation overall regulations of 0.13, 0.1 and 0.09 for `sgnesR`, `sismonr` and `SGNSim`, respectively), with `GeneNetWeaver` ranking in between those (mean correlation overall regulations of 0.33). This is expected as stochastic simulation introduces noise in the abundance of the molecules and can cloud the regulator-target relationship.

### Running time

For this example, `sismonr` generated the simulations in under five minutes, with an average of 2.85 seconds per simulation of an *in silico* individual. The other simulators took under a minute to generate

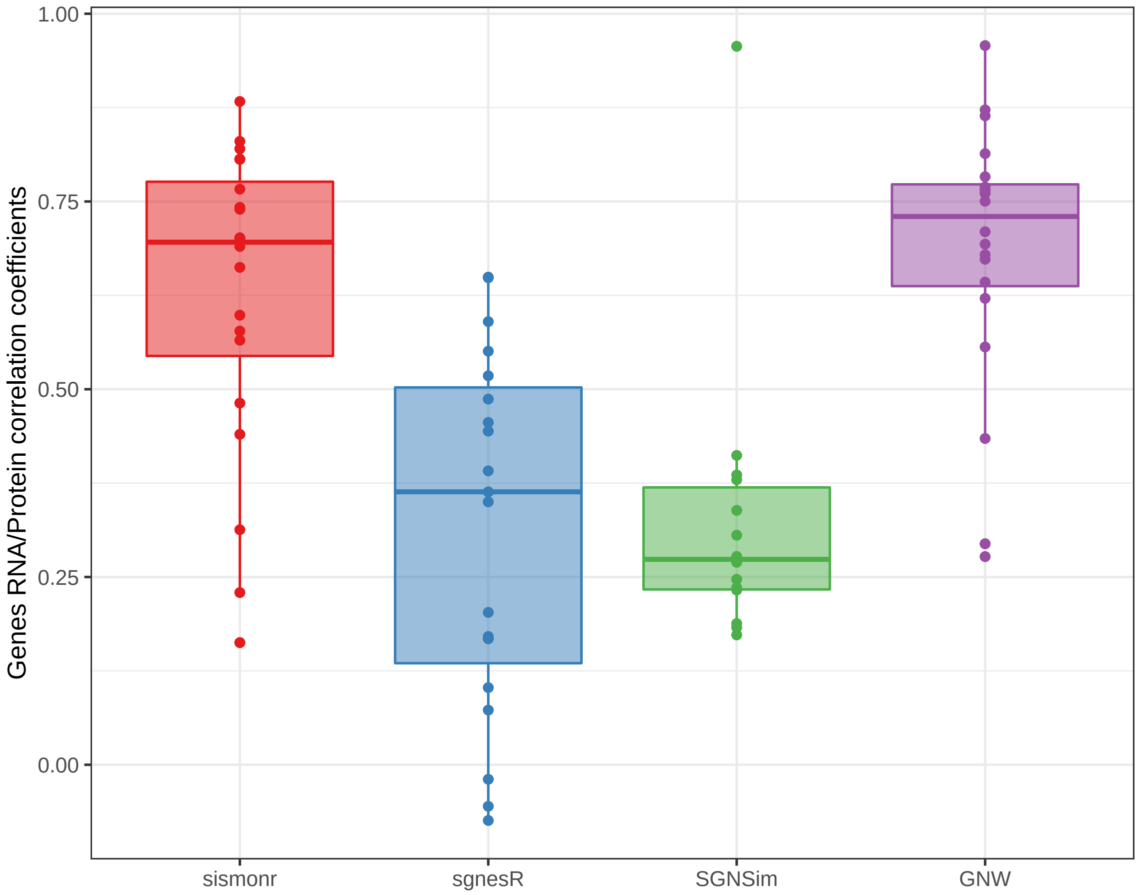


Figure 2.8: Distribution of the correlation coefficient between RNA and protein levels of the different genes in the system for each simulator. These correlations were computed at the last time-point of the simulations, across 100 simulations for each simulator.

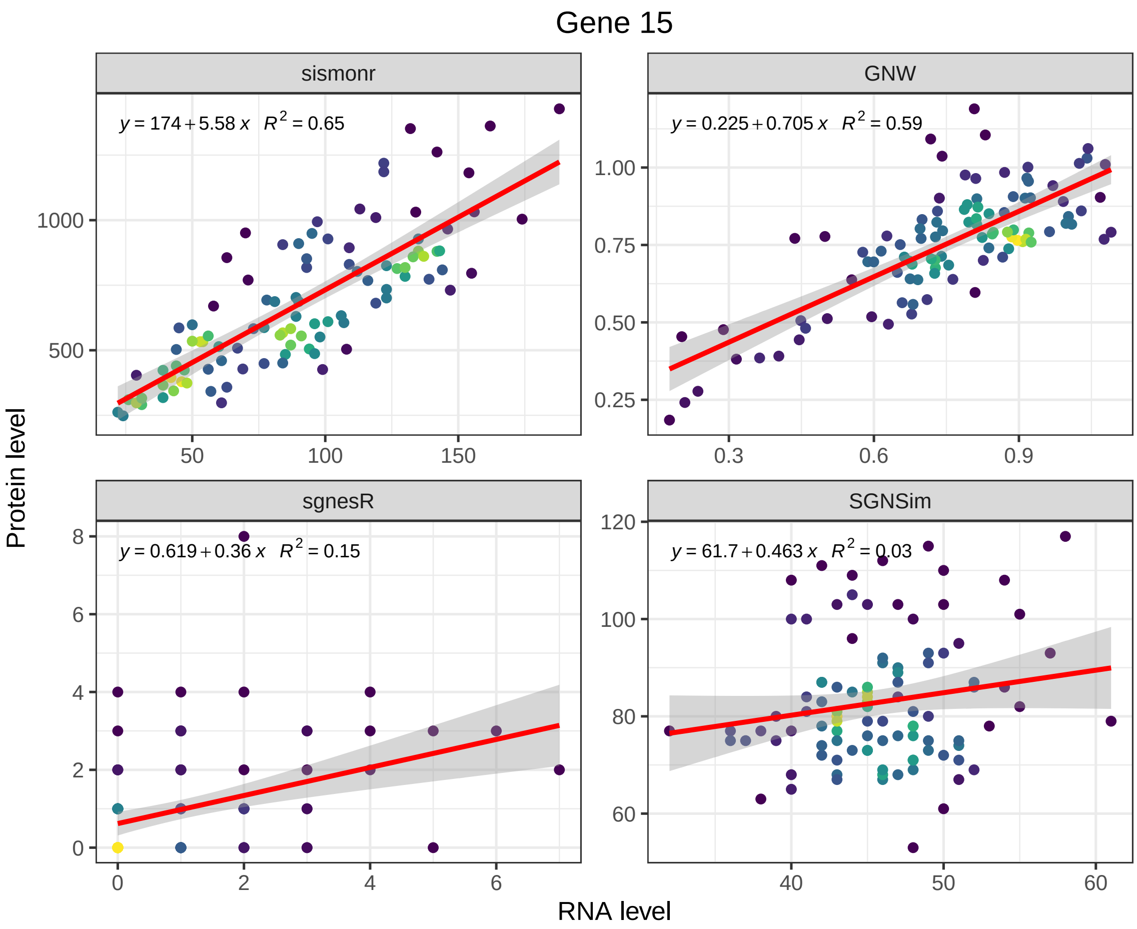


Figure 2.9: RNA and protein abundance (for `sismonr`, `sgnesR` and `SGNSim`) or relative concentration (for `GeneNetWeaver`) of gene 15 in the 100 simulations performed by each simulator. The colour of the points represent the point density in this region: a darker (blue) colour signifies lower point density, while a lighter (yellow) colour represents a higher density of points.

the simulations. This depends of course on the size of the system in terms of number of regulatory relationships and on the abundance of the different molecules. The strength of (semi-)deterministic simulators such as `GeneNetWeaver` and `SynTReN` is the ability to generate very large datasets in only a couple of seconds. At the other end of the spectrum, stochastic simulators have to simulate the evolution of molecules abundance reaction by reaction, leading to increased running time. It is especially the case for `sismonr`, as the default kinetic parameter distributions lead to realistic *in silico* systems in which the different molecules can be present in tens of thousands of copies. This is the price to pay to obtain more realistic simulations.

## 2.6 Concluding remarks

I present `sismonr`, an R package for the generation and the simulation of in silico biological systems. sismonr simulates the expression profiles of the genes linked via a regulatory network. I illustrated the behaviour of `sismonr`, `sgnesR`, `SGNSim`, `GeneNetWeaver` and `SynTReN` by simulating the expression of genes (and proteins when available) for a 20 gene-network. It was shown that, contrary to existing simulators, `sismonr` generate genes expression profiles in which the transcription and translation of genes is uncoupled, and with protein-to-RNA ratios in the range of observed experimental values. This comparison also highlighted (i) the impact of simulating genetic variability in the resulting RNA-protein levels correlations, and (ii) the impact of simulating biological noise to generate more realistic regulator-target relationships. In addition, `sismonr` offers some unprecedented features that cannot be compared to existing simulators.

#### Modelling of non-coding genes and post-transcriptional regulation

One of the main novelties of `sismonr` is the ability to simulate post-transcriptional regulations among genes. Namely, in addition to transcription regulations, `sismonr` allows a regulator to control a target gene at the level of: translation (the regulator influences the rate at which an RNA molecule is translated into a protein), RNA decay (the regulator can increase the decay of the RNAs of its target), protein decay (the regulator can increase the decay rate of the proteins of its target), and post-translational modification (the regulator can transform its target proteins into their modified form). Post-transcriptional regulatory relationships are key factors in the reconstruction of real regulatory networks from experimental data, as they modify the correlation patterns between the RNA levels of regulators and targets and between the RNA and protein levels of coding genes. This explains the impeded ability to decipher regulatory networks from transcriptomics data alone. Modelling post-transcriptional regulation is hence an important consideration when generating benchmark data to be used for assessing the performance of network inference methods.

`sismonr` also includes non-coding regulators –transcripts are not translated into proteins and instead directly act as regulators– in the generated systems, while in existing simulators, all genes are translated into proteins. This is an important aspect of generating realistic *in silico* simulations, as regulation by non-coding regulators is pervasive in biological systems (Storz et al., 2005).

#### Modelling of ploidy and genetic mutations

The second main improvement made by `sismonr` over existing simulators is the explicit modelling of the ploidy of the system. The number of copies of each gene present in the system can be defined by the user during the generation of an *in silico* system, and is not restricted to the haploid situation as in other tools. Furthermore, `sismonr` keeps track of the homolog of origin of each molecule, allowing to simulate and compare the expression of different homologs of a same gene.

Additionally, `sismonr` includes genetic variability in the simulations for each gene by defining a set of different alleles. Each of these alleles encodes different genetic mutations that affect the kinetic properties (transcription rate, translation rate, etc.) of the gene. The user can thus simulate the expression of genes in a system for different *in silico* individuals, each carrying a potentially different set of alleles for the genes in the system. This idea is similar to the concept of multi-factorial perturbations in `GeneNetWeaver`, allowing to repeat the simulation of a system by introducing small random perturbations of the genes transcription rates in each simulation. However, with the concept of *in silico* population and different alleles, the user can choose to introduce a population structure in the set of simulations, as some individuals will carry the same alleles for some genes.

A similar concept is implemented in SysGenSIM, which permits the modelling of the impact of eQTLs on gene expression. These eQTLs can affect the basal transcription rate of genes (cis-eQTLs) or the strength of a transcription regulation (trans-eQTLs). `sismonr` improves upon this idea by introducing genetic mutations affecting other aspects of the expression process of each gene like the translation of the gene or its products efficiency to bind to their targets.

#### Improved user interaction with the generated networks and mathematical models and visualisation capacity

Existing tools such as `GeneNetWeaver` or `sgnesR`, in addition to the random generation of synthetic networks, allow the user to provide an existing network as input. However, once the network is generated, the user cannot (easily) interact with it to modify the regulatory interactions or the kinetic parameters of the associated mathematical model. With `sismonr`, I provide functions to easily add or remove genes and regulatory interactions from the network. In addition, all kinetic parameters are stored in clearly labelled data frames, which are accessible to the user, who can also interact with them and modify their values. Furthermore, `sismonr` offers visualisation tools to represent the regulatory network as a graph (box SG3 in Figure 2.1), as well as functions to plot the results of a simulation, either in the form of abundance curves or time-dependent heatmaps (box GES3 in Figure 2.1).

The package can be used to generate benchmark datasets for the evaluation of network inference methods. As the algorithm provides the abundance of both RNAs (coding and non-coding) and proteins, and models the impact of genetic variations, the benchmark datasets can also be used to validate multi-omics integration methods.

### Areas of future work

`sismonr` uses a preferential attachment scheme to ensure that the in- and out-degree distributions of the generated networks follow a power-law distribution, which has been shown to be an important property of biological networks. Biological networks also exhibit other properties (such as betweenness centrality, closeness, or local topology properties like the presence of motifs), which have not been explicitly accounted for in the network generation by `sismonr`. This modelling choice is motivated by the fact that, while most of these properties have been reported for transcription regulatory networks, less is known about gene regulatory networks encompassing other types of regulators. This is especially true for the presence of motifs that are typical to transcription factors. It would be interesting to integrate such properties in the network generation process.

When simulating the binding of transcription factors to their target, `sismon` makes the assumption that different regulators bind independently to their binding sites. However, this assumption might be violated in biological systems, for example due to the opening and closing of the chromatin affecting the binding of successive regulators. This was not considered in the current version of `sismonr`, as it would introduce additional complexity. It could however be implemented by varying the binding propensities of regulators according to the number of previously bound regulators.

Lastly, any simulation tool would benefit from a validation of the generated simulations against real data. In the case of `sismonr`, comparing the topology of generated networks to regulatory networks reconstructed from experimental data for a range of organisms would be valuable. However, such comparison is hindered by the fact that our knowledge about organism-specific regulatory networks is incomplete. Experiments have to date only uncovered a fraction of all regulatory interactions between genes' products. Therefore, it is possible that the true underlying biological networks and the reconstructed networks exhibit slightly different properties. Similarly, comparison of generated gene expression profiles to measured RNA protein levels would require 1) knowledge of the different rates in the biological system investigated and 2) the ability to measure the absolute abundance of RNAs and proteins.

# Chapter 3

# Investigating causal inference methods performance in uncovering transcription and post-transcriptional gene regulatory networks

## 3.1 Introduction

In biological organisms, information encoded in the DNA is processed through the expression of genes: genes are first transcribed to produce messenger RNAs (mRNAs), which are later translated into proteins. Gene expression is a complex multi-step process, each of these steps being tightly regulated by molecular factors such as proteins (e.g. transcription factors) or regulatory RNAs. This complex regulation allows cells to adequately respond to environmental perturbations or extracellular cues. Therefore, deciphering patterns of regulation among genes and their products provides insight the functioning of cells and allows us to answer questions about diverse topics such as the development of disease or the molecular mechanisms driving specific biological phenomenon (e.g. the colouration of plants). The advent of high-throughput technologies such as microarrays or more recently RNA-sequencing data brought us one step closer to reconstructing these regulations at a genome-wide scale through measurements of RNA levels for a large fraction of the genes expressed in an organism of interest. Historically, study of gene expression regulation has been focused on regulation of the transcription step. This is mainly due to the fact that measurements of mRNA levels (i.e. transcriptomics datasets) were easier to obtain and thus used as a proxy for gene expression. However, it has been shown than post-transcriptional regulation is a pervasive mechanism in biological systems (Angelin-Bonnet et al., 2019; Buccitelli & Selbach, 2020), and thus plays a crucial role in gene expression regulation.

Information about gene-gene interactions – regulatory interactions as one gene's products regulates the expression of another gene, or interaction between the two genes' products, etc. – can be summarised in graphs in which nodes represent the genes and edges between them depict evidence that the genes interact or are associated (de Jong, 2002). When these graphs inform us about regulatory interactions, i.e. the impact of a gene or its products on the expression of a target gene, they are termed gene regulatory networks or GRN. A common approach to gaining a better understanding of such regulatory networks is through the reconstruction of co-expression among the genes, typically based on transcriptomics data. Reconstructing co-expression networks can be done by using a similarity metric such as correlation or mutual information (Altay & Mendi, 2017; Li et al., 2015). However, such reconstructed networks typically lack information about the directionality of the regulations. In recent years, attention has thus shifted from co-expression networks to the search for a better understanding of the flow of information through genes. The concept of causal inference is appealing, as it allows to reconstruct the chain of effects and causes among the measured variables – for example between genes based on a measure of their expression (Drton & Maathuis, 2017). Similarly to regulatory networks, causal graphs can be used to summarise our knowledge about causal relationships between variables or genes.

There are different approaches to reconstructing causal relationships among a set of observed variables. A first option is to consider the (in)dependences between variables. Testing whether two variables are independent conditionally on a set of other variables allows us to decide whether the two investigated variables are causally related (Colombo & Maathuis, 2014; Spirtes & Glymour, 1991; Tsamardinos et al., 2003a). Methods that rely on conditional independence tests are referred to as constraint-based methods. Another approach is to iteratively reconstruct the causal graph by testing how modifications in the candidate graph improve its fit to the data (Chickering, 2003; Ramsey et al., 2017). Methods implementing this strategy are termed score-based methods; as they rely on a score to assess the fit of a candidate graph to the data. Lastly, hybrid methods aim at leveraging the advantages of both constraint- and score-based methods by merging the two strategies, typically using the former to constrain and guide the latter (Nandy et al., 2018; Tsamardinos et al., 2006). Importantly, one of the common hypotheses of causal inference is that of causal sufficiency: we assume that all variables that play a causal role in the system under investigation are observed. However, it is rather common that some hidden confounders affect the measured variables. In consequence, methods were developed that do not assume causal sufficiency and instead seek to assess the impact of these confounders or hidden variables along with the causal relationships among the observed variables (Ogarrio et al., 2016; Spirtes et al., 1999; Tsirlis et al., 2018). It is of special interest in the context of gene regulation as it is a multi-step process that involves different molecules, e.g. RNAs and proteins, while the reconstruction is typically performed on measurements of a single cellular level, often RNAs.

A number of reviews and comparison studies have been published, that aim at contrasting existing causal inference methods (e.g. Heinze-Deml et al., 2018; Constantinou et al., 2020; Scutari et al., 2019); however comparisons of such methods in the context of reconstructing biological networks are scarce, with studies generally focusing on causal inference between triplets of variables (Ahmed et al., 2018; Auerbach et al., 2018; Hill et al., 2016). Often, such studies use a mathematical model to simulate data in order to assess the performance of the causal inference methods considered. The advantage of using simulated data is that it allows researchers to focus the evaluation on some specific aspects of the data. For example, the recent review by Constantinou and collaborators (Constantinou et al., 2020) focuses on the impact of noise, missing or incorrect values, merged states (for discrete variables) and presence of unobserved variables on the reconstruction of causal graphs. In a similar spirit, I aim at assessing the performance of a number of causal inference methods on simulated datasets that mimic biological datasets, specifically in the presence of post-transcriptional regulation among genes. This is important as it affects the patterns of associations observed between genes at the RNA level, an thus the reconstruction of regulatory networks from observations of RNA abundance.

In the present study, I evaluate the performance of nine state-of-the-art causal inference methods along with two popular network inference methods across a number of different simulation configurations, that differ in the type of regulation occurring between the genes. The simulated datasets are designed to match as closely as possible typical experimental data, via the use of a stochastic simulator that mimics biological noise, small genetic perturbations across the observations in each given dataset, presence of genes unrelated to the network and confounders in the form of protein products that are not observed. These simulations have been designed to test the performance of causal inference methods in realistic scenarios, i.e. in cases where the assumptions made by the methods – causal sufficiency, i.e. all variables are observed, or acyclicity, i.e. absence of feedback loops in the causal graphs – are violated. This will allow us to determine how well the methods are robust to deviations from their assumptions and whether they can still detect causal signal in such scenarios. I start by presenting the settings of the evaluation, from the generation of simulated datasets using the New Zealand eScience Infrastructure (NeSI) high-performance computer, with a brief presentation of the different methods evaluated, to the methodology used for the performance evaluation. I then analyse the performance of the different methods across the simulation scenarios, including a discussion about the running time of each method, the choice of the different tuning parameters, and the performance in reconstructing different types of causal relationships via answering causal queries.

## 3.2 Materials and methods

A schema of the analysis workflow used throughout this chapter is presented in Figure 3.1.

Table 3.1: Configurations investigated: for each configuration, I simulated networks with 20 genes, among which 10 regulator genes. The configurations differ in the number of transcription and post-transcriptional regulators as well as the type of post-transcriptional regulation modelled.

| Configuration | Number of TC* regulators | Type of post-TC* regulation | Number of post-TC* regulators |
|---|---|---|---|
| Configuration 1 | 10 | - | 0 |
| Configuration 2 | 7 | Translation | 3 |
| Configuration 3 | 7 | RNA decay | 3 |
| Configuration 4 | 7 | Protein decay | 3 |
| Configuration 5 | 7 | Protein PTM† | 3 |
| Configuration 6 | 5 | Translation | 5 |
| Configuration 7 | 5 | RNA decay | 5 |
| Configuration 8 | 5 | Protein decay | 5 |
| Configuration 9 | 5 | Protein PTM† | 5 |
| Configuration 10 | 3 | Translation | 7 |
| Configuration 11 | 3 | RNA decay | 7 |
| Configuration 12 | 3 | Protein decay | 7 |
| Configuration 13 | 3 | Protein PTM† | 7 |

* TC = transcription

† PTM = post-transcriptional modification

### 3.2.1 Simulation settings

The R package `sismonr` (Angelin-Bonnet et al., 2020) was used to generate the simulated data. Briefly, `sismonr` generates random regulatory networks satisfying default or user-controlled properties, and a set of *in silico* individuals, each carrying different versions or alleles of each gene. For each gene, the different alleles carry unique genetic mutations that affect the kinetic properties of the gene or of its regulatory interactions. `sismonr` then uses a stochastic simulation algorithm to simulate, for each *in silico* individual, the evolution over time of the absolute abundance of the gene products (RNAs and proteins) for all genes in the regulatory network. For more details about the `sismonr` package, the reader is referred to Chapter 2.

I evaluated the performance of causal inference methods for 13 different simulation configurations. Each configuration differs in the type of regulation occurring between the genes, and in the number of transcription and post-transcriptional regulators present in the regulatory networks (Table 3.1). For each configuration, 20 regulatory networks of 20 genes were generated, among which 10 regulator genes, and the remaining 10 genes considered as target genes i.e. that cannot regulate gene expression (box DS1 in Figure 3.1). The topologies of the 20 networks generated for each configuration are unique, i.e. not repeated between the configurations. Each network was obtained using the network

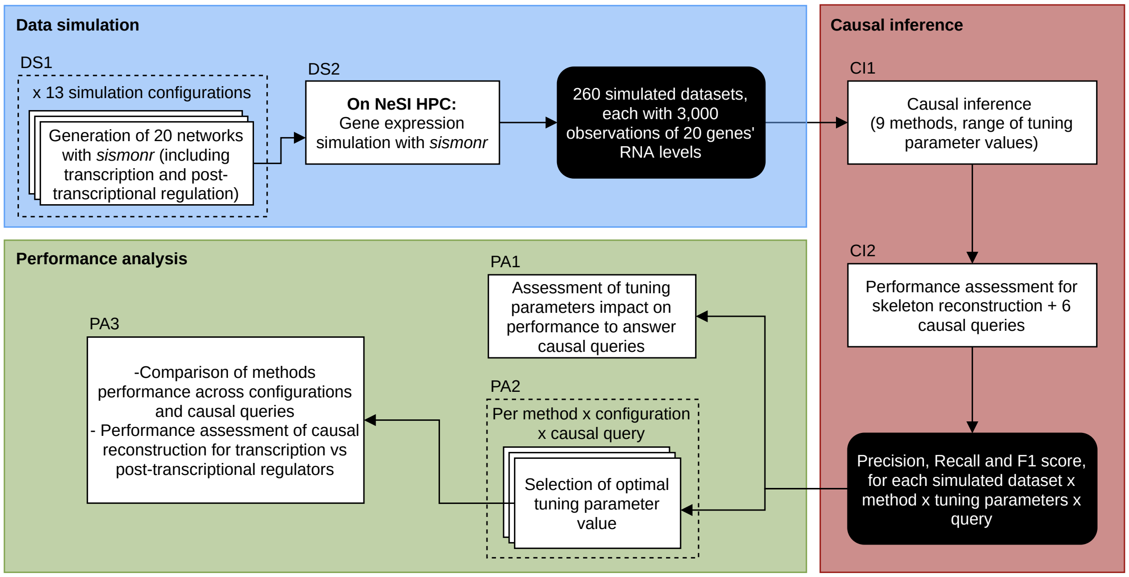

Data simulation
DS1
x 13 simulation configurations
Generation of 20 networks with *sismonr* (including transcription and post-transcriptional regulation)
DS2
**On NeSI HPC:** Gene expression simulation with *sismonr*
260 simulated datasets, each with 3,000 observations of 20 genes' RNA levels
Causal inference
CI1
Causal inference (9 methods, range of tuning parameter values)
CI2
Performance assessment for skeleton reconstruction + 6 causal queries
Precision, Recall and F1 score, for each simulated dataset x method x tuning parameters x query
Performance analysis
PA1
Assessment of tuning parameters impact on performance to answer causal queries
PA2
Per method x configuration x causal query
Selection of optimal tuning parameter value
PA3
-Comparison of methods performance across configurations and causal queries
- Performance assessment of causal reconstruction for transcription vs post-transcriptional regulators

Figure 3.1: Schema of the analysis workflow used throughout the chapter. Input/output datasets and endpoints are presented in black rounded boxes and analysis steps are shown in white rectangle boxes. The coloured rectangles outline different themes in the analysis.

generator implemented in `sismonr`. The choice of generating networks with 20 genes, and using 20 networks for each configuration, stems from a need for computational efficiency. This choice will be discussed in Section 3.4 (Concluding remarks). This yielded 260 simulated datasets. Note that not all 20 genes in a network are linked by regulatory interactions. Genes not involved in regulation act as decoy in the inference task, i.e. observed variables that do not play a causal role. Moreover, feedback loops are possible in the generated networks, consistently with observed biological networks. For each network, 3,000 observations of the RNA and protein abundance of the genes were generated, by simulating the expression of the genes for 3,000 *in silico* individuals. More specifically, for a given network, 100 different alleles were generated for each of the 20 genes , where one allele corresponds to one unique set of perturbation parameters for the given gene. The perturbation parameters have a small impact on the “basal” properties of the gene and its regulatory interactions. Then, for each *in silico* individual, two alleles are sampled with replacement for each gene from the pool of 100 alleles created for this gene. I chose to generate 100 alleles for each gene so that the generated individuals can be quite different, similarly to sampling from a population of unrelated individuals. The number of 3,000 individuals was chosen as it can be considered as a upper limit to the number of observations typically available in experimental datasets, but provides enough observations to assume that causal inference can be performed. I then simulated the expression of the genes for 2,000 (simulated) seconds for each *in silico* individual. The expression data for a given network was obtained by recording for each individual the RNA abundance of each gene at the final simulation time.

### 3.2.2 NeSI High Performance Computing Platform

The simulations were run on the NeSI high-performance computing (HPC) platform (`https://www.nesi.org.nz` – box DS2 in Figure 3.1). The different networks were simulated in parallel on 260 cores on the Mahuika cluster, using a SLURM (set of commands passed to the cluster job scheduler) and a R scripts to run the simulation of each network on a different core. Prior to running the full simulation, the running time and memory usage of the full job (i.e. simulation of 3,000 individuals for each network) were estimated by simulating for all networks a subset of the individuals of increasing size, ranging from 10 to 90. For each subset size, the running time in seconds and memory usage in MB were recorded and a linear regression model was used to predict the running time and memory usage for 3,000 individuals. As the estimated running time varied across the networks, I selected the maximum running time and memory usage across all networks, plus 25% of the estimated value, as a limit for the full simulation. Note that this method provided an upper limit for the running time of the simulations, but some simulations finished in one-third of this upper limit. Therefore, running time estimation could be improved by estimating the total running time independently for each simulated network. The simulation yielded 260 datasets containing mRNA levels for 20 genes across 3,000 observations.

### 3.2.3 Causal inference methods investigated

This evaluation was focused on methods that had an available implementation in R. I also included two popular network inference methods that are not inferring causal relationships, for comparison. Therefore, the present study includes the following algorithms:

#### Constraint-based methods

- **PC** (Peter-Clark – Colombo & Maathuis, 2014): The PC algorithm starts with a full undirected graph (i.e. all pairs of variables linked). Starting from $i = 0$, the algorithm seeks for each possible pair of variables connected in the graph a subset of size $i$ of their neighbours conditionally on which the two variables are independent. Conditional independence can be assessed by a choice of statistical tests, depending on the nature of the variables (discrete or continuous). If such a subset exists, the edge between the two considered variables is removed from the graph and the conditioning subset is recorded. Once all pairs of variables have been tested, the algorithm increases $i$ by one and repeats the search, until no pair of variables still linked in the graph has $i$ or more neighbours. The algorithm then tries to direct the remaining edges using information about the conditioning sets found in the previous step, starting with the v-structures in the graph (i.e. triplets of variables $A, B, C$ such that $A$ and $B$ are adjacent, $B$ and $C$ are adjacent but $A$ and $C$ are not adjacent, and $B$ is not in the conditioning subset that separates $A$ and $C$). It then iteratively orients as many edges as possible without creating new v-structures. I used the stable version of the algorithm presented by Colombo & Maathuis

(2014). The PC algorithm makes the assumption of causal sufficiency, i.e. it assumes that all variables involved in the causal system are observed.

- **FCI** (Fast Causal Inference – Spirtes et al., 1999): The FCI algorithm uses a search strategy identical to the PC algorithm in order to infer the skeleton of the causal graph, and to direct the v-structures in the resulting graph. Using this orientation, the FCI algorithm then updates the skeleton to account for the presence of confounders by performing additional conditional independence tests, before orienting as many edges as possible using a set of orientation rules. Contrary to the PC algorithm, FCI does not makes the assumption of causal sufficiency, and therefore accounts for latent variables (also called hidden confounders) that impact two or more observed variables.

- **FCI+** (Claassen et al., 2013): The FCI+ algorithm is a variant of the FCI algorithm that seeks to reduce the computational burden of the additional conditional independence tests performed in the second phase of the skeleton reconstruction in FCI. Specifically, the FCI+ algorithm proposes a faster alternative to selecting the subsets of variables used as possible conditioning sets in the additional conditional independence tests.

#### Score-based methods

- **GES** (Greedy Equivalent Search – Chickering, 2003): The GES algorithm uses a two-step approach to reconstruct the causal inference graph for a set of observed variables. In a first step termed the forward phase, the algorithm iteratively tests all possible one-edge additions in the candidate causal graph and selects the one that best improves the fit of the candidate causal graph to the data, according to a scoring criterion such as the Bayesian information criterion (BIC) or log-likelihood criterion. Once a local maximum is attained, the algorithm moves to the second or backward phase. In this phase, it iteratively seeks edge removals that best increase the score, until a local maximum is reached.

- **FGES** (Fast Greedy Equivalence Search – Ramsey et al., 2017): The FGES algorithm is a variant of the GES algorithm, that aims at optimising computations through the use of caching and parallelisation.

#### Hybrid methods

- **MMHC** (Max-Min Hill Climbing – Tsamardinos et al., 2006): The hybrid algorithm MMHC aims at making the best out of the concepts of both constraint-based and score-based algorithms. In a first phase termed the restrict phase, the MMHC algorithm computes the set of potential parents of each variable, using the Max-Min Parent and Children algorithm (MMPC) (Tsamardinos et al., 2003b). The latter uses the concept of mutual information to search a subset of variables to which a given variable is highly associated, conditionally on the other variables.

In the second or maximise phase, MMHC uses a greedy hill-climbing scheme to iteratively add, remove or re-orient edges in the graph in order to best improve a score measuring the fit of the candidate causal graph to the data. Edge additions are restricted to edges between any variable and its set of potential parents as computed in the first phase.

- **ARGES** (Adaptive Restricted Greedy Equivalence Search – Nandy et al., 2018): The ARGES algorithm seeks to improve the score-based GES algorithm, by restricting the set of possible edges that can be added to the candidate causal graph during the forward phase. The set of possible edges is computed using the graphical LASSO (gLASSO) graph estimation method (Meinshausen & Bühlmann, 2006), a variable selection scheme similar to LASSO with penalisation to enforce sparsity.

### Network-inference methods

- **ARACNe** (Algorithm for the Reconstruction of Accurate Cellular Networks – Margolin et al., 2006): The ARACNe algorithm is a network inference method, i.e. it does not seek causal relationships among variables. Instead, it uses the concept of mutual information to detect pairs of variables that are highly associated, and removes indirect interactions in triplets of variables by removing the edge between the two variables that share the lowest association score.
- **GENIE3** (Gene Network Inference with Ensemble of trees – Huynh-Thu et al., 2010): GENIE3 is another network inference method that relies on random forest to learn independently for each gene the list of its regulators. The average importance score of a candidate regulator in a set of many trees is used as confidence score or weight in the resulting adjacency matrix between all genes or variables.

For the PC, FCI, FCI+, GES and ARGES algorithms, I used the versions implemented in the `pcalg` package (Kalisch et al., 2012). For the FGES algorithm, I used the implementation available in the `rcausal` package (Wongchokprasitti, 2019). For the MMHC algorithm, I used the `bnlearn` package (Nagarajan et al., 2013). For the ARACNe algorithm, I used the version implemented in the `minet` package (Meyer et al., 2008). For the GENIE3 algorithm, I used the `GENIE3` package (Huynh-Thu et al., 2010).

In addition to the algorithms mentioned above, a number of other causal inference methods have been proposed in the literature, which use additional assumptions on the data distribution in order to distinguish between different DAGs from a same Markov Equivalence class (e.g. Shimizu et al., 2006; Hoyer et al., 2008). Such algorithms were not included in the present comparison because of the additional assumptions they make. In addition, I left out of the comparison other variants of the PC and FCI algorithms (e.g. Colombo et al., 2012), as there already is a version of these algorithms included.

### 3.2.4 Types of output causal graph

A key point when evaluating the performance of different causal inference methods is to understand the type of causal graph returned by each method. Indeed, different methods return causal graphs of different nature, that cannot be interpreted in the same way. Ignoring this difference during the evaluation leads to biased results. In the present section I briefly describe the different types of causal graphs returned by the methods considered (see also Table 3.2) and how to interpret them in terms of causal relationships (for a definition of causal graph the reader is referred to Section 1.4.2).

#### DAGs

Directed acyclic graphs (DAGs) are fully directed graphs with no loops. The nodes in the DAG each represent a variable, and when it is interpreted as a causal graph, a directed edge from one variable $A$ to another variable $B$ (i.e. $A \rightarrow B$) indicates that $A$ has a direct causal effect on $B$. We refer to $A$ as a parent of $B$. Any node $C$ for which there exists a directed path from $C$ to $B$ (e.g. $C \rightarrow A \rightarrow B$) is called an ancestor of $B$. By convention, the parent $A$ is also considered as an ancestor of $B$. If the DAG represents a causal graph, it informs about the conditional independences among the considered variables.

#### CPDAGs

Several DAGs sharing a same skeleton (i.e. set of edges when ignoring their orientation) but with different edges orientation can encode the same set of conditional independences among a set of variables. These DAGs are termed Markov equivalent, and the set of all Markov equivalent DAGs is named the Markov equivalence class of the DAGs (see also Section 1.4.2). In consequence, methods such as PC or GES cannot infer the orientation of all edges based on observational data alone, and instead aim at reconstructing the Markov equivalence class of the true causal graph. Therefore, the

Table 3.2: Output graph and tuning parameters of the causal inference methods evaluated

| | Type of output graph | | | | Tuning parameter | | |
|---|---|---|---|---|---|---|---|
| Method | Undirected graph | DAG | CPDAG | PAG | $\alpha$ | Penalty | Threshold |
| PC | | | × | | × | | |
| FCI | | | | × | × | | |
| FCI+ | | | | × | × | | |
| GES | | | × | | | × | |
| FGES | | | × | | | × | |
| MMHC | | × | | | × | × | |
| ARGES | | | × | | | × | |
| ARACNe | × | | | | | | × |
| GENIE3 | | × | | | | | × |

output of these methods is a completed partially directed DAG (CPDAG). In a CPDAG, a directed edge is present from $A$ to $B$ (i.e. $A \rightarrow B$) if this edge is present in all DAGs in the Markov equivalence class with the same orientation. The edge indicates that $A$ is a parent of $B$ (i.e. has a direct causal effect on $B$). On the contrary, the edge between $A$ and $B$ will be undirected (i.e. $A \text{ --- } B$) if this edge is present with one orientation in some of the Markov equivalent DAGs (e.g. $A \rightarrow B$) and with the opposite orientation in others (e.g. $A \leftarrow B$). Therefore, an undirected edge in a CPDAG can be interpreted as information that the two variables are causally related, but we cannot say that one is parent of the other solely based on the available data.

### PAGs

DAGs and CPDAGs assume that all variables involved are observed (this is the assumption of causal sufficiency). However, it is possible that unobserved or hidden variables causally influence some of the observed variables. The presence of these latent variables hinder our ability to reconstruct parental causal relationships among observed variables. Information about causal relationships between observed variables in the presence of hidden variables can be represented with a maximal ancestral graph (MAG). In such graph, the edges cannot be interpreted in the same way as in DAGs; instead, the endpoints of the edges hold the causal meaning. An edge from $A$ to $B$ with a tail at $A$, $A \mathrel{-\!\!\!\ast} B$, where the star symbol means “any type of endpoint symbol” (i.e. either a tail or an arrow), signifies that $A$ is an ancestor of $B$. On the contrary if the edge has an arrowhead pointing toward $A$, $A \mathrel{\leftarrow\!\!\!\ast} B$, it means that $A$ is not an ancestor of $B$. A bidirected edge between the variables, $A \longleftrightarrow B$ indicates the presence of a hidden variable affecting both $A$ and $B$. Methods that account for the possible presence of hidden variables, such as FCI and FCI+, return a partial ancestral graph (PAG), which represents the Markov equivalence class of the underlying MAG. In a PAG, arrowheads and tails can be interpreted in terms of ancestral causal relationships in the same way as for a MAG. In addition, uncertainty about the orientation of an edge endpoint is denoted as $A \mathrel{\circ\!\!-\!\!\!\ast} B$, with the circle representing the method’s inability to orient the endpoint as either an arrow or a tail.

### 3.2.5 Methods tuning parameters

Each method investigated relies on one or more tuning parameters (see Table 3.2), which impact the causal graph reconstruction. Therefore, I evaluated the performance of the nine methods over a range of values for these parameters (box CI1 in Figure 3.1). I briefly present below the meaning of each tuning parameter as well as the range of values investigated in this study.

- $\alpha$: a constraint-based method needs to make a decision on the result of a conditional independence test between two variables. The tuning parameter $\alpha$ is used as a significance threshold for the p-values of such tests. If the p-value of a conditional independence test is smaller than $\alpha$, the null hypothesis of the test, stating that the two investigated variables are conditionally independent, is rejected and the two variables are considered as conditionally dependent. This

Table 3.3: Values tested for the tuning parameters of the different causal inference methods.

| Tuning parameters | Tested values |
|---|---|
| $\alpha$ | $1 \times 10^{-4}$, $2.5 \times 10^{-4}$, $5 \times 10^{-4}$, $7.5 \times 10^{-4}$, $1 \times 10^{-3}$, $2.5 \times 10^{-3}$, $5 \times 10^{-3}$, $7.5 \times 10^{-3}$, $5 \times 10^{-4}$, 0.01 to 0.4 every 0.01 |
| Penalty | $1 \times 10^{-4}$, $5 \times 10^{-4}$, $1 \times 10^{-3}$, 0.05 to 0.15 every 0.01, 0.2 to 5 every 0.2, |
| Threshold | 0.01 to 0.99 every 0.01 |

leads to adding an edge in the causal graph between the two variables (or not removing the edge from the complete graph). Therefore, lower values of $\alpha$ yield sparser inferred causal graphs as less p-values are below the $\alpha$ threshold and thus more variables are considered conditionally independent. For the evaluation, I tested values for $\alpha$ ranging non-linearly from $1 \times 10^{-4}$ to $0.4$ (see Table 3.3). While this upper limit is an unrealistic value to use in practice (as it would lead to graphs that are very dense), high $\alpha$ values were included to assess the behaviour of the methods for these extreme cases.

- Penalty: a score-based method has to assess the fit of a given candidate causal graph to the data using some model selection criterion, e.g. the log-likelihood criterion. Such scores balance the fit of the model with its complexity, in order to avoid overfitting, by penalising the number of edges in the causal graph. High penalty values will result in sparser estimated causal graphs as the addition of edges during the graph reconstruction is more heavily penalised. For GES, MMHC and ARGES, the penalisation constant $\lambda$ used is the penalty times the log of the number of observations. For the FGES algorithm, the `penalty discount` parameter passed to the R function corresponds to two times the penalty in the scoring criterion. For this evaluation, I tested values for the penalty ranging non-linearly from $1 \times 10^{-4}$ to $5$ (see Table 3.3).

- Threshold: the two network inference methods considered, namely ARACNe and GENIE3, return weighted adjacency matrices, with values between zero and one, indicating the evidence or confidence in the existence of an edge between any two variables. In order to interpret this output as a causal graph, I use a threshold on the edges weight in order to retain only edges with weight above this threshold. A high threshold will result in sparser estimated graphs as only edges with the highest scores are retained. I tested here values ranging from 0.01 to 0.99 (see Table 3.3).

To reduce the computational burden, runs of a method (for a given network with a particular value of tuning parameter) exceeding 10 minutes are stopped and removed from the analyses.

### 3.2.6 Different causal queries and evaluation metrics

For each configuration, the performance of the different network and causal inference methods was evaluated with respect to different inference tasks (box CI1 in Figure 3.1). I assessed the ability of the methods to correctly reconstruct the skeleton of the causal graph, i.e. to correctly infer the presence or absence of an edge between two variables or genes, without considering the orientation of the edge or its causal interpretation. In addition, in order to evaluate fairly the performance of each method with respect to the type of output graph it infers, the methods' answers to different causal queries were evaluated, that each reflect the ability to detect a different type of causal relationship, as suggested in Heinze-Deml et al. (2018). Namely, for each pair $(A, B)$ of variables, the methods have to answer the following queries:

- the parent query: is $A$ a causal parent of $B$? The answer is "yes" if there is a direct causal link from $A$ to $B$. As the FCI and FCI+ algorithms only infer ancestral relationships, they cannot answer this causal query. Moreover ARACNe, which returns undirected graphs, cannot answer this query either.
- the potential parent query: is $A$ a potential causal parent of $B$? The answer is "yes" if there is a direct causal link from $A$ to $B$ or if the orientation of the causal edge between $A$ and $B$ could not be determined. Parents of a given variable are also considered as potential parents of the variable.
- the ancestor query: is $A$ an ancestor of $B$? The answer is "yes" if there is a direct or indirect causal path directed from $A$ to $B$. ARACNe, which returns undirected graphs, cannot answer this query.
- The potential ancestor query: is $A$ a potential ancestor of $B$? The answer is "yes" if there is a direct or indirect causal path from $A$ to $B$, possibly including causal edges for which the orientation could not be inferred.

In addition, I investigated the following negative queries, which are the complement of potential queries mentioned above:

- the not parent query: is $A$ not a parent of $B$? The answer is "yes" if there is no direct causal link from $A$ to $B$, even with undetermined orientation. It it the complement of the potential parent query.
- the not ancestor query: is $A$ not an ancestor of $B$? The answer is "yes" if there is no direct or indirect causal link from $A$ to $B$, even involving edges with undetermined orientation. It it the complement of the potential ancestor query.

However, the results are redundant with those presented for the potential queries (as they are complementary), and did not provide useful insight into the performance of the methods. Results for these two negative queries will therefore not be mentioned in the Results section.

Table 3.4 presents the types of edges/paths that are considered for a positive answer to each query, depending on the type of causal graph evaluated. The implementation of these causal queries found in the `CompareCausalNetworks` package (Heinze-Deml et al., 2018) was used. Each causal query is answered with respect to a given graph ($\mathcal{G}$), in the form of a $p \times p$ matrix $Q^{\mathcal{G}}$ (where $p$ is the number of observed variables). In the answer matrix, $Q^{\mathcal{G}}_{i,j} = 1$ ($1 \leq i, j \leq p$) if the answer is positive for the (ordered) pair of variables $(i, j)$ and 0 if the answer is negative. It is therefore possible to compare the answers obtained for a given query with two different graphs. In particular, one can compare the answers obtained with the network used to simulate data (the ground truth) to the graph predicted with a causal inference method.

For the skeleton reconstruction as well as each query, I compared the answers obtained with the causal graph reconstructed by a given method to the answers obtained with the true causal graph, to compute the number of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). These counts were used to calculate:

- The precision $Pr$: $Pr = \frac{\text{TP}}{\text{TP+FP}}$, i.e. the fraction of positive answers to the query that are correct;

Table 3.4: Types of edges between two variables $A$ and $B$ that, if present in a causal graph, lead to a positive answer for the corresponding causal query between $A$ and $B$.

| Query | DAG | CPDAG / undirected graph | PAG |
|---|---|---|---|
| $A$ parent of $B$ | $A \rightarrow B$ | $A \rightarrow B$ | $\varnothing$ |
| $A$ potential parent of $B$ | $A \rightarrow B$ | $A \rightarrow B$<br>$A - B$ | $A \mathrel{-\!\!\ast} B$<br>$A \mathrel{\circ\!\!-\!\!\ast} B$ |
| $A$ not parent of $B$ | Complement of potential parent query | | |
| $A$ ancestor of $B$ | path from $A$ to $B$ with edges $A \rightarrow B$ | path from $A$ to $B$ with edges $A \rightarrow B$ | path from $A$ to $B$ with edges $A \mathrel{-\!\!\ast} B$ |
| $A$ potential ancestor of $B$ | path from $A$ to $B$ with edges $A \rightarrow B$ | path from $A$ to $B$ with edges $A \rightarrow B$ and $A - B$ | path from $A$ to $B$ with edges $A \mathrel{-\!\!\ast} B$ and $A \mathrel{\circ\!\!-\!\!\ast} B$ |
| $A$ not ancestor of $B$ | Complement of potential ancestor query | | |

- The recall $Re$ (also termed sensitivity): $Re = \frac{\text{TP}}{\text{TP+FN}}$, the fraction of all correct answers that are positive;

The $F_1$ score (Rijsbergen, 1979), which corresponds to the harmonic average of the precision and recall scores, was also computed as follows:

$$F_1 = 2\frac{Pr \times Re}{Pr + Re}$$

The $F_1$ score quantifies the trade-off between prevision and recall. It takes values between zero and one, with high values indicating both a good precision and good recall. As each method depends on one or more tuning parameters, the different scores presented above were first computed for each simulated dataset and each tested value of the tuning parameters. To compare the different methods, the value(s) of the tuning parameter(s) yielding the highest mean $F_1$ score across the 20 datasets in a considered configuration were selected, independently for each configuration and each query. Using the $F_1$ score as criterion is a sensible choice as it allows to assess the trade-off between precision, i.e. the ability to make correct predictions and recall, i.e. the ability to not miss information.

The skeleton reconstructed by the different methods were also compared by computing a Skeleton Similarity Score $Sk$. Given an inferred graph $\mathcal{G}_A$ with $N$ nodes, its symmetrical skeleton adjacency matrix is defined as $S_A = \{S^A_{ij}\}_{1 \leq i,j \leq N}$ with $S^A_{ij} = S^A_{ji} = 1$ if there is an edge between the nodes $i$ and $j$ in $\mathcal{G}_A$ and 0 otherwise. Thus, for two graphs $\mathcal{G}_A$ and $\mathcal{G}_B$, both with the same set of $N$ nodes, we have:

$$Sk = 1 - \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} |S^A_{ij} - S^B_{ij}|}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} S^A_{ij} + \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} S^B_{ij}}$$

For each method and causal query, the impact of setting the tuning parameters to different values was evaluated (box PA1 in Figure 3.1). Next, for each causal query, configuration and inference method independently, the value of the tuning parameter(s) yielding the best average $F_1$ score across the 20 datasets was selected (box PA2 in Figure 3.1). The methods performance in each configuration for the different causal queries was investigated (box PA3 in Figure 3.1).

## 3.3 Results and Discussion

The aim of this study is to assess how well different causal inference methods are able to reconstruct gene regulatory networks from RNA level data in the presence of post-transcriptional regulation. `sismonr` was used to simulate the expression of genes organised in regulatory networks that comprise transcriptional and post-transcriptional regulation, i.e. regulation of translation, RNA decay, protein decay, and protein post-translational modification. `sismonr` allows to simulate the expression of the gene products in a same network for different *in silico* individuals, each carrying genetic mutations affecting the kinetic properties of the network. These mutations are key to reconstructing causal relationships from observational data, as they are source of random noise between the observations (e.g. see Thomas & Conti, 2004). The performance of the causal inference methods was assessed across 13 simulations scenarios, each differing in the number of post-transcriptional regulators and the type of post-transcriptional regulation occurring in the simulated networks. These configurations are summarised in Table 3.1.

### 3.3.1 Running time

I present in Figure 3.2 the running time of the evaluated methods as a function of the values of the tuning parameters, across the 260 simulated datasets. The figure shows that the running time of the methods depending on the $\alpha$ parameter increases with the values of $\alpha$. Higher values of $\alpha$ lead to retaining more edges in the causal graph, thus increasing the search space for conditioning subsets during the graph reconstruction. For the same reason, the running time of methods using the penalty parameter decreases when the penalty value increases. Exploration moves in the graph search space are more restricted for high penalty values, since any addition of an edge in the candidate causal graph is heavily penalised. The running time of the network inference methods (ARACNe and GENIE3) do not depend on the threshold parameter as it is applied to the adjacency matrix returned after the computation. While the $\alpha$ parameter heavily influences the running time of the causal inference methods, the running time of methods using the penalty had a smaller variation across the range of values tested. Among the causal inference methods, FGES was the fastest, with an average running time of 0.23 seconds (SD 0.056s) per dataset across all configurations and tuning parameter values. MMHC was the second fastest, with an average running time of 0.47s (SD 1.49s), and FCI+ was third with an average running time of 0.88s (SD 2.23s). FCI was on average the slowest of the causal inference methods (28s, SD 85.9s) and was the method for which most runs were interrupted for high $\alpha$ values because they exceeded the 10 minutes limit. Among the network inference methods tested, ARACNe was on average faster than all causal inference methods, with an average running time of 0.02s (SD 0.010s), while GENIE3 was the slowest of all methods with an average running time of 287s (SD 35.2s), possibly due to its R implementation.

### 3.3.2 Methods performance when changing the tuning parameters

The performance of the causal inference methods was evaluated across a wide range of values for the different tuning parameters. Supplementary Figure E.1 presents the number of edges in the graphs inferred by the different methods across the 260 simulated datasets. As expected, the number of inferred edges increases with the tuning parameter $\alpha$, and decreases with the penalty and the threshold parameters. In order to assess the performance of the methods in answering the different queries, the

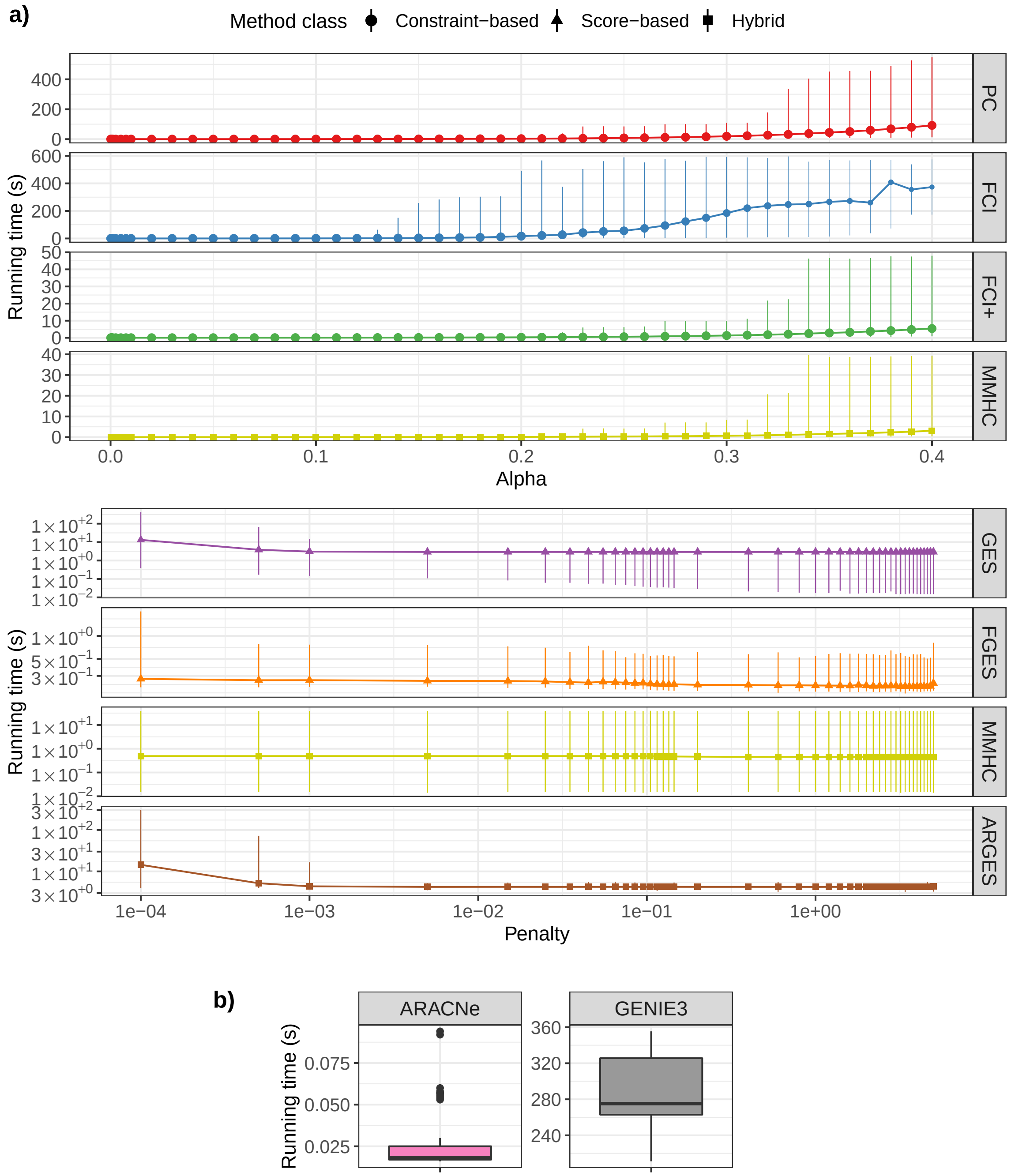


Figure 3.2: Running time (in seconds) of the different methods across all simulated datasets. a) Running time of the causal inference methods, as a function of the value of the tuning parameters. The points show the average running time of the methods across the 260 simulated datasets, with the vertical bars showing the minimum and maximum values. The size of the points represent the fraction of runs that finished within the 10-minutes limit (i.e. smaller points indicate that more runs exceeded the limit and were interrupted). b) Running time of the network inference methods across the 260 simulated datasets, for comparison, as it does not depend on any tuning parameter.

$F_1$ score of the predicted network was computed for each task in each configuration. In Figures 3.6 and 3.7, I present the mean $F_1$ scores for the six causal queries investigated for each method, in each scenario, across the range of tuning parameter values tested.

First, it can be seen that the mean $F_1$ score obtained for the parent and ancestor queries are quite low across all investigated methods (see Figure 3.6), with values not exceeding 0.4. Higher mean scores are obtained for the potential parent and potential ancestor queries. This is due to the fact that the parent and ancestor queries require the methods to correctly assess the orientation of edges, and non-oriented edges are therefore considered as false negative for these queries (see Table 3.4). On the contrary the potential parent and potential ancestor queries account for the uncertainty in edges orientation. Note that the MMHC and GENIE3 methods both return fully directed graphs. Therefore, their answers to the parent query will be identical to their answers for the potential parent query (see Table 3.4). Indeed, the answers to the parent query rely on directed edges while the answers to the potential parent queries also include undirected edges; however there will be none with MMHC and GENIE3. Similarly, their answers to the ancestor and potential ancestor queries will be identical. Moreover, the $F_1$ scores obtained for the parent and potential parent queries are in general higher than those obtained for the ancestor and potential ancestor queries. This is because the ancestor queries informs about the correct inference of paths in the graph, and are thus more sensitive to errors made during the graph reconstruction.

For the MMHC method, which depends on both $\alpha$ and the penalty, the impact of one tuning parameter's value on the results depends on the value of the other tuning parameter. In particular, when $\alpha$ is already large, further increase in its value only impacts the reconstructed graph if the penalty is very small. This is because high values of $\alpha$ means that the space of possible edges to add in the graph during the score-based phase of the algorithm is quite high, however if the penalty is not low, the cost of adding additional edges in the candidate graph prevents a lot of these potential edges to be included in the final reconstructed graph. I also note that the GES and ARGES methods yield almost identical results. As ARGES is an extension of the GES method that aims at improving the reconstruction by constraining the search space, it means that this additional constraint during the causal graph reconstruction does not impact or improve much the inference. Also, the results of PC and FCI for the potential parent query are very similar, which is to be expected as the FCI algorithm relies on the PC algorithm as a first step to reconstruct the skeleton of the causal graph.

The values of the tuning parameters for which the maximum mean $F_1$ score is achieved for each method and each simulation configuration are presented in Figures 3.3 and 3.4. For any given causal query, these values are quite consistent across the different configurations for each method, except in some cases, e.g. the FGES method for the parent query. In such cases, the method performs badly across all values of the tuning parameter. Therefore the value yielding the best $F_1$ score will be different across the configurations as it reflects small fluctuations in the $F_1$ score rather that a true improvement of the performance with this value of tuning parameter. On the contrary, when comparing between causal queries, the value(s) of the tuning parameters yielding the best mean $F_1$ score varies, specially between the parent and possible parent queries, and between the ancestor and possible ancestor queries. This is explained by the fact that the queries are not equally sensitive to different errors made in the reconstruction process. For example, the parent query ignores indirect edges between variables and thus is insensitive to inferred causal relationships that are false positive if the orientation of the causal relation couldn't be resolved (i.e. undirected edges in the inferred graph). The plots show that for the hybrid method MMHC, the choice of the $\alpha$ parameter has more impact on the resulting performance than choosing a value for the penalty parameter. This can be due to the fact that the the value of $\alpha$ chosen provides enough constraint on the sparsity of the graph, and guides the selection of the potential parents of each variable. Thus having a non-restrictive penalty does not reduce the performance of the method as spurious associations between variables have already been removed during the first phase of the algorithm. In general (i.e across methods, configurations and queries) $\alpha$ values ranging from $5 \times 10^{-3}$ to $5 \times 10^{-2}$ and penalty values ranging from $5 \times 10^{-2}$ to 1 seem to yield reasonable performance. For the threshold parameter, values below 0.3 yield sensible results for ARACNE, while results for GENIE3 depend heavily on the configuration and query considered.

### 3.3.3 Comparison of methods' performance

For each causal query, method and simulation configuration, the value of the tuning parameter that yielded the best average $F_1$ score across the 20 simulated networks was retained (see Section 3.3.2). This allows us to compare the best performance of the methods within and across the different configurations. The resulting mean $F_1$ scores are shown in Figure 3.5.

The best scores across the parent, ancestor and possible parent/ancestor causal queries are obtained for the configuration with only transcription regulation. Across all methods, the mean $F_1$ score decreases when the number of post-transcriptional regulators in the system increases. Slightly better scores are obtained for configurations involving regulation of RNA decay, as opposed to other types of post-transcriptional regulation. It can be easily explained, as the RNA decay regulation impacts the RNA levels of the genes, which are used for the causal graph reconstruction. Again, we

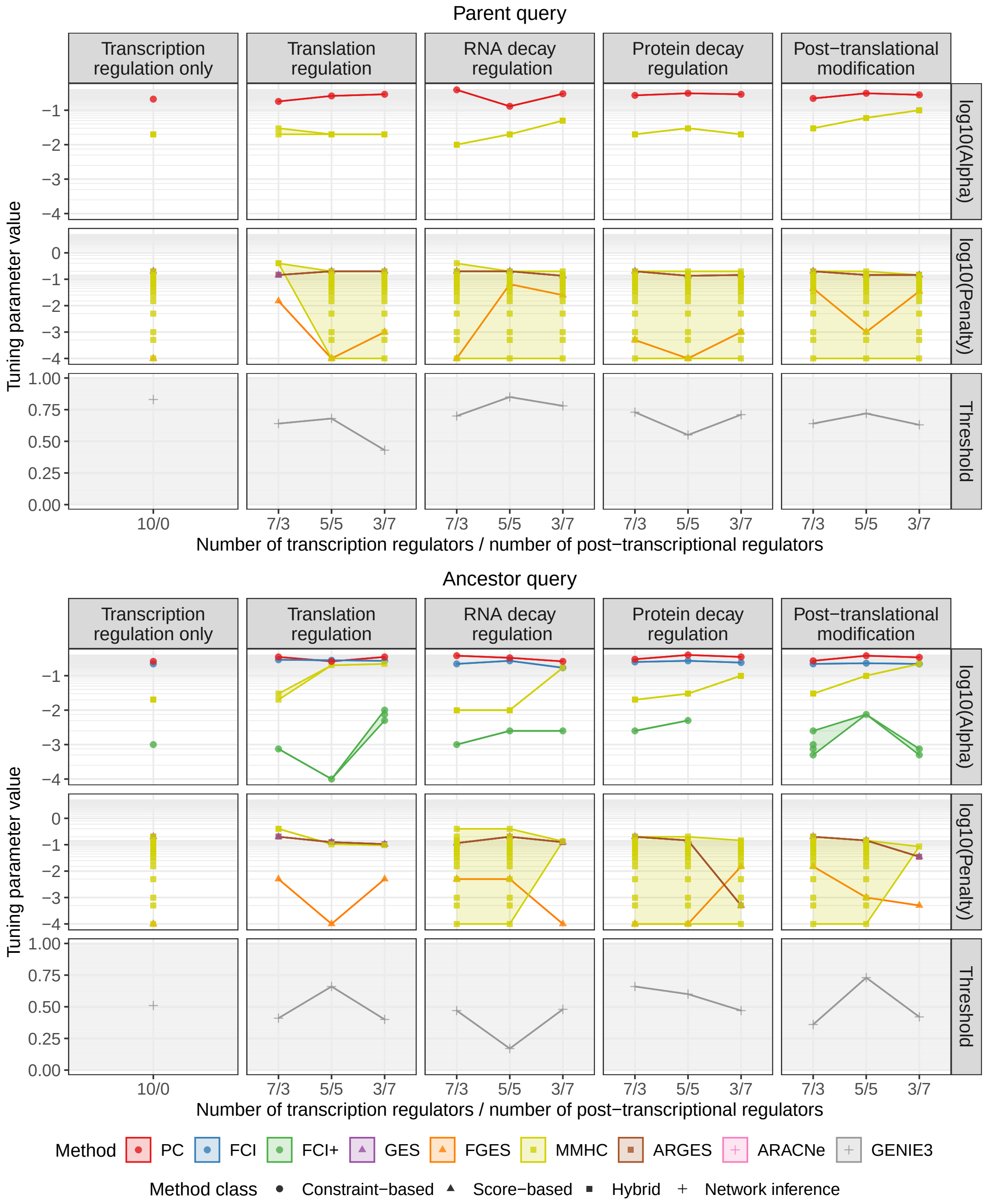


Figure 3.3: Values of the tuning parameters for which the highest mean $F_1$ score is obtained across the 20 datasets for each configuration (x axis) and method, for the parent (top panel) and ancestor (bottom panel) queries. Each point corresponds to a value of the corresponding tuning parameter for which the highest $F_1$ score is obtained for the method, configuration and query considered, with shaded areas and lines delineating the range of values for which the maximum mean score is obtained.

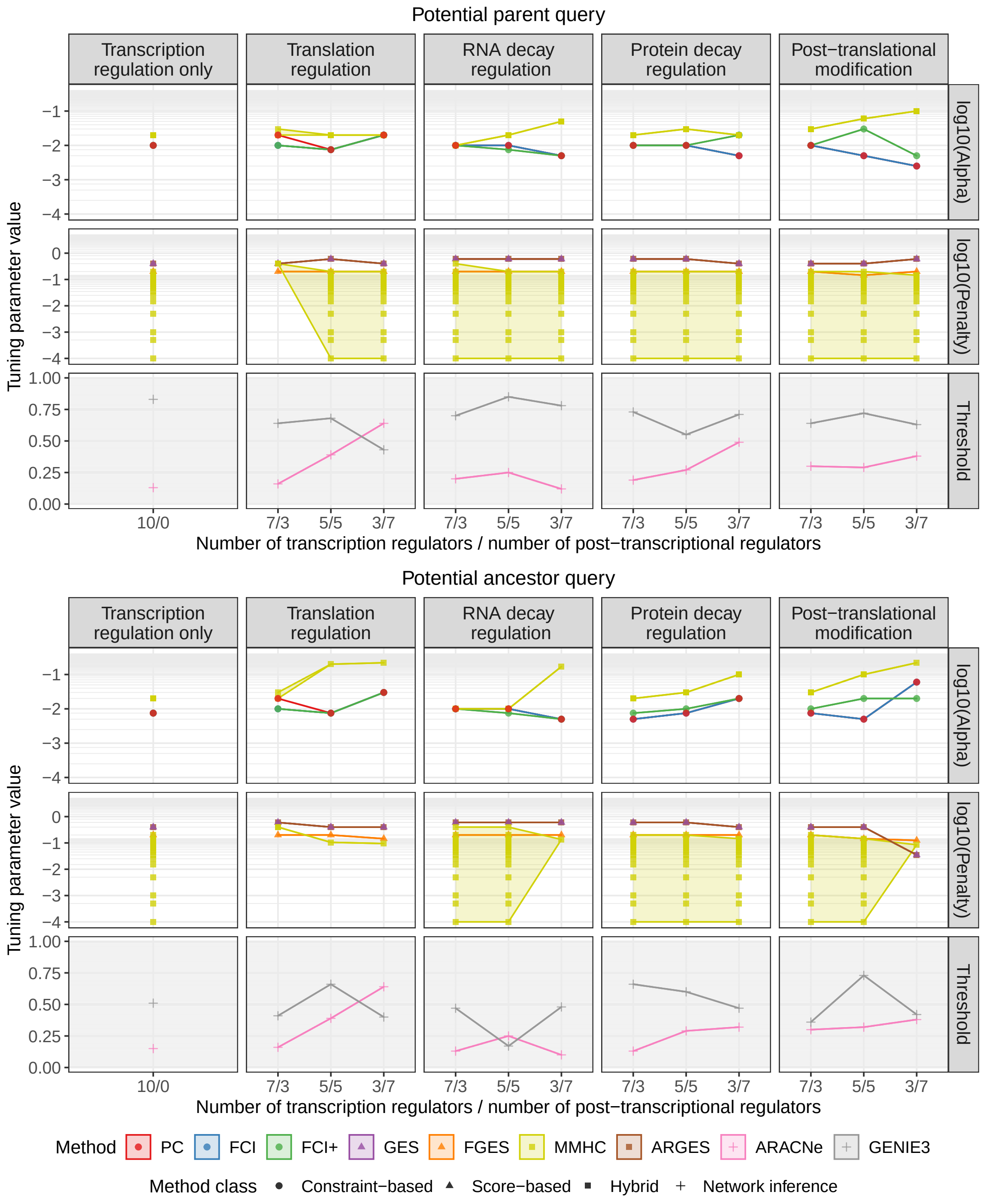


Figure 3.4: Values of the tuning parameters for which the highest mean $F_1$ score is obtained across the 20 datasets for each configuration (x axis) and method, for the potential parent (top pannel) and potential ancestor (bottom panel) queries. Each point corresponds to a value of the corresponding tuning parameter for which the highest $F_1$ score is obtained for the method, configuration and query considered, with shaded areas and lines delineating the range of values for which the maximum mean score is obtained. The corresponding $F_1$ values are shown in Figure 3.5

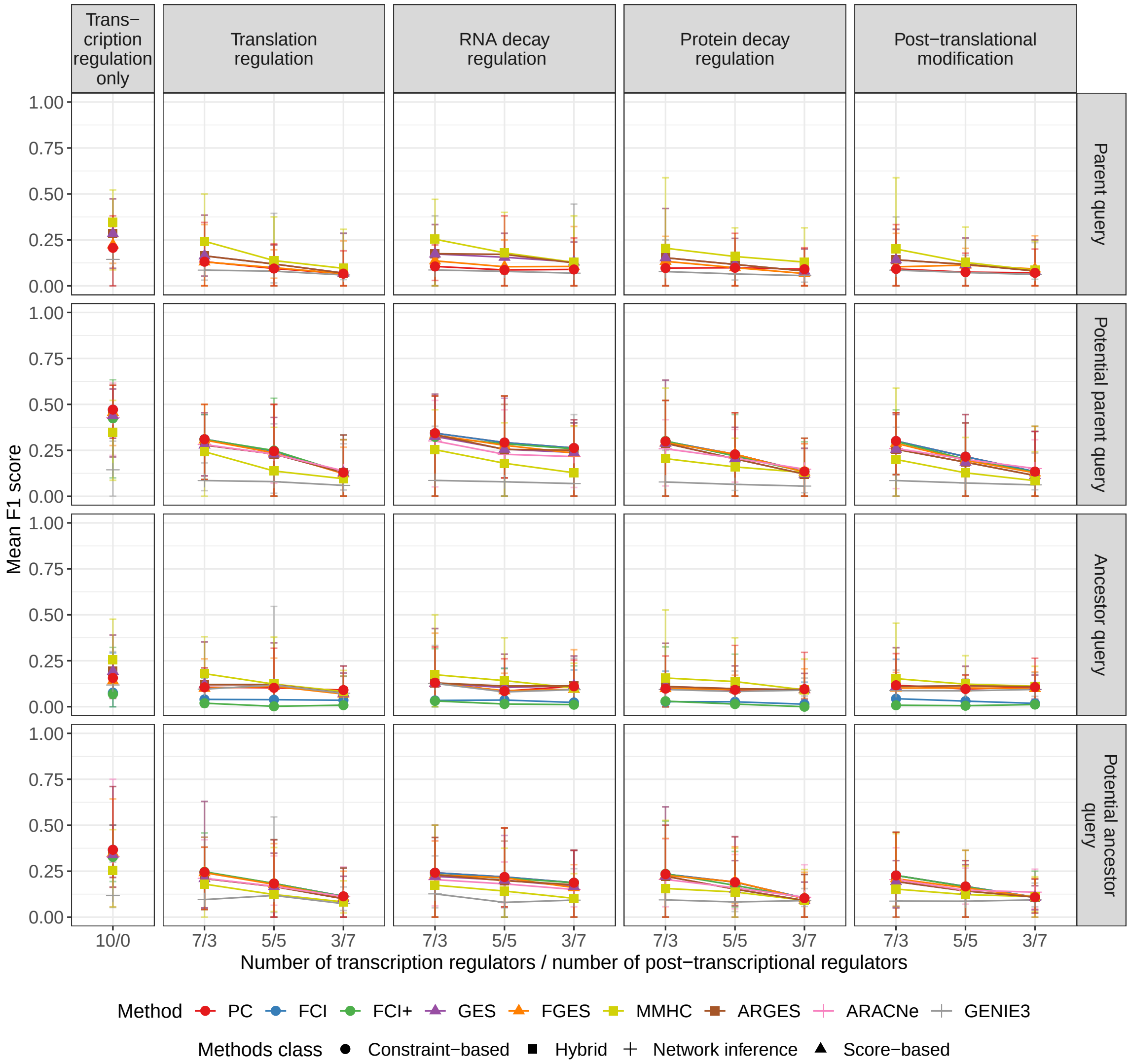


Figure 3.5: Mean $F_1$ score obtained with each method across the 20 simulated datasets for each configuration (x axis) and causal query (rows facetting), with the optimal values of the tuning parameters. The error bars represent the minimum and maximum $F_1$ score obtained across the 20 datasets of each configuration.

can observe higher average scores for the parent and potential parent queries compared to the ancestor and potential ancestor queries, as the last two require correct inference of paths in the graph and are thus more sensitive to errors in the graph reconstruction. In addition, a variable has more ancestors than parents in the graph, so there is more possibilities to make errors in the case of the ancestor query. Similarly, better scores are obtained for the potential parent/ancestor queries, compared to the parent/ancestor queries, as for the former there is no need to infer the correct orientation of the edges. I illustrate this point with an example network from configuration 1 (only transcription regulators). The consensus skeletons of the graphs returned by the different methods for each causal query are presented in Figure 3.8. The consensus skeleton is defined here as the graph of genes in which an

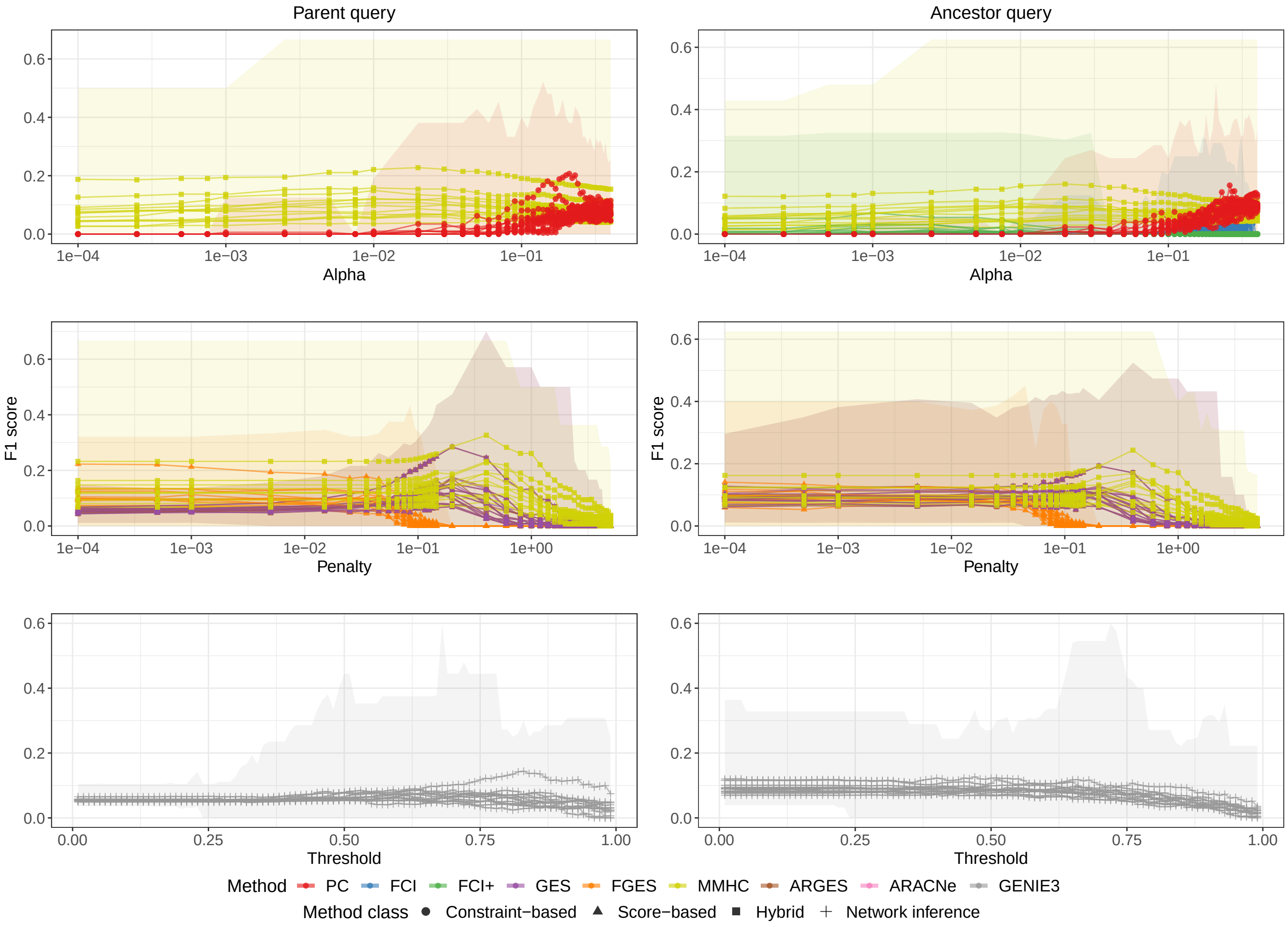


Figure 3.6: $F_1$ score obtained by the different methods as a function of the value of the tuning parameters for the parent (left panel) and ancestor (right panel) queries. The points represent the mean $F_1$ score obtained by the methods across the 20 datasets for each configuration, with each line corresponding to a different simulation configuration. The shaded areas represent the range of observed $F_1$ score values across all the configurations.

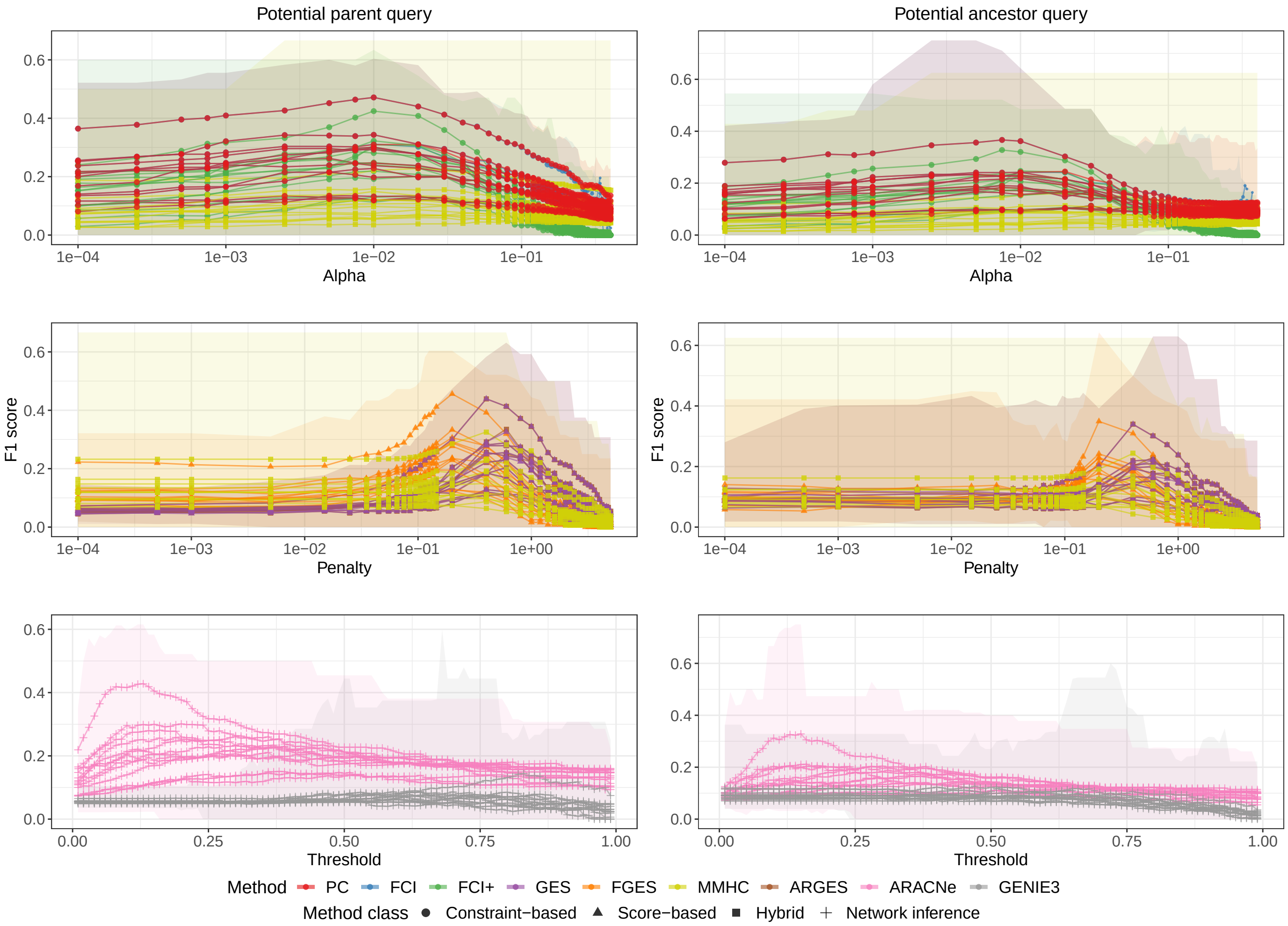


Figure 3.7: $F_1$ score obtained by the different methods as a function of the value of the tuning parameters for the potential parent (left panel) and potential ancestor (right panel) queries. The points represent the mean $F_1$ score obtained by the methods across the 20 datasets for each configuration, with each line corresponding to a different simulation configuration. The shaded areas represent the range of observed $F_1$ score values across all the configurations.

edge is added between two nodes if at least one of the investigated method inferred this edge. A weight is given to each edge, which corresponds to the number of inference methods that predicted the edge. We can see that in order to attain a good $F_1$ for the parent and ancestor query, the methods have to infer a lot of false positive relationships between the genes, in order to assess the orientation of the edges. However non oriented false positive edges are not penalised for this query. It is even more pronounced in the ancestor query, as the methods have to infer correctly oriented paths. On the contrary, less false positive relationships are inferred when the causal queries account for uncertainty in the edges orientation, i.e. for the potential parent and potential ancestor queries. Note that a number of false positive relationships are detected only by GENIE3 for the potential ancestor query. In Figure 3.8, the thick red lines correspond to non-existing causal relationships consistently inferred by all methods. This is caused by a correlation between two genes' RNA level, due to the random allocation of the genes' alleles amongst the individual, stochastic noise or low expression.

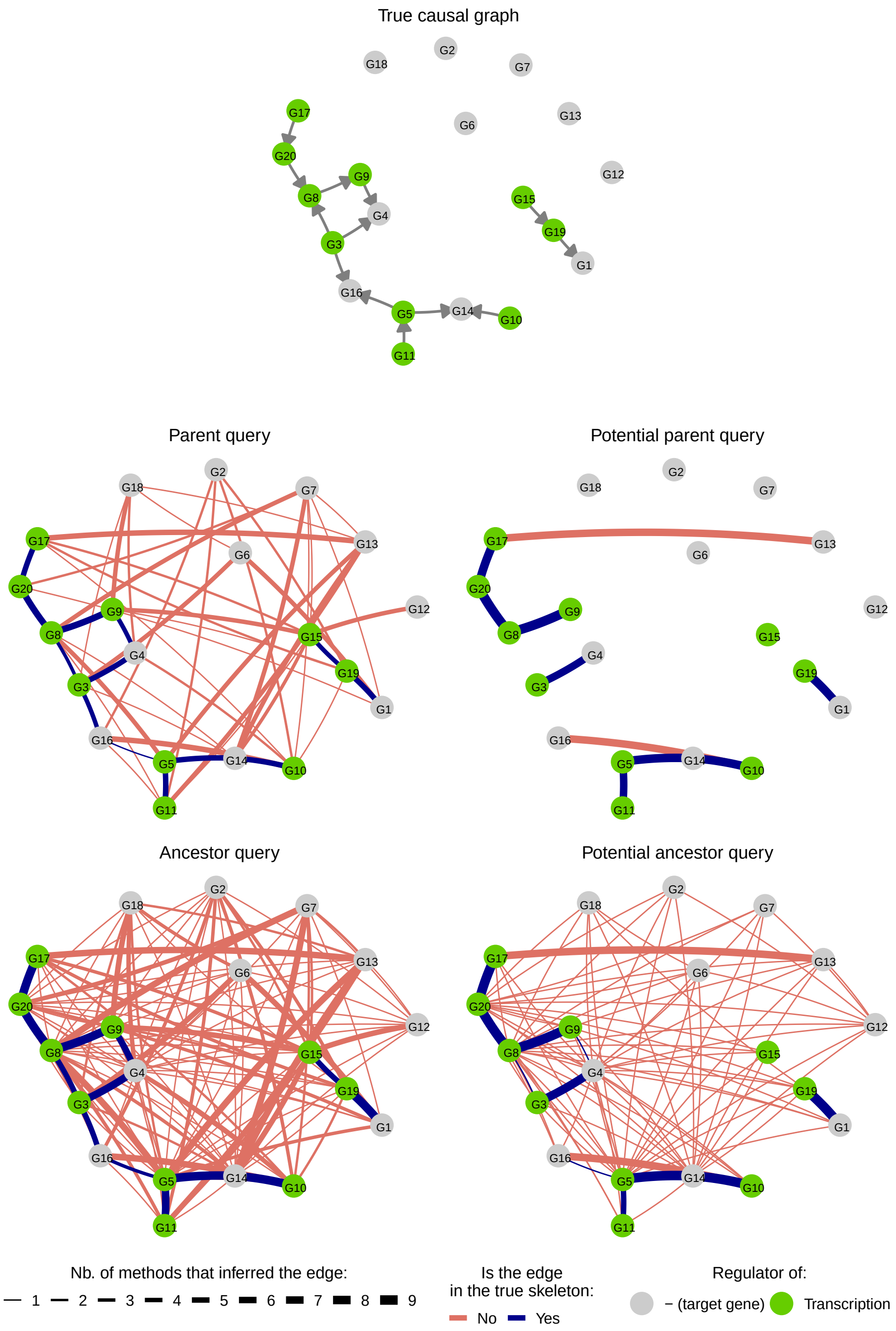
For the parent and ancestor queries, the hybrid method MMHC yields the best average $F_1$ scores for configurations with mostly regulation of transcription: for example in configuration 1 it obtains an average of 0.347 (SD 0.103) against an average of 0.284 (SD 0.113) for ARGES which is the second best. However its performance becomes similar to other methods when the number of post-transcriptional regulators in the system increases. For example, for the parent query, it reaches an average $F_1$ score of 0.095 (SD 0.084) for configuration 10 (with seven translation regulators), against 0.069 (SD 0.087) for the next best performer which is ARGES. GES and ARGES are the next best performers (mean of 0.284 and SD of 0.113 with GES for configuration 1 and parent query), followed closely by FGES and PC (mean of 0.223 and 0.207 respectively for configuration 1 and parent query, with SD 0.064 and 0.113 respectively). For the ancestor query, FCI and FCI+ yield the lowest mean scores across the configurations (e.g. for configuration 1 mean $F_1$ score of 0.077 and 0.068 respectively). For both parent and ancestor queries, MMHC ranks first by obtaining the best precision (i.e. fraction of the positive queries that are true), at the cost of a slightly lower recall than GES and ARGES (i.e. the fraction of true queries that are positive). An example is given in Supplementary Figure E.2 for the parent query and Supplementary Figure E.3 for the ancestor query. It can be seen that in order to infer and orient as many true edges as possible, the constraint-based and score-based methods as well as ARGES included a large number of false positive in their reconstructed graphs. In the case of constraint-based methods for example, they rely on the presence of v-structures in the reconstructed graph in order to orient as many edges as possible. Therefore if the reconstructed graph is very sparse, there will be less v-structures and therefore less edges oriented. For the ancestor query, the poor performance of FCI and FCI+ is due to a very low recall. In the example presented in Supplementary Figure E.3, this low recall is due to the orientation of the edges that could not be correctly inferred.

For the potential parent and potential ancestor queries, however, MMHC is outperformed by other methods, which all yield very similar mean $F_1$ scores, except for GENIE3. In the example presented in Supplementary Figures E.4 for the potential parent query and Supplementary Figure E.5 for the potential ancestor query, we can see that this is because the graphs inferred by the different methods all share the same skeleton. However, as MMHC returns a fully directed graph, it has to decide on a direction for each edge. Other methods returning CPDAGs or PAGs on the other hand can account for the uncertainty in edges direction. Therefore, MMHC is handicapped in its ability to detect potential parents and cannot be compared with methods including uncertainty. GENIE3 obtains the lowest scores across all configurations. This poor performance is explained by the high recall but low precision of the method, i.e. it infers too many causal relationships, mostly false positive (see the example in Supplementary Figures E.4 and E.5). The other network inference method investigated, ARACNe, performs on par with the other methods for queries that do not require orientation of the edges. This is interesting as it is not a method that infers causal relationships, but merely association.

For each method and configuration, the value of tuning parameters yielding the best $F_1$ score when comparing the skeleton of the inferred graphs to the skeleton of the true causal graph was selected. The resulting mean $F_1$ scores for each configuration are presented in Figure 3.9. We can see that all methods perform very similarly, except for GENIE3 which yield the lowest scores. The mean $F_1$ scores obtained for the configuration with only transcription regulation are quite high, at around 0.65, while they are below 0.5 for configurations with post-transcriptional regulation, and below 0.25 for configurations with seven post-transcriptional regulators. The very similar scores obtained across the different methods suggest that the methods detect the same causal relationships among the variables. The similarity between the skeleton of graphs inferred by any pair of methods across all 260 simulated datasets with the optimal values of the tuning parameters are presented in Figure 3.9. Interestingly, the three constraint-based methods, i.e. PC, FCI and FCI+, consistently infer the same skeleton graph across all simulated datasets. The MMHC algorithm also returns graphs with very similar skeletons to the constraint-based methods. Not surprisingly, GES and ARGES also infer very similar skeletons. In general, the similarity score across the causal inference methods is above 80%, indicating that, when selecting appropriate values for the different tuning parameters, the methods differ the most by the orientation of the edges in the reconstructed causal graph rather than by the edges themselves.

### 3.3.4 Reconstruction of post-transcriptional regulation

I next break down the performance of the methods in answering a causal query "is $A$ a (potential) parent/ancestor of $B$?" based on the biological role of $A$, i.e. whether it is a regulator of transcription or a post-transcriptional regulator. I want to see whether the different methods are able to detect causal relationships when the regulation between the genes occur at a different step than transcription. We could expect that methods accounting for hidden variables might perform better at detecting these types of interactions than methods assuming that all variables are observed. When focusing on


True causal graph

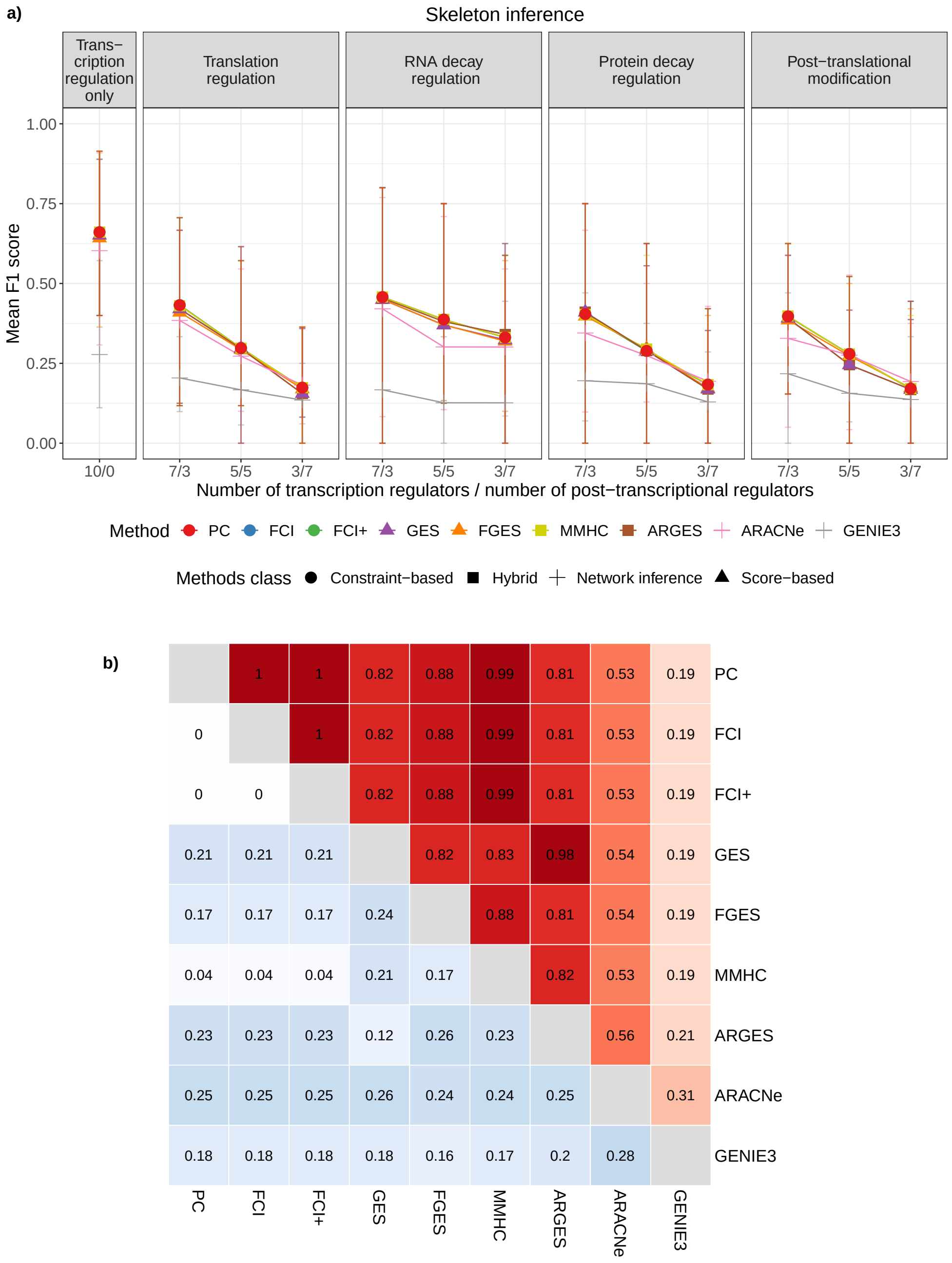
Parent query

Potential parent query

Ancestor query

Potential ancestor query

Figure 3.8: A network simulated for configuration 1, and the consensus skeleton of the graphs inferred by the different methods for each of the causal queries. The colour of the edges indicates whether the edge is present in the true skeleton (dark blue if yes, light red if not), and the width of an edge indicates the number of methods that inferred the presence of the edge. The biological role of genes in the network are indicated in colours, with regulators of transcription highlighted in green, while target genes are shown in gray.



| | PC | FCI | FCI+ | GES | FGES | MMHC | ARGES | ARACNe | GENIE3 |
|---|---|---|---|---|---|---|---|---|---|
| PC | | 1 | 1 | 0.82 | 0.88 | 0.99 | 0.81 | 0.53 | 0.19 |
| FCI | 0 | | 1 | 0.82 | 0.88 | 0.99 | 0.81 | 0.53 | 0.19 |
| FCI+ | 0 | 0 | | 0.82 | 0.88 | 0.99 | 0.81 | 0.53 | 0.19 |
| GES | 0.21 | 0.21 | 0.21 | | 0.82 | 0.83 | 0.98 | 0.54 | 0.19 |
| FGES | 0.17 | 0.17 | 0.17 | 0.24 | | 0.88 | 0.81 | 0.54 | 0.19 |
| MMHC | 0.04 | 0.04 | 0.04 | 0.21 | 0.17 | | 0.82 | 0.53 | 0.19 |
| ARGES | 0.23 | 0.23 | 0.23 | 0.12 | 0.26 | 0.23 | | 0.56 | 0.21 |
| ARACNe | 0.25 | 0.25 | 0.25 | 0.26 | 0.24 | 0.24 | 0.25 | | 0.31 |
| GENIE3 | 0.18 | 0.18 | 0.18 | 0.18 | 0.16 | 0.17 | 0.2 | 0.28 | |

Figure 3.9: a) Mean $F_1$ score obtained with each method across the 20 simulated datasets for each configuration, with the optimal values of the tuning parameters, for the skeleton reconstruction. The error bars represent the minimum and maximum $F_1$ score obtained across the 20 datasets for each configuration. b) Similarity between the skeletons of the graphs inferred by the different methods for the optimal values of the tuning parameters. Upper triangle (red scale): mean similarity score between the skeletons inferred by pairs of methods across the 260 simulated datasets. Lower triangle (blue scale): standard deviation of the similarity score between the skeletons inferred by pairs of methods across the 260 simulated datasets.

regulatory interactions stemming from a transcription regulator, the ranking of the methods observed in the previous section holds, and the mean $F_1$ scores obtained are higher than those computed over all genes regardless of their role. For example, for the potential parent query, the PC algorithm obtained for the second configuration (three translation regulators) a mean $F_1$ score of 0.31 (SD 0.102). When considering only interactions arising from regulation of transcription, the average $F_1$ score increases to 0.525 (SD 0.160). On the contrary, when considering post-transcriptional regulations, all methods perform very poorly, except when the post-transcriptional regulators affect their target's RNA decay. For example with the potential parent query for configuration 2, PC obtained an average $F_1$ scores of only 0.093 (SD 0.126). An example is shown in Figure 3.10, in which I present the mean $F_1$ score, precision and recall of each method for the potential parent query, depending on whether the query focuses on a transcription or post-transcriptional regulator. Similar conclusions can be drawn for the other causal queries; with very poor average $F_1$ scores being obtained with all methods when focusing on regulations stemming from post-transcriptional regulation. The mean recall for queries about post-transcriptional regulators is very low across all methods, except for GENIE3, however at the price of a low precision. It means that the methods are unable to correctly detect the causal relation between two genes when the regulator does not directly affect the level of mRNAs of the target. In the case of GENIE3, the higher recall observed is due to the fact that the method infers dense graph, with a lot of false positive as well. It is interesting to see that even methods that account for latent variables (FCI and FCI+) cannot detect relationships between genes that affect the protein level rather than RNA level of the genes. It might be because in this case the hidden variable (i.e. the protein level) acts as a missing link mediating the effect of one gene on another, rather than an unobserved variable affecting the two mRNA levels.

One possible option to improve the causal reconstruction in presence of post-transcriptional regulation is to make use of additional information about the genes expression. As the `sismonr` package simulates the abundance of both RNAs and proteins for each gene, the causal inference task was repeated on a few example networks from different configurations, using the protein level of the genes as a measurement for their expression (Figure 3.11 and Supplementary Figures E.6 and E.7). Remarkably, when using protein measurements for the causal inference on the example network investigated that contains only transcription regulation, the different causal inference methods were able to detect most of the regulations uncovered at the RNA level, with the added advantage that most false positive relationships detected with the RNA measurements were not detected at the protein levels. Indeed, in the absence of post-transcriptional regulation, the correlation between the RNA and protein levels of a given gene is high. As a results, transcription regulation can be detected both at the RNA and protein levels. Moreover, correlations between RNA levels of genes that arise due to noise or low abundance and not regulatory interactions tend to disapear at the protein level.

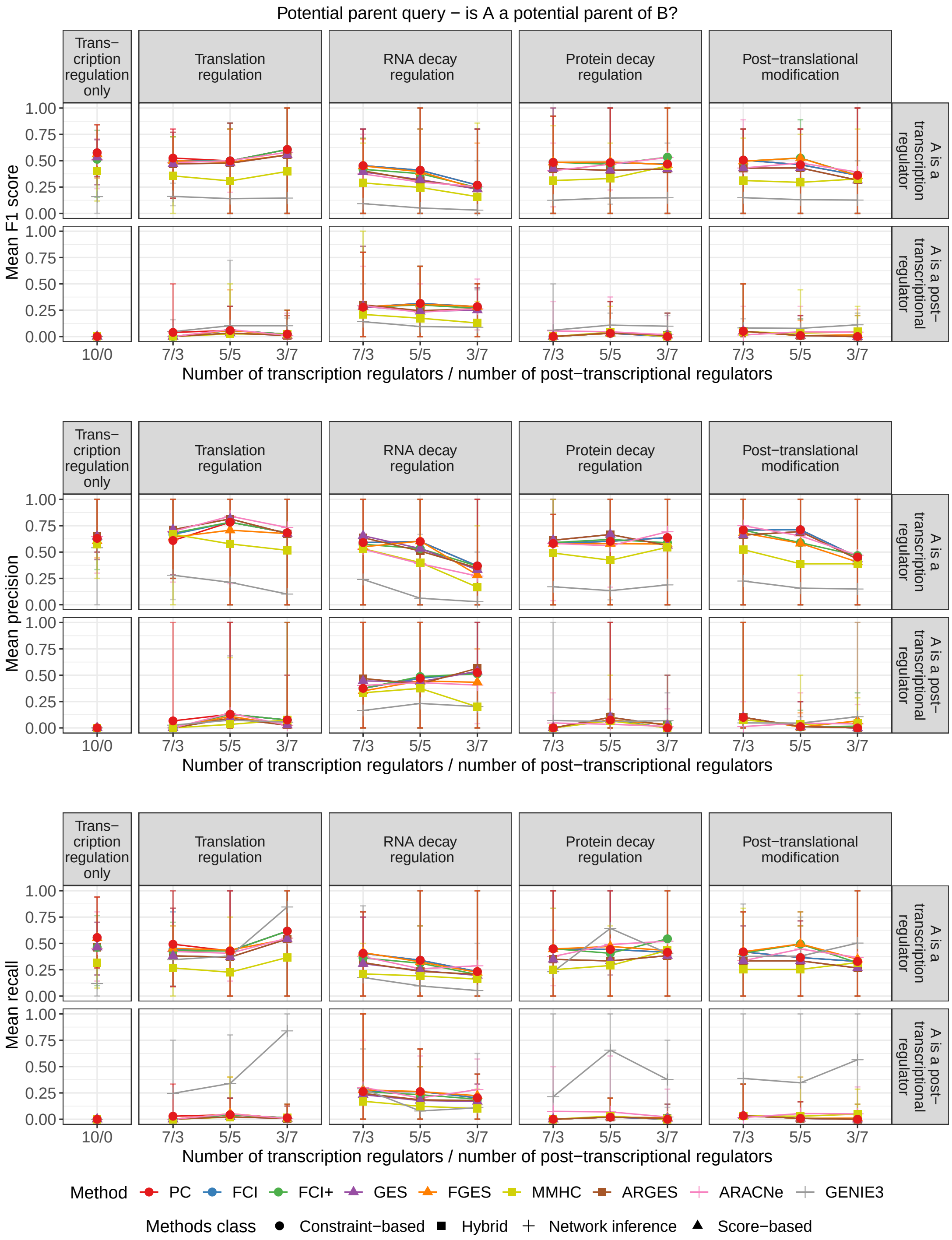
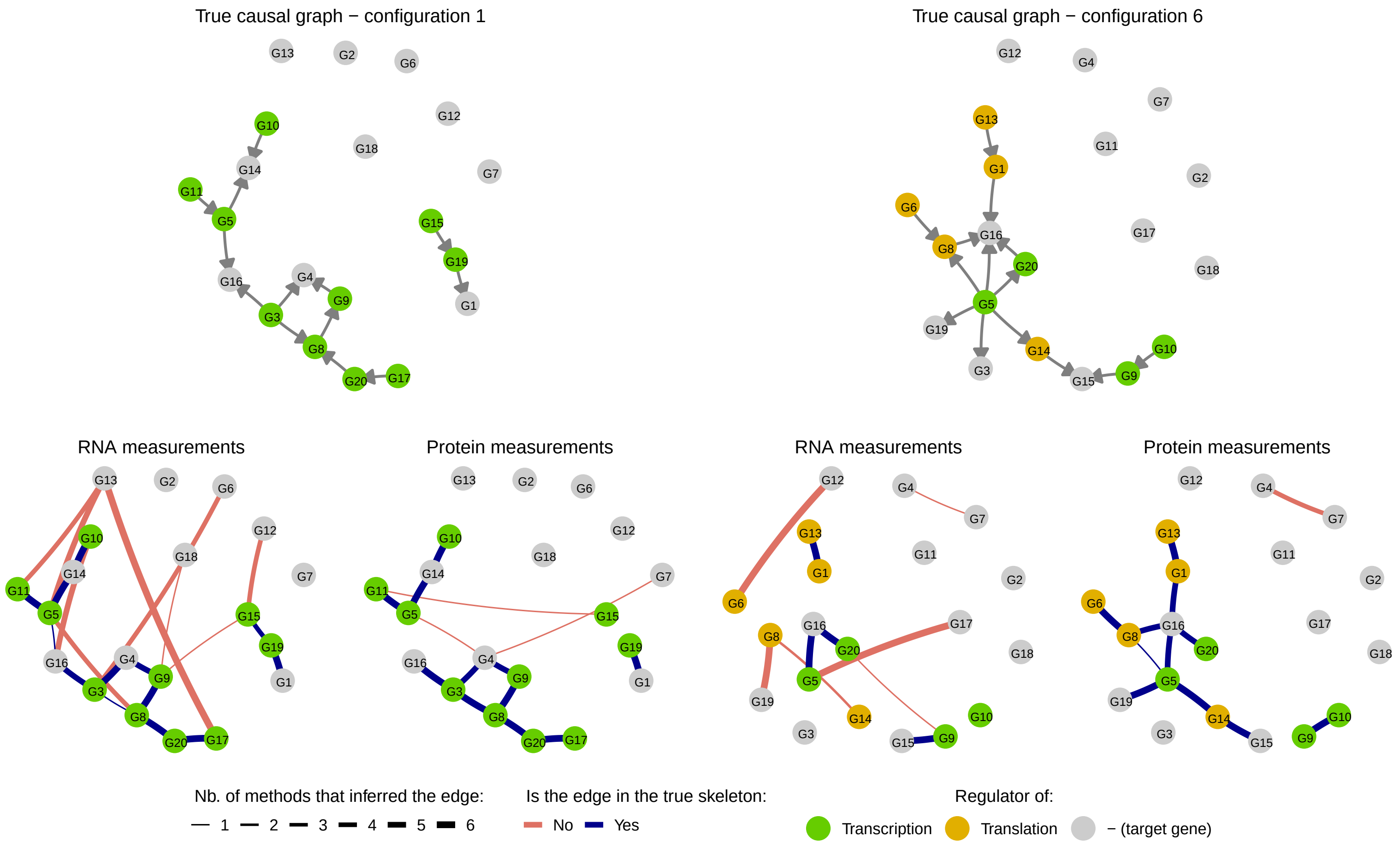


Figure 3.10: a) Mean $F_1$ score, precision and recall obtained with each method across the 20 simulated datasets for each configuration, with the optimal values of the tuning parameters, for the potential parent query. The values are separated according to whether the causal query implicates a transcription regulator or a post-transcriptional regulator. The error bars represent the minimum and maximum values obtained across the 20 datasets of each configuration.


True causal graph − configuration 1

True causal graph − configuration 6

RNA measurements

Protein measurements

RNA measurements

Protein measurements

G1 G2 G3 G4 G5 G6 G7 G8 G9 G10 G11 G12 G13 G14 G15 G16 G17 G18 G19 G20
Nb. of methods that inferred the edge:
1
2
3
4
5
6
Is the edge in the true skeleton:
No
Yes
Regulator of:
Transcription
Translation
− (target gene)

Figure 3.11: Two examples of network from configuration 1 (left) and 6 (right), with the consensus skeleton of the graphs inferred by the different methods (except GES, ARGES and GENIE3 that were excluded from this plot), when using RNA and protein measurements for the causal inference. The colour of the edges indicates whether the edge is present in the true skeleton (dark blue if yes, light red if not), and the width of an edge indicates the number of methods that inferred the presence of the edge. The biological role of the genes in the network are indicated in colours, with regulators of transcription highlighted in green, regulators of translation in orange, while target genes are shown in gray.

Interestingly, however, GENIE3, GES and ARGES performed poorly for this reconstruction task, and inferred many false associations. They were thus excluded from Figure 3.11 and Supplementary Figures E.6 and E.7 for the sake of clarity. In example networks that contain post-transcriptional regulation, performing the causal inference on protein measurements increased both the recall and precision of the methods. For each method and each causal query independently, the value of tuning parameter yielding the best $F_1$ score for each of the examples presented in Figure 3.11 was selected. For the parent query, the average $F_1$ scores obtained with the mRNA measurements across all methods (except GENIE3, GES and ARGES) were 0.171 and 0.065 for the example networks from configuration 1 and configuration 6, respectively. These values increased to 0.180 and 0.194, respectively, when using the protein measurements. For the example network from configuration 6, for the potential parent query, the average $F_1$ score increased from 0.242 with the mRNA levels to 0.547 with the protein levels. It can be noted for example that in the example network from configuration 6 (Figure 3.11), an edge is incorrectly inferred by most methods between genes 8 and 19 (G8 and G19 in the graph) when using RNA measurements. This is due to a regulator gene 5 (G5) regulating the transcription of both genes. On the contrary with the protein measurements these genes are not causally related in any of the methods' inferred graphs, and genes 5 and 19 are correctly linked. In addition, in the examples presented, the false positive relationships detected by most methods with one type of data (i.e. RNA or protein measurements) are different from the ones detected with the other type of data. An interesting avenue for causal inference of regulatory networks could thus be to weight the results of the inference obtained with both RNA and protein measurements in order to assign an higher confidence scores to relationships detected at both levels. However it is still uncommon to obtain for an experiment both mRNA and protein levels for the same set of samples.

## 3.4 Concluding remarks

The aim of this study was to investigate the performance of seven popular causal inference methods along with two network inference methods in reconstructing causal relationships among genes in the presence of post-transcriptional regulations. The evaluation aimed at assessing the ability of the methods to detect relevant signal in realistic settings in which typical assumptions of causal inference methods (e.g. causal sufficiency or acyclicity) are violated. RNA and protein levels were simulated for 20 networks of 20 genes for each of the 13 different gene regulation configurations, yielding a total of 260 simulated datasets. For each network, nine different causal and network inference methods were applied to the simulated RNA levels of the genes over a range of values for the different tuning parameters of the methods, and compared the precision and recall of each method in answering different causal queries for pairs of genes. The $F_1$ score was used to quantify the trade-off between precision and recall. I observed that choosing an appropriate value for the significance threshold $\alpha$ of conditional independence tests (for constraint-based methods) or for the penalty in the graph scoring criterion (for score-based methods) is crucial to the performance of the inference. In general, methods are better at answering queries about parental causal relationships as opposed to ancestral relationships, as the latter involves correctly inferring paths in the causal graph, and are thus more sensitive to reconstruction errors. Similarly, it is important for methods to be able to express uncertainty about the edges orientation, and methods such as MMHC or GENIE3 that reconstruct fully directed graphs are more likely to yield a lower performance when assessing the possible parents or ancestors of a given gene, compared to methods that return CPDAGs or PAGs. Overall, however, there is not much difference in the average $F_1$ scores obtained by the different methods over the simulated datasets in each configuration; except for GENIE3 which performed worst than other methods. This is specially true when evaluating the skeleton of the causal graphs returned by the methods, i.e. when ignoring the orientation of the causal edges. In particular, methods accounting for latent variables did not perform better than the other methods in presence of post-transcriptional regulations.

The performance of all methods decreased when the number of post-transcription regulators in the networks increases. The methods specifically struggle to correctly detect causal relationships between genes caused by post-transcriptional regulators, except to some extent when the regulators target RNA decay. The use of protein measurements for the causal inference was investigated, and I found that it could lead to increased precision and recall both for transcription and specially post-transcriptional regulations. Therefore, merging with some weighting scheme the causal inference results obtained from both RNA and protein measurements could increase the accuracy of reconstructed graphs. However, such an approach would still be affected by the presence of biological mechanisms triggering the activation of transcription factors or other regulators without concurrent changes in the expression of the corresponding gene. It can also be noted that rather than focusing on one causal inference method, it is advantageous to combine graphs inferred by several methods, as it has been encouraged by Vignes et al. (2011) for example. However, with these simulated datasets, different classes of methods inferred very similar relationships, and the difference between inferred graphs lays more in the orientation of the causal edges rather than in the skeleton of the reconstructed graphs.

There are a number of limitations to this comparison. The main one is that I generated networks of small size (20 genes in total) with only a subset of them involved in expression regulation. This is relatively small for biological networks, as the latter can include hundreds of genes. This could however be a good approximation of small modules within regulatory pathways. This choice stemmed from a need for computational feasibility. In order to test a number of simulation configurations, the networks had to be reasonably small so that the data could be generated and analysed in a timely manner. It would be interesting to repeat this comparison to see if the results obtained hold on a larger scale, i.e. with thousands of genes, which is closer to the number of genes measured in typical transcriptomics datasets. Similarly, a higher numbers of networks could be simulated per simulation scenario to improve variability estimates. I also used simulated gene expression that contains biological noise but no additional noise to emulate experimental variability. I am aware that this leads to optimistic performance scores, as technical noise that characterises experimental datasets hinders the reconstruction process. The scores obtained here must therefore be used with caution as they likely represent an upper limit to the performance of the methods for these scenarios.

It could also be interesting to compare the methods on other types of performance metrics. For example, the $F_1$ score is a special case of the $F_\beta$ score, which allows us to tune the relative importance of the precision and recall. For example, Ahmed et al. (2018) used instead the $F_{0.5}$ score, with which the precision is considered twice as important as the recall. This is of interest in the setting of molecular biology, as it is both expensive and time-consuming to experimentally validate inferred relationships between any two genes. It is therefore desirable to obtain inferred graphs with high precision, i.e. a low number of false positive, even at the expense of missing more true relationships, rather than a denser graph that captures more true relationships but in which a large fraction of the inferred edges are false positive. Other metrics that could be applied are methods that quantify the distance in graph space between the true and inferred graph, for example the Balanced Scoring Function (Constantinou, 2019). Future comparisons could include these metrics in order to draw wider conclusions. Also, it would be interesting to develop a score that accounts for the difference in the types of causal relationships (parental vs ancestral) that each method can reconstruct. This would circumvent the need for comparing the methods' ability to answer different causal queries. Moreover, I illustrated on some examples the interest of using protein measurements to reconstruct regulatory networks. I concluded that it could be interesting to compare the results of causal inference obtained on mRNA levels to those obtained with protein measurements. Future work could include performing a complete evaluation of the methods on protein measurements, to see if the conclusions drawn here on examples can be applied at a larger scale. In particular, it could be interesting to explore a weighting scheme to combine causal inference results from RNA and protein measurements. Another option would be to combine both datasets and perform the causal inference on mRNA and proteins as variables. This however raises the issue of proper normalisation in order to combine these two heterogeneous types of data. Additionally, as mentioned previously, this would require validation on experimental datasets including both mRNA and protein measurements. The use of prior knowledge to guide the reconstruction process could also improve the performance of the causal inference methods. One option would be to use information about the presence of binding motifs present upstream of a potential target, in order to confirm the presence of a potential regulatory relationship. Note that this would be limited to organisms for which a high-quality reference genome is available.

Lastly, it is to be noted that the choice of causal inference method to use will depend on the application and dataset considered. The results from these simulations have highlighted that methods accounting for latent variables might not provide a substantial advantage when reconstructing gene regulatory networks. However, there may be cases in which we can expect unmeasured confounders to impact several of the observed variables. In such cases, FCI or FCI+ might be more appropriate. This comes at the cost of an increased computational burden. In our experiments, GENIE3, FCI and FCI+ were the slowest methods amongst all evaluated methods. It can thus be expected that they will not be appropriate for datasets in which the number of variables is high, or in which the number of expected relationships is large (see for example Ogarrio et al. (2016) for indications of running time). For such large systems, faster methods such as MMHC or FGES might be preferred.

## 3.5 Acknowledgements

OAB wishes to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities, consulting support and/or training services as part of this research. New Zealand's national facilities are provided by NeSI and funded jointly by NeSI's collaborator institutions and through the Ministry of Business, Innovation & Employment's Research Infrastructure programme. URL `https://www.nesi.org.nz`.

# Chapter 4

# Investigating genotype-phenotype relationships for tuber bruising in autotetraploid potatoes using genomics and transcriptomics data

## 4.1 Introduction

Genome-wide association studies (GWAS) use statistical methods to detect causal variants in the genome that control a trait of interest. Such analyses make use of random genomic variations segregating in a relevant population, and seek statistical associations between these variations and the corresponding phenotypic values measured across a panel of individuals. This panel typically consists of a large population of individuals replicated over several environments, to separate the effect of environmental and genetic variations on the phenotype of interest. Although it is very difficult to detect the precise causal variant(s) influencing a considered trait, GWAS studies rely on linkage disequilibrium between the causal variants and nearby markers (i.e. non-random association of alleles at the considered loci, due to genomic proximity, selection, or other factors) whose genotype can be measured in order to detect genomic regions of interest for the phenotype. Such regions can then be further investigated to uncover genes or regulatory features controlling the trait, notably with the addition of transcriptomics data that provide information about the expression of genes into mRNAs. Combining GWAS results with analyses at the transcriptomics levels (such as differential expression analysis) allows to further investigate the effects of genetic mutations on the expression of genes and uncover the genes mediating the link between genotype and phenotype.

While such genomic association studies have been extensively investigated in diploid species, more work is still required for polyploid species, owing to their high genetic complexity. Among them, the potato, *Solanum tuberosum*, is a tetraploid crop of great economic importance. In particular, a better understanding of its agronomic and quality traits is crucial for breeding programs, in order to develop improved lines. This has been hindered by the complex genetic architecture of these traits, the high heterozygosity of the crop, and the lack of tools dedicated to polyploid genetic studies. Therefore, previous studies have made use of diploid species of potatoes (Bisognin et al., 2018; Hara-Skrzypiec et al., 2018; Kloosterman et al., 2013; Werij et al., 2007), biparental populations (Bisognin et al., 2018; Bradshaw et al., 2008; Kloosterman et al., 2013; Werij et al., 2007), restricted the analysis to dominant markers or diploidised the markers dosage (i.e. the number of alternate alleles carried by a sample) in order to use diploid tools (e.g. Malosetti et al., 2007), or focused on candidate genes for the association analysis (Baldwin et al., 2011; Carpenter et al., 2015; Fischer et al., 2013; Li et al., 2008; Schönhals et al., 2016; Schreiber et al., 2014; Urbany et al., 2011). Despite these drawbacks, numerous studies have investigated the genetic basis of important potato traits, notably through the use of GWAS (D'hoop et al., 2014; Rosyara et al., 2016; Schönhals et al., 2017; Schreiber et al., 2014; Sharma et al., 2018). Recently, a R package for GWAS analysis of tetraploid organisms, GWASpoly, was proposed by Rosyara et al. (2016). GWASpoly implements several genetic models to explain the impact of a marker's dosage on the phenotype. Importantly, it allows the inclusion of population structure in the model in order to correct for the impact of possible subpopulations on the resulting marker scores. Population structure can lead to spurious marker-trait associations as the presence of distinct subpopulations with different trait distributions and allele frequencies can lead to statistical association between the trait and unrelated markers (Bazakos et al., 2017; Michaelson et al., 2009). Rosyara et al. (2016), and later Sharma et al. (2018), investigated the impact of failing to account for population stratification on the results of an association analysis for potato.

Another critical step in a GWAS analysis is to select a significance threshold for the marker scores that properly accounts for multiple testing. Indeed, a statistical test of association with the phenotype is performed for each investigated genetic marker, resulting in tens to hundreds of thousands of p-values computed. Hence, proper correction of these p-values is necessary in order to avoid an inflation of false positive detections (Goeman & Solari, 2014). There are two popular approaches to multiple testing correction. The first consists in controlling the family-wise error rate (FWER), which corresponds to the probability of making at least one false detection across the entire set of tests. The Bonferroni correction is a popular example of such FWER-based correction (Bland & Altman, 1995). It is applied by dividing the intended threshold on type-I errors (false positives) by the number of tests performed. The resulting value is then used as a significance threshold for the markers' p-values. Such correction is however very conservative, as it is controlling the probability of making even one false positive detection across all the markers. Thus, the reduction in false positive comes at the expense of a large decrease in power to detect associations. To alleviate this problem, a second approach is to control the False Discovery Rate (FDR), which is the proportion of falsely declared positive (i.e. significant) tests among all tests coming out as positive. The Benjamini-Hochberg correction (Benjamini & Hochberg, 1995), sometimes referred to as FDR correction, and Storey's q-value (Storey, 2002), are often used to this end. Both rely on the assumption that, under the null hypothesis, p-values are uniformly distributed, and therefore the correction is performed based on the entire set of p-values across all markers. FDR-based corrections are less stringent and therefore more powerful than FWER-based corrections. However, the application of multiple testing correction to a GWAS setting must be handled with care. In particular, linkage disequilibrium between the investigated markers render the tested hypotheses dependent. Specifically, several closely located markers each with a significant score might not correspond to independent discoveries, but instead reflect the presence of a single genomic region impacting the phenotype (Brzyski et al., 2017). This complicates the definition of false discovery rate and therefore its estimation, potentially leading to under- or over-estimation of the global FDR (Korthauer, Chakraborty, et al., 2019). In response, methods have been proposed to control the overall FDR based on groups of tests representing single hypotheses (Barber & Ramdas, 2015; Brzyski et al., 2017; Siegmund et al., 2011). In the case of the Bonferroni correction, the impact of linkage disequilibrium on the correction can be alleviated by deriving from the correlation between markers the effective number of independent tests (Gao et al., 2010). This effective number then replaces the total number of tests when calculating the threshold to be used. The choice of a specific multiple testing correction method must also be guided by the relative cost of false positives and false negatives (Noble, 2009). If it is really important to detect as much true associations as possible, or if the follow-experiments used to validate the results are accessible enough that a certain fraction of false positives can be tolerated, then a FDR-based correction will ensure that more true signals are detected. If, on the contrary, validating detected associations with follow-up experiments is costly, it would be preferable to avoid too many false positive, and thus a FWER-based correction would be preferred. Note that currently the `GWASpoly` packages offers the Bonferroni, Storey's q-value and permutation-based corrections.

In this study, I focus on unravelling the genotypic component of tuber bruising. Tuber bruising, also termed enzymatic or blackspot bruising, is the browning of the tuber flesh below the skin following a mechanical shock. It is caused by the oxidation of phenolic compounds by polyphenol oxidases (PPO), leading to the formation of melanin pigments that give the brown colouration. Tuber bruising is an important quality trait as it affects the appearance and flavour of the tubers and thus impacts their fitness for sale. The development of potato lines that are more resistant to bruising is therefore a desirable objective for breeding programs, rendering the genetic analysis of this trait an important task. Previous studies have identified several candidate causal genes and QTL regions for the bruising phenotype. Urbany et al. (2011) focused on genes from the PPO family, in particular the POT32 gene which is a major isoform expressed in potato tubers, and other genes involved in cell structure and shape, membrane stability, as well as carbohydrate metabolism. Some of these candidate genes were identified by first comparing the concentration of the corresponding proteins to the trait (Urbany et al., 2012). They uncovered markers in several of these candidate genes associated with tuber bruising, notably in the PHO1A, LipIII27, PPO isoforms POT32 and potpoloxA, 4CL and HQT genes, located on chromosomes 2, 3, 5, 7 and 8. The influence of PPO enzymes on tuber bruising was already established by Werij et al. (2007), who uncovered a QTL region on chromosome 8 that co-localised with the POT32 gene sequence. Hara-Skrzypiec et al. (2018) later identified additional candidate genes on chromosome 5, 8 and 12.

Capture sequencing data targeting exonic regions was used on a breeding population of half-sibling families to uncover regions of interest for the bruising phenotype as well as other agronomic traits. Furthermore, I used RNA sequencing data to investigate how genetic variations impact the phenotype of interest. I demonstrate that even as capture sequencing only allows us to measure genetic variations in a subset of the genome, it is possible to uncover interesting and biologically meaningful genotype-phenotype associations, especially when combining the GWAS results with additional information, here transcriptomics data. Moreover, these associations were obtained with samples selected from a breeding program, demonstrating that available genomics and transcriptomics data from populations not specifically designed for association study can be used to uncover genomic regions of interest. Other contributions of the article are as follows: (i) I demonstrate the use of GWAS on a population of related individuals with complex population structure, (ii) I investigate the impact of the genetic model used for the association analysis, (iii) I propose a new visualisation to summarise and interpret GWAS results in combination with differential expression analysis. This study is a first step toward bridging the gap between genotype and phenotype by combining multi-omics data (genomics, transcriptomics, phenotypic), to ultimately unravel the mechanisms of potato tuber bruising.

## 4.2 Contribution

The plants trial, phenotype recording, and sample acquisition for genotyping and RNA sequencing were performed by Rebecca Bloomer, Katrina Monaghan, Susan Thomson and Samantha Baldwin (Plant and Food Research). The preliminary analysis of the genomics data (step G1 in Figure 4.1), as well as the preliminary processing of the RNA sequencing data (step T1 in Figure 4.1) were performed by Susan Thomson. I performed all subsequent analyses.

## 4.3 Materials and Methods

A schema of the analysis workflow used throughout this chapter is presented in Figure 4.1.

### 4.3.1 Plant materials and phenotyping

Seedlings were obtained by crossing 40 different parental lines (not all possible crosses were realised). Following seedling selection, progeny lines were cultivated in Lincoln, New Zealand and culled.

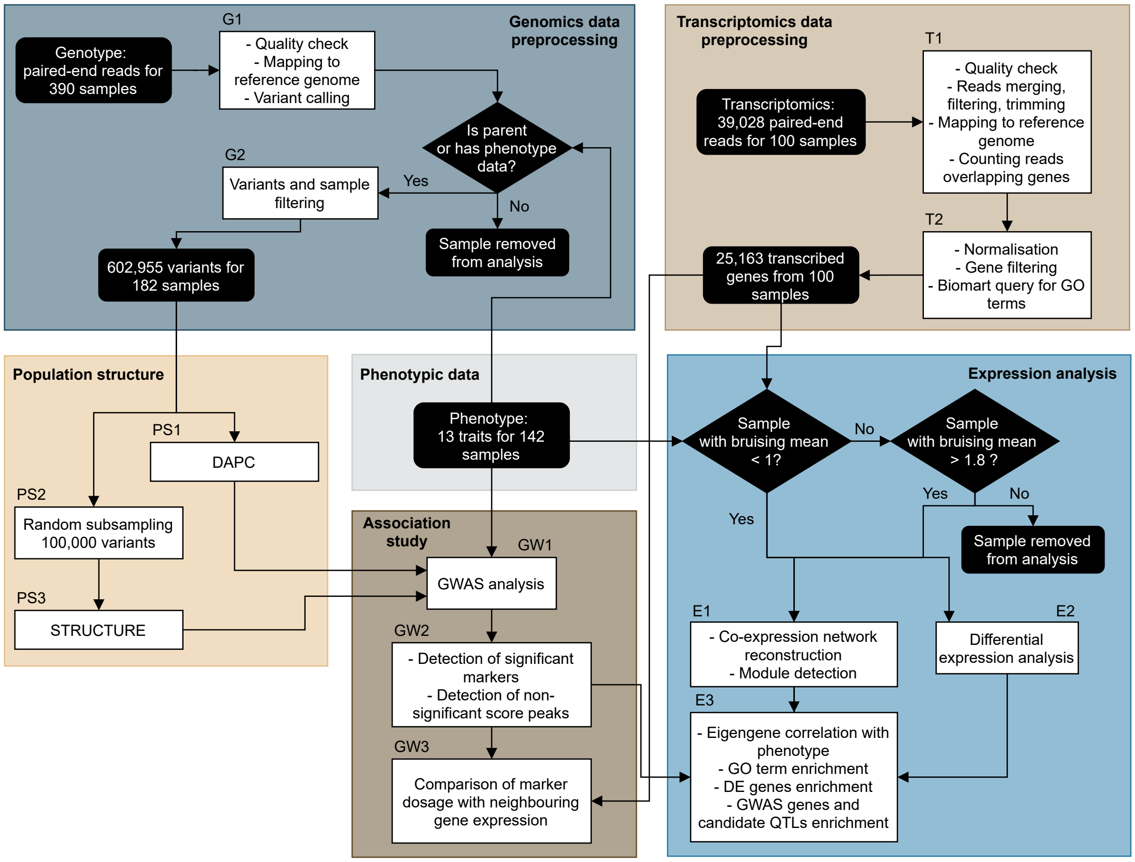

Genomics data preprocessing
G1
Genotype: paired-end reads for 390 samples
- Quality check
- Mapping to reference genome
- Variant calling
Is parent or has phenotype data?
Yes
No
G2
Variants and sample filtering
Sample removed from analysis
602,955 variants for 182 samples
Transcriptomics data preprocessing
T1
Transcriptomics: 39,028 paired-end reads for 100 samples
- Quality check
- Reads merging, filtering, trimming
- Mapping to reference genome
- Counting reads overlapping genes
T2
- Normalisation
- Gene filtering
- Biomart query for GO terms
25,163 transcribed genes from 100 samples
Population structure
PS1
DAPC
PS2
Random subsampling 100,000 variants
PS3
STRUCTURE
Phenotypic data
Phenotype: 13 traits for 142 samples
Association study
GW1
GWAS analysis
GW2
- Detection of significant markers
- Detection of non-significant score peaks
GW3
Comparison of marker dosage with neighbouring gene expression
Expression analysis
Sample with bruising mean < 1?
No
Sample with bruising mean > 1.8 ?
Yes
No
Yes
Sample removed from analysis
E1
- Co-expression network reconstruction
- Module detection
E2
Differential expression analysis
E3
- Eigengene correlation with phenotype
- GO term enrichment
- DE genes enrichment
- GWAS genes and candidate QTLs enrichment

Figure 4.1: Schema of the analysis workflow used throughout the chapter. Input/output datasets and endpoints are presented in black rounded boxes, analysis steps are shown in white rectangle boxes, and decision steps are drawn as black diamond boxes. The coloured rectangles outline different themes in the analysis.

For progeny lines from crosses involving either Crop20 or Crop52, the following phenotypes were recorded:

- Maturity (`Maturity`): score of plant senescence as a proxy for plant maturity (from 1 for a dead plant to 9 for a plant in full flower);
- Vigour (`Vigour`): score of above-ground biomass and amount of leaves of the plant (from 1 for a very small amount of biomass to 9 for a very vigorous plant);
- Yield (`Yield_tha`): amount of tubers produced by the plant in kilograms;
- Relative yield (`RelativeYield`): amount of tubers produced by the plant, relatively to a standard genotype (percentage);
- Sprouting (`spr`): score of dormancy; for each line, four to five tubers were harvested, and kept in the dark at ambient temperature. They were checked every two weeks for sprouting. The first tubers to sprout were given a score of 1, the ones sprouting two weeks later were given a score of 2, etc;

- Dry matter content (`Dmperc`): dry matter content of the tubers, quantified by measuring the specific gravity over 40 tubers;
- Sugar content (`sugar`): glucose content of tuber in mg/g fresh weight tuber, measured with a diabetes test kit;
- Market (`Market_tha`): amount of marketable tubers produced by the plant in kilograms, i.e. after removing tubers that are diseased, green, too small or with unusual shapes;
- Relative marker (`RelativeMarket`): amount of marketable tubers produced by the plant, relatively to a standard genotype (percentage);
- Percentage of saleable tubers (`Perc_saleable`): percentage of produced tubers that are marketable;
- General impression score (`General_Impression`): subjective breeder score (from 1 to poor line to 9 for very good line);
- Bruising score: For each progeny genotype, tubers from two biological replicates were harvested. Three tubers for each biological replicate were selected. Each tuber was bruised on two opposite sides by letting a lead weight fall from controlled height on the tuber. The zone of impact was recorded by inking the weight before bruising. After 24 hours, tubers were sliced at the site of bruising, and an image was taken of the bruising for the three tubers of the two biological replicates per genotype. A bruising score was attributed to each tuber upon visual inspection of the images, from zero (no visible bruising) to five (extensive dark bruise), in accordance to a predefined visual scale. Two scores were derived per genotype: the bruising mean score (`bruising_mean`) was computed as the mean bruising score over all tubers from both biological replicates (i.e. mean over six values). The bruising fraction score (`bruising_frac`) was calculated as the fraction of tubers (six in total) with a bruising score of three or more. This bruising experiment was designed to replicate as closely as possible commercial conditions, in which potatoes are susceptible to fall from a palet onto a rock or the edge of a piece of equipment.

In total, phenotypic measurements were obtained for the 13 above traits across 142 progeny samples. I used a Shapiro Wilk's test to assess whether the distribution of recorded values followed a normal distribution.

### 4.3.2 Genotyping

Samples were taken from young leaf tissue. Genotyping was performed by Rapid Genomics; further details can be found in Motazedi et al. (2018). Capture-sequencing data was obtained for 390 samples by paired-end Illumina HiSeq 2000 technology for 20,035 baits. The baits were designed to target random genes selected to reflect gene density in the corresponding genomic region. The resulting 100bp paired-end reads were processed as follows (box G1 in Figure 4.1). First, they were assessed for quality control using FasQC v0.11.2 (Andrews et al., 2018). In particular, FastQC Screen v0.5.2 (Wingett & Andrews, 2018) was used to check for contamination. Trimming and filtering was performed with Trimmomatic v0.36 (Bolger et al., 2014). The paired reads were aligned to the reference genome PGSC-DM v4.03 (Sharma et al., 2013; Xu et al., 2011) using BWA mem v0.7.15 (Li, 2013), then sorted and converted using SAMtools v1.3.1 (Li et al., 2009). Picard-tools v2.10.1 ("Picard toolkit," 2019) was used to add group information to the reads. TargQC v1.4.4 and BEDTools v2.21.0 (Quinlan & Hall, 2010) were used to estimate the on/off target rates after mapping. Variant calling was performed with FreeBayes (Garrison & Marth, 2012).

Variant data pre-processing was performed with the python library `scikit-allel` (Miles et al., 2020). Out of the 390 genotyped samples, 184 samples that either had phenotypic measurements or were parents of progeny samples with phenotypic measurements were retained for further analysis. Only biallelic SNPs that were polymorphic in the considered samples were retained for the analysis. Variants were filtered out if they didn't meet each of the following conditions: (i) less than 10% of samples with missing genotype, and (ii) QD score of two or more. The QD score of each variant was computed as the quality score of the variant divided by the sum of its coverage for non-homozygous samples (box G2 in Figure 4.1). In addition, samples were filtered out if they had missing values for more than 10% of the variants. Problematic variants were defined as variants for which one allele was observed in at least one progeny sample but not in the parents.

### 4.3.3 Transcriptomics

Two hours after the bruising experiment, samples were taken from one bruised side of each tuber, and the samples obtained from the three tubers of a biological replicate were pooled and snap-frozen, grinded, and stored at -80°C. One biological replicate for each genotype was chosen for RNA measurement. Transcriptomics measurements were obtained for 100 samples (all progeny samples of Crop52) using an Illumina NovaSeq platform, with a Lexogen SENSE mRNA polyA library. A copy of the RNA extraction protocol can be found In Appendix F, Section F.1. Resulting reads were processed as follows (box T1 in Figure 4.1). Reads quality was assessed with FastQC v0.11.7 (Andrews et al., 2018), and rRNA contaminants were removed using SortMeRNA (Kopylova et al., 2012) using default parameters. Trimming was performed with BBtools v37.93 BBDuk (Bushnell, 2016), with the flag `forcetrimleft` set to 15, `trimpolyg` to 30, `trimpoly` to 30, `k` to 13, `qtrim` to `r`, `trimq` to 10 and `minlength` to 50. The reads were then aligned to the reference genome PGSC-DM (genome assembly version v4.03 and gene annotation v4.03) (Sharma et al., 2013; Xu et al., 2011) and the number of reads overlapping each gene in the reference genome was computed, using STAR v2.6.1 (Dobin et al., 2013), with the following flag settings: `alignMatesGapMax` to 20000, `outQSconversionAdd` to -31, `outFilterScoreMinOverLread` to 0, `outFilterMatchNminOverLread` to 0, `outFilterMatchNmin` to 40, `alignSJDBoverhangMin` to 10, `alignIntronMax` to 200000, `quantMode` to `GeneCounts`. Genes for which a read count was computed are thereafter referred to as transcribed genes (although the term gene will also be used when no confusion is possible).

Next, the remaining processing was performed with R (box T2 in Figure 4.1). The raw read counts were converted to RPKM counts (to correct for transcript length and library size) by dividing the read count of transcribed gene $i$ for sample $j$ by the length of the gene in kilobases and the library size per million (i.e. total read counts divided by $1.10^6$) of sample $j$. For comparison, the raw read counts were also normalised using the Variance Stabilising Transformation method (Anders & Huber, 2010), implemented in the R package `DESeq2` (Love et al., 2014). The Variance Stabilising Transformation is estimated by modelling genes' raw read counts with a negative binomial distribution accounting for the library size of the samples as well as a gene-specific dispersion and genes' length. This ensures that the resulting normalised counts are approximately homoskedastic and not impacted by library size and gene length. Transcribed genes were discarded if their expression didn't meet the following conditions: (i) total (raw) read count across all samples higher than 10, (ii) RPKM count across all samples higher than 0.75 and (iii) less than 95% of samples with a raw read count inferior to five. The threshold of 10 for the first condition was chosen according to similar studies in the literature. The threshold of 0.75 for the second condition was chosen to yield a similar percentage of rejected genes to the first condition. In practice, transcribed genes that did not fulfil one of the first two conditions also failed the third one, which was the most conservative. The `BiomaRt` package (Durinck et al., 2009) was used to retrieve the description of genes and associated GO annotations from their ensembl ID.

### 4.3.4 Population structure analysis

The population structure amongst the potato samples was investigated using discriminant analysis of principal components or DAPC (Jombart et al., 2010 – box PS1 in Figure 4.1) and STRUCTURE (Pritchard et al., 2000 – boxes PS2 and PS3 in Figure 4.1), using as input the dosage of the retained variants across the samples. Briefly, STRUCTURE is a Bayesian model-based clustering method that investigates the population structure amongst a set of samples using multilocus genotype data. DAPC is a multivariate approach that clusters samples in different groups and seeks a low-dimensional space that maximises the variance between groups while minimising the variance within groups. For both methods, all genotyped samples were used in the analysis (i.e. 182 samples), including samples for which no phenotype was recorded.

For computational efficiency, 10,000 (out of 602,955) variants were randomly sampled (box PS2 in Figure 4.1) and used to infer population structure using STRUCTURE (box PS2 in Figure 4.1). The number of subpopulations was estimated by running STRUCTURE with a number of subpopulations K varying from 1 to 10, using the admixture model (samples can be an admixture of the subpopulations) and the correlated allele frequency model (assumes similar allele frequencies across the subpopulations), with a burn-in length of 10,000 and a run length of 20,000. Each run for a particular value of K was replicated five times, using the R package `ParallelStructure` (Besnier

& Glover, 2013). I selected the optimal value of K by inspection of the $\Delta K$ values (Evanno et al., 2005), which represents the ratio of the average over the five runs of the second derivative of the likelihood (with respect to K) over the likelihood variance. As two values of K yielded similar $\Delta K$, the smallest of the two was chosen as the optimal number of subpopulations amongst the samples. To correct for the effect of label switching, i.e. the fact that in different runs of STRUCTURE with identical parameters, the uncovered subpopulations are not assigned the same label, the correlation between the computed samples posterior membership probabilities of the five runs of STRUCTURE (with the optimal value of K) was used to cluster the populations discovered with each run. The final posterior membership probabilities of each sample (i.e. the estimated probability that the sample belongs to each of the K inferred subpopulations) was computed as the mean of the estimated posterior membership probabilities across the five runs.

The DAPC analysis was performed using the R package `adegenet` (Jombart & Ahmed, 2011). First, principal component analysis (PCA) was performed over the samples' variants dosage, and all 181 principal components (PCs) were used to perform a k-means clustering of the samples. The detected clusters correspond to the different subpopulations detected by the algorithm amongst the investigated population. In order to estimate the optimal number of clusters amongst the samples, the k-means clustering was repeated for a number of clusters k ranging from 1 to 40, and the value yielding the lowest BIC score was retained as the optimal number of clusters. Next, a discriminant analysis was performed on the principal components obtained with the PCA. In order to avoid overfitting, only a subset of the principal components were retained for the discriminant analysis. The optimal number of principal components to retain was estimated by cross-validation: 90% of the samples were randomly assigned to a training set, and the discriminant analysis was performed on the training set. The remaining 10% of the samples (validation set) were used to estimate the performance of the resulting model in assigning the samples to their respective cluster. The cross-validation scheme was repeated with a different number of principal components used for the discriminant analysis, and the optimal number of PCs to retain was estimated as the highest value for which the performance was still high, and for which the RMSE (Root Mean Squared Error) was still low. A measure similar to the posterior membership probability of samples was computed, which reflects the probability of each sample to be assigned to each of the k clusters given its estimated coordinates in the discriminant analysis space.

### 4.3.5 GWAS analysis

The association analysis was performed using the `GWASpoly` package (Rosyara et al., 2016 – box PS1 in Figure 4.1). `GWASpoly` models the effect of the SNPs on the phenotype using the following linear mixed model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{S}\boldsymbol{\tau} + \mathbf{Q}\mathbf{v} + \mathbf{Z}\mathbf{u} + \epsilon$$

where $\mathbf{y}$ is a $n \times 1$ vector of measured phenotypic values; $\boldsymbol{\beta}$ is a $p \times 1$ vector of covariate effects, with $\mathbf{X}$ the $n \times p$ covariate incidence matrix; $\boldsymbol{\tau}$ is a $s \times 1$ vector of SNP effects, with $S$ is the $n \times s$ incidence matrix; $\mathbf{v}$ is a $q \times 1$ vector of subpopulation effects, with $\mathbf{Q}$ the $n \times q$ incidence matrix relating the samples to the subpopulations (in practice $\mathbf{Q}$ is the matrix of samples posterior membership for the different subpopulations); $\mathbf{u}$ is a $n \times 1$ vector of random polygenic effects with variance $Var(\mathbf{u}) = \sigma_g^2 \mathbf{K}$, where $\sigma_g^2$ is the genetic variance and $\mathbf{K}$ the $n \times n$ kinship matrix (i.e. matrix describing the relationship between samples), with $\mathbf{Z}$ the $n \times n$ genotype/phenotype incidence matrix; and $\epsilon$ is a $n \times 1$ vector of residuals, with variance $Var(\epsilon) = \mathbf{I}\sigma_e^2$, where $\sigma_e^2$ corresponds to the residual variance. The $F$-test is used to compute a p-value for each marker, with the null hypothesis that all SNPs effects (i.e. all values in the vector $\boldsymbol{\tau}$ of SNP effects) are equal to zero. I refer thereafter to the negative log10 of such p-value as the marker score.

Six possible genetic models, that describe the impact of a marker's dosage on the phenotype, were considered, namely:

- the general model: each dosage can have a different arbitrary impact on the phenotype;
- the additive model: the effect of the marker on the phenotype is linear with the marker's dosage;
- the simplex reference dominant model, denoted as 1-dom-ref: the effect of the marker on the phenotype is determined by the presence of at least one copy of the reference allele;
- the simplex alternate dominant model, denoted as 1-dom-alt: the effect of the marker on the phenotype is determined by the presence of at least one copy of the alternate allele;
- the duplex reference dominant model, denoted as 2-dom-ref: the effect of the marker on the phenotype is determined by the presence of at least two copies of the reference allele;
- the duplex alternate dominant model, denoted as 2-dom-alt: the effect of the marker on the phenotype is determined by the presence of at least two copies of the alternate allele.

In addition, similarly to Sharma et al. (2018), different population settings, that correct for population structure and/or individual relatedness were evaluated:

- Naive model: no correction for population structure nor individual relatedness (kinship matrix $\mathbf{K}$ set to a matrix of zeros with diagonal elements set to one);
- K-only model: accounts for individual relatedness only by computing the kinship matrix as the realised relationship matrix $\mathbf{K} = \mathbf{M}\mathbf{M}^T$ (VanRaden, 2008), where $\mathbf{M}$ is the genotype matrix of variants dosage;
- Q only model: accounts for population structure only by adding the samples posterior membership probabilities for the uncovered subpopulations as covariates in the analysis.
- K+Q model: accounts for both individual relatedness and population structure as described above.

In order to correct for population structure, the samples posterior membership probabilities computed with STRUCTURE and DAPC were used, which gave six different population settings: naive model, K model, $Q_{STRUCT}$, $Q_{DAPC}$, K+$Q_{STRUCT}$ and K+$Q_{DAPC}$. This yielded a total of 36 different GWAS settings (six genetic models × six population settings).

For each GWAS setting, the ability of the model to control for false positive and false negative was evaluated by computing the inflation factor, which quantifies the deviation of the estimated marker scores from their expected values. For each setting, the inflation factor was computed as the regression coefficient of the markers' observed scores over their expected scores under the null hypothesis.

To correct for multiple testing, the Bonferroni correction was used for each GWAS setting to compute the significance threshold for markers score with a type-I error level of 0.05. The Bonferroni correction was preferred to the Storey's q-value correction due to the linkage disequilibrium between investigated markers. Indeed, as markers were obtained from baits designed to capture specific regions of the genome, we expect groups of markers originating from a single bait to represent one unique hypothesis. Therefore, the calculation of the FDR as performed by the Storey's q-value correction will be biased. The permutation-based correction was not used due to its computational burden. Markers were then retained as significant QTLs if and only if (i) their computed score was above the significance threshold set with the Bonferroni scheme and (ii) the inflation factor yielded by the corresponding GWAS setting was between 0.95 and 1.1 (box GW2 in Figure 4.1). In addition, any marker whose score was above the $99.99^{th}$ percentile of the markers score distribution (for this GWAS setting) was retained as high-scoring marker.

The dosage of high-scoring markers was compared to the expression of the transcribed genes in which they are found using a correlation test as well as a Kruskal-Wallis test (box GW3 in Figure 4.1). The later compares the median of the transcribed gene's expression between samples grouped based on the dosage of the marker. Contrary to the correlation test, the Kruskal-Wallis test assesses whether at least one of the dosage groups has a different median gene expression than the others, which allows us to detect changes in gene expression that are non linear with the dosage of the marker. The FDR correction (Benjamini & Hochberg, 1995) was used to correct resulting p-values for multiple testing; it was preferred to the Bonferroni correction in order to increase the detection power. For some genes of interest, the association between the marker's dosage and neighbouring genes' expression was also assessed, using the same methodology. The analysis was also performed with the raw, RPKM and VST counts for comparison, however only the results obtained with the RPKM data are shown.

### 4.3.6 Differential expression analysis

As the RNA measurements were done on one biological replicate for each genotype, the bruising mean score of each sample was computed as the mean of the bruising score obtained for the three tubers of the corresponding biological replicate. For the differential expression analysis (box E2 in Figure 4.1), all samples that had a bruising mean score inferior or equal to one were first classified as the “low bruising” group. This yielded a “low bruising” group of 41 samples. Consequently, the 41 samples with the highest bruising mean score were selected and classified them as “high bruising” group. The remaining samples were removed from the analysis. The `DESeq2` package (Love et al., 2014) was used to perform the differential expression analysis, with the untransformed read counts of the transcribed genes as input. The `DESeq2` packages computes the Variance Stabilising Transformation as described in Section 4.3.3, and then tests for each gene the null hypothesis that the expression strength of the gene is identical between the two bruising groups using a Wald test on the log fold-change of the gene. The transcribed genes p-value obtained were corrected for multiple testing using the FDR correction (adjustment performed by the `DESeq2` package) and the Independent Hypothesis Weighting (IHW) correction (Ignatiadis et al., 2016) for comparison, which uses the average normalised read counts of each transcribed gene across samples as a weight applied to the gene’s p-value to improve the power of detection of differentially expressed genes. Both methods rely on controlling the False Discovery Rate, which was favoured here to detect as many true differentially expressed genes are possible. Transcribed genes were considered as significantly differentially expressed if either their FDR-adjusted or IHW-adjusted p-value was below 0.05. Enrichment of the molecular function- and biological process-related GO categories for differentially expressed genes was computed with the `gage` R package (Luo et al., 2009). `gage` assesses whether a given set of genes (in this case, a set of genes grouped in the same GO category) is enriched for differentially expressed genes by comparing its mean gene score to the mean gene score of all the genes not in the set (referred to as the background set). Here, I used the differential expression score of the genes, i.e. -log10 of their adjusted p-value, as the gene scores. The comparison is done using a prototype two-sample t-test, which contrasts the set of genes of interest to a virtual random set of genes from the background set of the same size. The resulting set p-values were corrected for multiple testing via the FDR correction. GO terms were detected as significantly enriched for differentially expressed genes if the corresponding adjusted p-value was below 0.05. In addition, the distribution of genes’ distance to the nearest GWAS high-scoring marker was compared between differentially expressed genes and non-differentially expressed genes using a two-sample Wilcoxon test.

### 4.3.7 Network co-expression reconstruction

The co-expression network of transcribed genes was reconstructed from transcriptomics measurements with the R package `WGCNA` (Langfelder & Horvath, 2008), using the VST transformed counts restricted to the samples used in the differential expression analysis (box E1 in Figure 4.1). I briefly present the main steps of the analysis thereafter. First, the similarity matrix (giving the similarity between each pair of transcribed genes) is computed, as the absolute value of the correlation between each pair of transcribed genes. The adjacency matrix is obtained by elevating each value in the similarity matrix to the soft thresholding power $\beta$, which increases the similarity between transcribed genes with high correlation while further reducing the similarity between transcribed genes with low correlation.

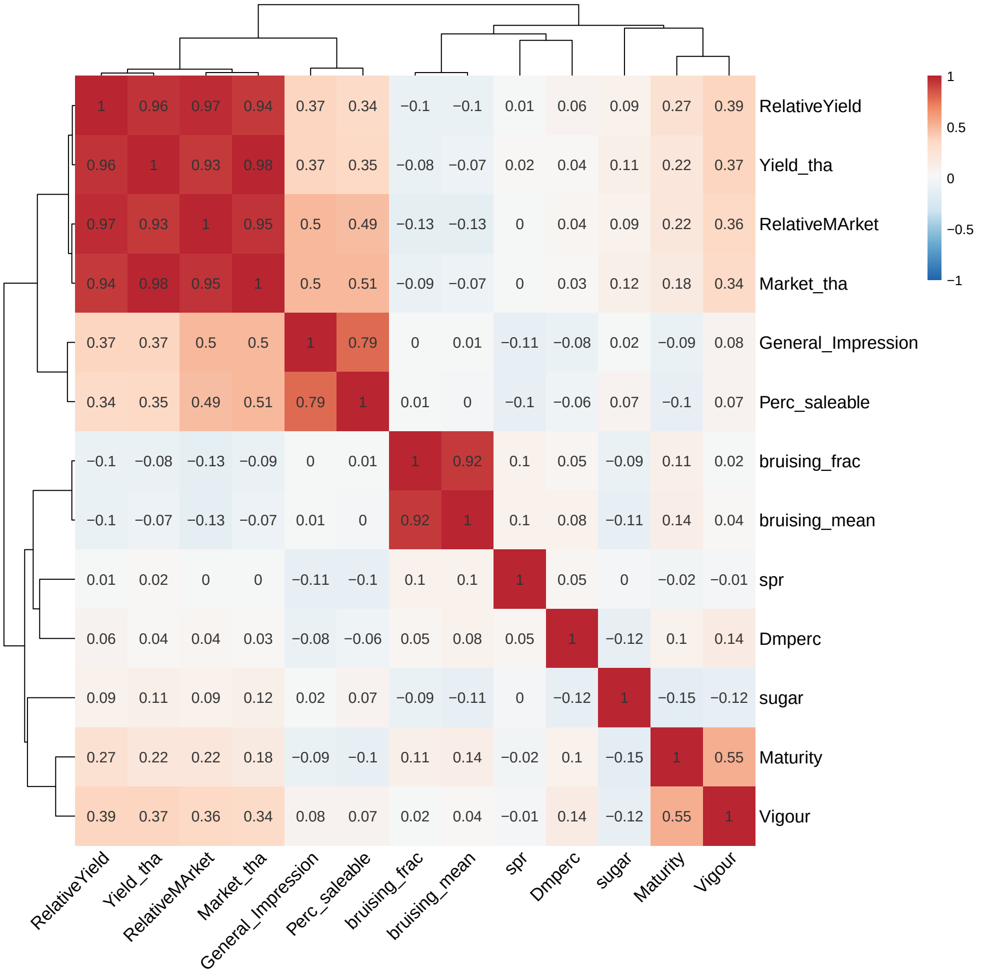
This soft thresholding power $\beta$ is selected based on how well the resulting network approximates a scale-free topology (Zhang & Horvath, 2005). Here a value of six was chosen for $\beta$. Next, the adjacency matrix is used to compute a topological overlap matrix (TOM) that describes for each pair of transcribed genes how many common neighbours they share in the adjacency graph. Two transcribed genes that share a lot of common neighbours will be assigned a high TOM score (Yip & Horvath, 2007). The dissimilarity between pairs of transcribed genes is then computed as one minus their topological overlap. The later is used for module detection in the inferred network, using hierarchical clustering. Modules can be extracted from the resulting dendrogram using the dynamic tree cut method (Langfelder et al., 2008). The `deepSplit` parameter was set to three, and the minimum cluster size to 10, in order to detect small modules of highly coexpressed transcribed genes. Lastly, modules with transcribed genes with highly similar expression profiles are merged. In the present case, because the number of transcribed genes is large, the genes were first split into blocks using a rough clustering method, and the TOM and resulting modules were separately computed for each block of genes.

An eigengene was computed for each module, which is the first principal component of a PCA performed on the expression of the transcribed genes within the module (box E3 in Figure 4.1). The eigengenes provide a summary of the expression profile of the transcribed genes within each module. A correlation test was used to detect modules whose eigengene is significantly correlated with the bruising mean score of the samples. The resulting p-values were corrected for multiple testing using the FDR correction. The enrichment of each module for genes containing high GWAS-scoring markers, as well as for previously detected candidate QTL genes, was investigated with a Fisher's exact test, through the function `runGSAhyper` from the `piano` Bioconductor package (Väremo et al., 2013). Modules that were significantly enriched for transcribed genes with a high differential expression score were detected using the `gage` package (Luo et al., 2009), as described in the previous section. Modules were detected as significantly enriched for differentially expressed genes if the corresponding adjusted p-value was below 0.05. A Fisher's exact test was also used to compute the enrichment of the detected modules for GO terms.

## 4.4 Results and Discussion

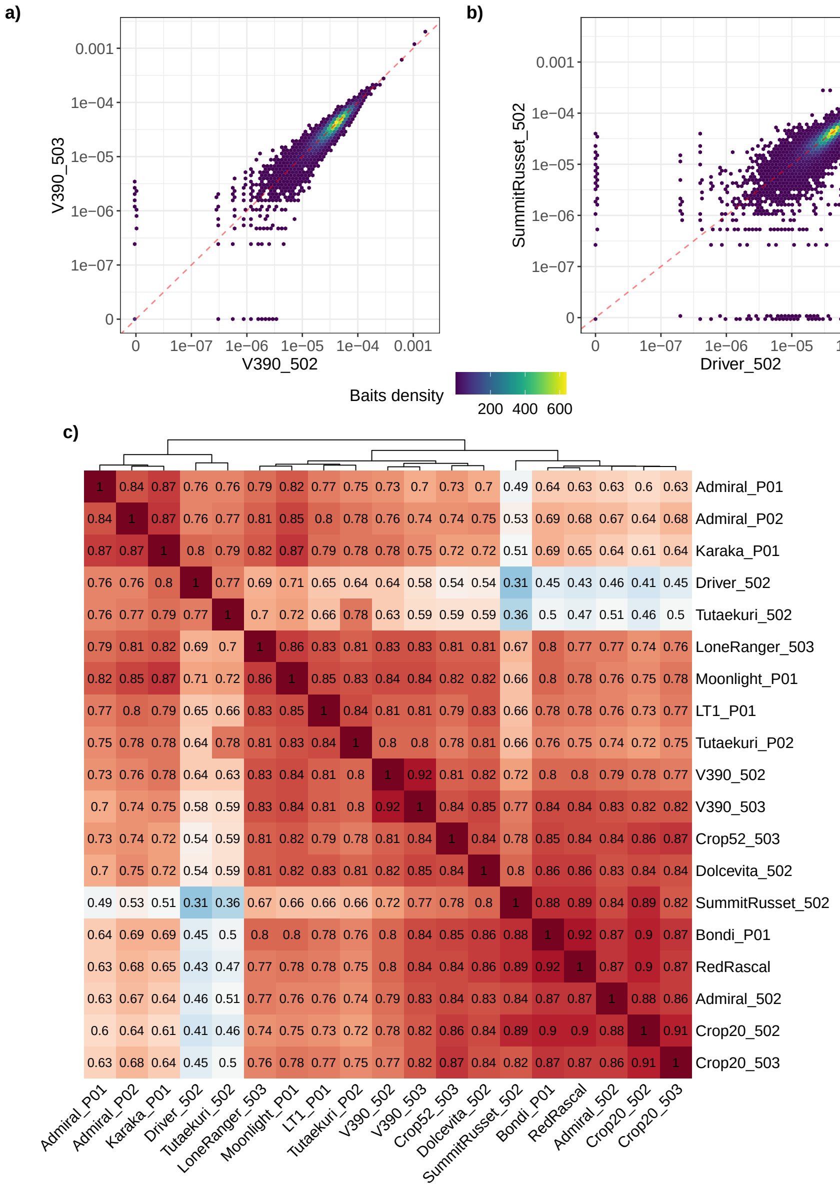
### 4.4.1 Phenotypes

The distribution of the different measured phenotypes is displayed in Supplementary Figure F.1. The yield, market and dry matter percentage traits follow a normal distribution (Shapiro Wilk's test p-value non significant). I show in Figure 4.2 the correlation between the different measured phenotypes. As expected, the pairs of phenotypes measuring a same trait (i.e. `Yield_tha` / `Relative_Yield`, `Market_tha` / `RelativeMarket`, and `bruising_mean` / `bruising_frac`) are highly correlated. The yield and market phenotypes share a strong correlation (0.93 to 0.98), and are also more modestly correlated with the the saleable, general impression, vigour and maturity phenotypes (correlation coefficients between 0.2 and 0.5). The correlation between the saleable trait and the general impression score is high (0.79), which indicates that the general impression score captures relatively well the marketable output of the crop. Maturity and vigour are correlated at 0.55. The bruising, dry matter content, sprouting and sugar phenotypes, which are post-harvest traits as opposed to previously discussed phenotypes, are not correlated to any of the measured traits. It is surprising in the case of the dry matter content, as several previous studies have observed a strong correlation between starch content and bruising phenotypes (Hara-Skrzypiec et al., 2018). However this can be explained by the fact that the parental lines used for the crops have been selected for tuber starch content, and consequently the major factors of tuber bruising related to starch content have been removed. Therefore the observed variation in tuber bruising arises from factors independent from the tuber starch content.



| | RelativeYield | Yield_tha | RelativeMArket | Market_tha | General_Impression | Perc_saleable | bruising_frac | bruising_mean | spr | Dmperc | sugar | Maturity | Vigour |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RelativeYield | 1 | 0.96 | 0.97 | 0.94 | 0.37 | 0.34 | −0.1 | −0.1 | 0.01 | 0.06 | 0.09 | 0.27 | 0.39 |
| Yield_tha | 0.96 | 1 | 0.93 | 0.98 | 0.37 | 0.35 | −0.08 | −0.07 | 0.02 | 0.04 | 0.11 | 0.22 | 0.37 |
| RelativeMArket | 0.97 | 0.93 | 1 | 0.95 | 0.5 | 0.49 | −0.13 | −0.13 | 0 | 0.04 | 0.09 | 0.22 | 0.36 |
| Market_tha | 0.94 | 0.98 | 0.95 | 1 | 0.5 | 0.51 | −0.09 | −0.07 | 0 | 0.03 | 0.12 | 0.18 | 0.34 |
| General_Impression | 0.37 | 0.37 | 0.5 | 0.5 | 1 | 0.79 | 0 | 0.01 | −0.11 | −0.08 | 0.02 | −0.09 | 0.08 |
| Perc_saleable | 0.34 | 0.35 | 0.49 | 0.51 | 0.79 | 1 | 0.01 | 0 | −0.1 | −0.06 | 0.07 | −0.1 | 0.07 |
| bruising_frac | −0.1 | −0.08 | −0.13 | −0.09 | 0 | 0.01 | 1 | 0.92 | 0.1 | 0.05 | −0.09 | 0.11 | 0.02 |
| bruising_mean | −0.1 | −0.07 | −0.13 | −0.07 | 0.01 | 0 | 0.92 | 1 | 0.1 | 0.08 | −0.11 | 0.14 | 0.04 |
| spr | 0.01 | 0.02 | 0 | 0 | −0.11 | −0.1 | 0.1 | 0.1 | 1 | 0.05 | 0 | −0.02 | −0.01 |
| Dmperc | 0.06 | 0.04 | 0.04 | 0.03 | −0.08 | −0.06 | 0.05 | 0.08 | 0.05 | 1 | −0.12 | 0.1 | 0.14 |
| sugar | 0.09 | 0.11 | 0.09 | 0.12 | 0.02 | 0.07 | −0.09 | −0.11 | 0 | −0.12 | 1 | −0.15 | −0.12 |
| Maturity | 0.27 | 0.22 | 0.22 | 0.18 | −0.09 | −0.1 | 0.11 | 0.14 | −0.02 | 0.1 | −0.15 | 1 | 0.55 |
| Vigour | 0.39 | 0.37 | 0.36 | 0.34 | 0.08 | 0.07 | 0.02 | 0.04 | −0.01 | 0.14 | −0.12 | 0.55 | 1 |

Figure 4.2: Correlation between measured phenotypes for 142 samples.



| | Admiral_P01 | Admiral_P02 | Karaka_P01 | Driver_502 | Tutaekuri_502 | LoneRanger_503 | Moonlight_P01 | LT1_P01 | Tutaekuri_P02 | V390_502 | V390_503 | Crop52_503 | Dolcevita_502 | SummitRusset_502 | Bondi_P01 | RedRascal | Admiral_502 | Crop20_502 | Crop20_503 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Admiral_P01 | 1 | 0.84 | 0.87 | 0.76 | 0.76 | 0.79 | 0.82 | 0.77 | 0.75 | 0.73 | 0.7 | 0.73 | 0.7 | 0.49 | 0.64 | 0.63 | 0.63 | 0.6 | 0.63 |
| Admiral_P02 | 0.84 | 1 | 0.87 | 0.76 | 0.77 | 0.81 | 0.85 | 0.8 | 0.78 | 0.76 | 0.74 | 0.74 | 0.75 | 0.53 | 0.69 | 0.68 | 0.67 | 0.64 | 0.68 |
| Karaka_P01 | 0.87 | 0.87 | 1 | 0.8 | 0.79 | 0.82 | 0.87 | 0.79 | 0.78 | 0.78 | 0.75 | 0.72 | 0.72 | 0.51 | 0.69 | 0.65 | 0.64 | 0.61 | 0.64 |
| Driver_502 | 0.76 | 0.76 | 0.8 | 1 | 0.77 | 0.69 | 0.71 | 0.65 | 0.64 | 0.64 | 0.58 | 0.54 | 0.54 | 0.31 | 0.45 | 0.43 | 0.46 | 0.41 | 0.45 |
| Tutaekuri_502 | 0.76 | 0.77 | 0.79 | 0.77 | 1 | 0.7 | 0.72 | 0.66 | 0.78 | 0.63 | 0.59 | 0.59 | 0.59 | 0.36 | 0.5 | 0.47 | 0.51 | 0.46 | 0.5 |
| LoneRanger_503 | 0.79 | 0.81 | 0.82 | 0.69 | 0.7 | 1 | 0.86 | 0.83 | 0.81 | 0.83 | 0.83 | 0.81 | 0.81 | 0.67 | 0.8 | 0.77 | 0.77 | 0.74 | 0.76 |
| Moonlight_P01 | 0.82 | 0.85 | 0.87 | 0.71 | 0.72 | 0.86 | 1 | 0.85 | 0.83 | 0.84 | 0.84 | 0.82 | 0.82 | 0.66 | 0.8 | 0.78 | 0.76 | 0.75 | 0.78 |
| LT1_P01 | 0.77 | 0.8 | 0.79 | 0.65 | 0.66 | 0.83 | 0.85 | 1 | 0.84 | 0.81 | 0.81 | 0.79 | 0.83 | 0.66 | 0.78 | 0.78 | 0.76 | 0.73 | 0.77 |
| Tutaekuri_P02 | 0.75 | 0.78 | 0.78 | 0.64 | 0.78 | 0.81 | 0.83 | 0.84 | 1 | 0.8 | 0.8 | 0.78 | 0.81 | 0.66 | 0.76 | 0.75 | 0.74 | 0.72 | 0.75 |
| V390_502 | 0.73 | 0.76 | 0.78 | 0.64 | 0.63 | 0.83 | 0.84 | 0.81 | 0.8 | 1 | 0.92 | 0.81 | 0.82 | 0.72 | 0.8 | 0.8 | 0.79 | 0.78 | 0.77 |
| V390_503 | 0.7 | 0.74 | 0.75 | 0.58 | 0.59 | 0.83 | 0.84 | 0.81 | 0.8 | 0.92 | 1 | 0.84 | 0.85 | 0.77 | 0.84 | 0.84 | 0.83 | 0.82 | 0.82 |
| Crop52_503 | 0.73 | 0.74 | 0.72 | 0.54 | 0.59 | 0.81 | 0.82 | 0.79 | 0.78 | 0.81 | 0.84 | 1 | 0.84 | 0.78 | 0.85 | 0.84 | 0.84 | 0.86 | 0.87 |
| Dolcevita_502 | 0.7 | 0.75 | 0.72 | 0.54 | 0.59 | 0.81 | 0.82 | 0.83 | 0.81 | 0.82 | 0.85 | 0.84 | 1 | 0.8 | 0.86 | 0.86 | 0.83 | 0.84 | 0.84 |
| SummitRusset_502 | 0.49 | 0.53 | 0.51 | 0.31 | 0.36 | 0.67 | 0.66 | 0.66 | 0.66 | 0.72 | 0.77 | 0.78 | 0.8 | 1 | 0.88 | 0.89 | 0.84 | 0.89 | 0.82 |
| Bondi_P01 | 0.64 | 0.69 | 0.69 | 0.45 | 0.5 | 0.8 | 0.8 | 0.78 | 0.76 | 0.8 | 0.84 | 0.85 | 0.86 | 0.88 | 1 | 0.92 | 0.87 | 0.9 | 0.87 |
| RedRascal | 0.63 | 0.68 | 0.65 | 0.43 | 0.47 | 0.77 | 0.78 | 0.78 | 0.75 | 0.8 | 0.84 | 0.84 | 0.86 | 0.89 | 0.92 | 1 | 0.87 | 0.9 | 0.87 |
| Admiral_502 | 0.63 | 0.67 | 0.64 | 0.46 | 0.51 | 0.77 | 0.76 | 0.76 | 0.74 | 0.79 | 0.83 | 0.84 | 0.83 | 0.84 | 0.87 | 0.87 | 1 | 0.88 | 0.86 |
| Crop20_502 | 0.6 | 0.64 | 0.61 | 0.41 | 0.46 | 0.74 | 0.75 | 0.73 | 0.72 | 0.78 | 0.82 | 0.86 | 0.84 | 0.89 | 0.9 | 0.9 | 0.88 | 1 | 0.91 |
| Crop20_503 | 0.63 | 0.68 | 0.64 | 0.45 | 0.5 | 0.76 | 0.78 | 0.77 | 0.75 | 0.77 | 0.82 | 0.87 | 0.84 | 0.82 | 0.87 | 0.87 | 0.86 | 0.91 | 1 |

Figure 4.3: Comparison of baits normalised read counts between parent samples. a) Comparison between two technical replicates of the V390 variety; the normalised read counts of the baits in replicate 503 are plotted against the corresponding counts in replicate 502. b) Comparison between two different varieties; the normalised read counts of the baits in the SummitRusset sample are plotted against the corresponding counts in the Driver sample. c) $R^2$ values of the linear regression of the baits normalised read counts between each pair of parent samples. Values above 0.5 are depicted in shades of red, while values below 0.5 are depicted in shades of blue. The samples have been clustered based on euclidean distance.

### 4.4.2 Baits quality

The baits coverage and depth of coverage are discussed for the parents samples. As expected, the coverage depth is highest for bases in the middle of the baits and decreases as we move away from the baits. For all parent samples, most bases within the baits have a coverage depth ranging from 10 to around 200, and up to 800 for some parent samples. For all parents samples, less than one percent of all bases within baits across the genome have no coverage, and more than 95% of the baits are completely covered. The average number of reads per bait across the parent samples (normalised for each sample by the total read count) is modestly correlated with the GC content of the baits (adjusted $R^2$ between the log10 of average read count per bait and GC content of 0.097, p-value < $2.10^{-16}$). For baits with a GC content below 0.3, baits with low GC content have on average less reads than baits with higher GC content. For baits with GC contents higher than 0.3, the GC content doesn't affect the average read counts much. As can be seen in Figure 4.3, the read counts per bait (normalised by the total read count per sample) is consistent between technical replicates (Figure 4.3 a)), while there is more variability between samples from different varieties (Figure 4.3 b)). This variability can arise from technical (e.g. due to batch effects) or indels or copy number variation in the regions targeted by the baits in one of the varieties, or the presence of off-targets. In particular, some baits are covered in one parent (i.e. reads detected for the bait) but not the others (null read count), possibly indicating some deletion or mutations in the targeted region. The $R^2$ of a linear regression between the baits read counts of each pair of samples in depicted in Figure 4.3. Higher $R^2$ are observed for pairs of parents that are related (e.g. Crop20 and Bondi or RedRascal), while lower values are observed for pairs of parents that are not related (e.g. Tutaekuri vs Crop20 and related parents).

### 4.4.3 Genomics data

A total of 1,388,205 variants were called, out of which 1,280,324 were SNPs (92.2%). Amongst them, 940,103 biallelic SNPs (67.7% of all variants), for which both alleles were observed in the samples of interest, were retained for further analysis. Furthermore, variants with 10% or more missing data, or with a QD score (quality normalised by coverage) of less than two were discarded, yielding a total of 602,955 variants (64.1% of biallelic SNPs) that passed the filtering step. This number is not unexpected given the high heterozygosity of tetraploid potatoes and the population design, in which parents with varied genotypes were crossed. The QD score is an interesting metric to perform variant filtering, as it allows to filter out variants with good quality score and depth of coverage but for which there is little evidence for the alternate allele. This is illustrated in Figure 4.4, in which we can see variants with reasonable quality and coverage that are assigned a null QD score. In addition, out of the 184 genotyped samples, two of them didn't meet the criteria of less than 10% of missing data and were consequently discarded, leaving 182 samples with genotype information. Interestingly, 43.2% of the variants (before filtering) were detected as problematic in at least one half-sibling family, i.e. one of the two alleles was observed in one or more progeny samples from the family, but not in the

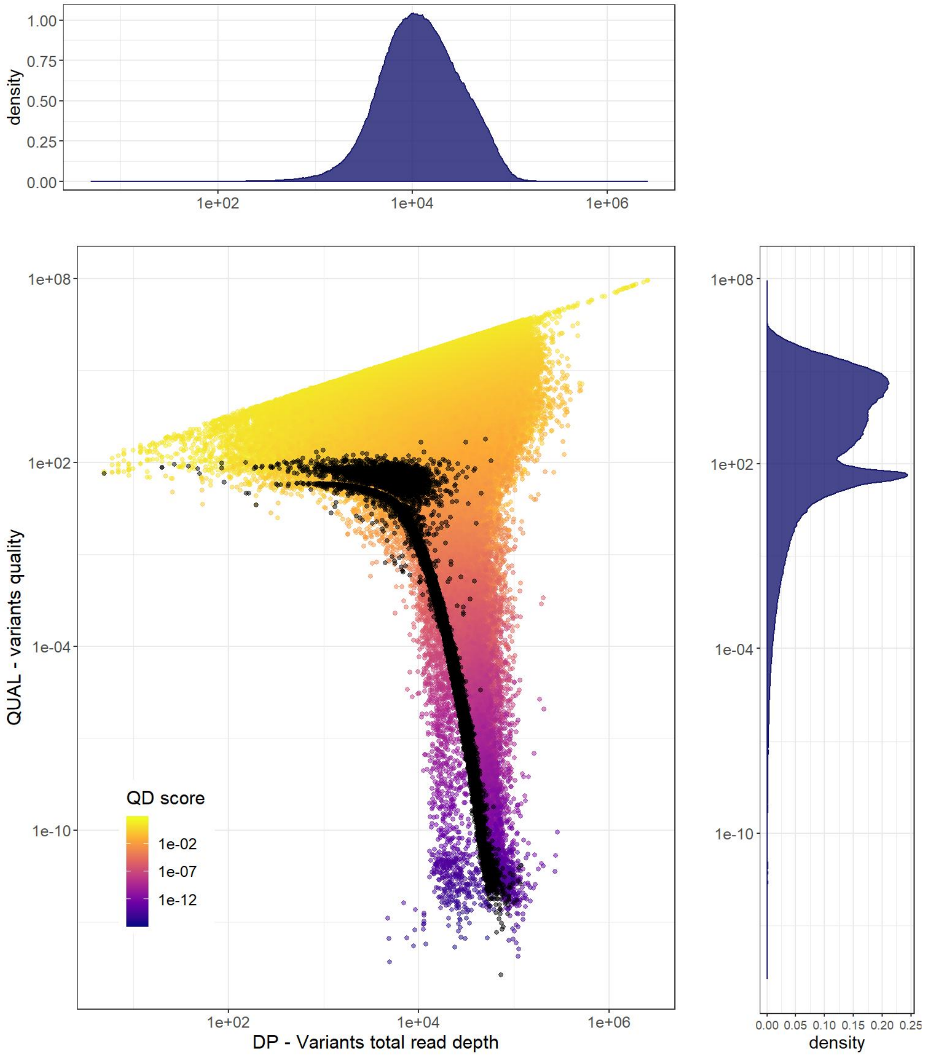


Figure 4.4: Variants depth of coverage (x axis), quality (y axis) and QD score (colour). Variants with a null QD score are represented in black. Note that variants with a quality score of zero were excluded from the plot (for the log10 transformation of quality scores) and are assigned a QD score of zero.

corresponding parents. This can be due to a sequencing error from the parents samples, i.e. the allele is incorrectly not detected in the parent samples, to a sequencing error from the progeny samples in which the problematic allele was detected or to read mapping errors. In consequence, they were not discarded from the analysis. Of these problematic variants, 47% (244,926 SNPs) were removed by filtering. This is similar to the fraction of the total number of variants removed by the filtering, which indicates that the filtering did not target specifically these problematic variants. This is not surprising, given that a good-quality variant might be flagged as problematic due to an incorrect dosage calling for one of the parent. Therefore, such variant should not be removed by the filtering. Lastly, it is to be noted that I did not filter out SNPs based on their minor allele frequency, because of the population structure of the samples. As I am working with half-sibling families, some alleles could only be present in one of the families, and therefore be considered as a rare allele, whilst carrying useful information about the difference between the families. Similarly, I did not filter variants for Hardy Weinberg equilibrium, as the individuals studied do not come from a natural population but from a complex multi-family design in which different parents were crossed. Moreover these individuals underwent selection for several traits of interests. Therefore I do not expect variants to follow Hardy Weinberg equilibrium. In addition, the goal of such filtering for association studies is to correct for bias arising from co-selection of alleles, which is done in this case by correcting for population structure instead (see Section 4.3.5).

As expected, the variants density along the chromosomes correlates with baits density, and decreases near the centromeres. This is illustrated in Supplementary Figure F.2, in which a higher density of transposons reflects the location of the centromeres. This is because the baits were designed to capture exons of annotated genes, therefore they preferentially target regions with high exons density, and are thus less abundant near the centromeres where there are less protein-coding genes. The average sample exonic heterozygosity, i.e the average percentage of heterozygous variants per sample, is 32.5%. This means that on average, a sample carries only one of two possible alleles for $\sim$ 67% of the variants; note that exons are less variables compared to introns or UTRs. This can be explained by the fact that for some variants one of the two alleles is unique to a certain parent (that is, not present in other parents) and is thus only carried by the progeny of crosses involving this parent. This is illustrated in Figure 4.5, where we can see the variants that are polymorphic (i.e. for which both alleles are present) in specific subsets of the half-sibling families' progeny. In the plot, we can see that only 118,121 (19.6%) variants are polymorphic in all half-sibling families. Other variants are only polymorphic in one family, or in two families that share a common parent (e.g. families 2134 and 2172 that share SummitRusset as mother). In total, 594,122 (98.5%) variants have both alleles present in at least one progeny sample, indicating that for the remaining 1.5% of variants one of the alleles is not passed on from the parents to the progeny. The average variants heterozygosity (i.e. average percentage of heterozygous samples per variant) across the genome is 32.41%.

The results of a PCA performed on the variant dosage information for the 182 samples are shown in Figures 4.6 and 4.7. These plots illustrate the population structure of the samples, and allows us to detect potential mislabelled samples. The first component of the PCA, which accounts for 8.2% of the variance in the data, clearly separates the Tutaekuri samples from the other lines (see Figure 4.6). This makes sense as Tutaekuri (also called Urineka) is a Taewa, i.e. a variety of indigenous purple potato, genetically quite different from the other varieties. A possible mislabelled sample can be identified in this plot: a progeny sample from a cross between Crop52 and Crop9 (yellow point at the center of the plot) is positioned close to the Tutaekuri progeny and far away from the other

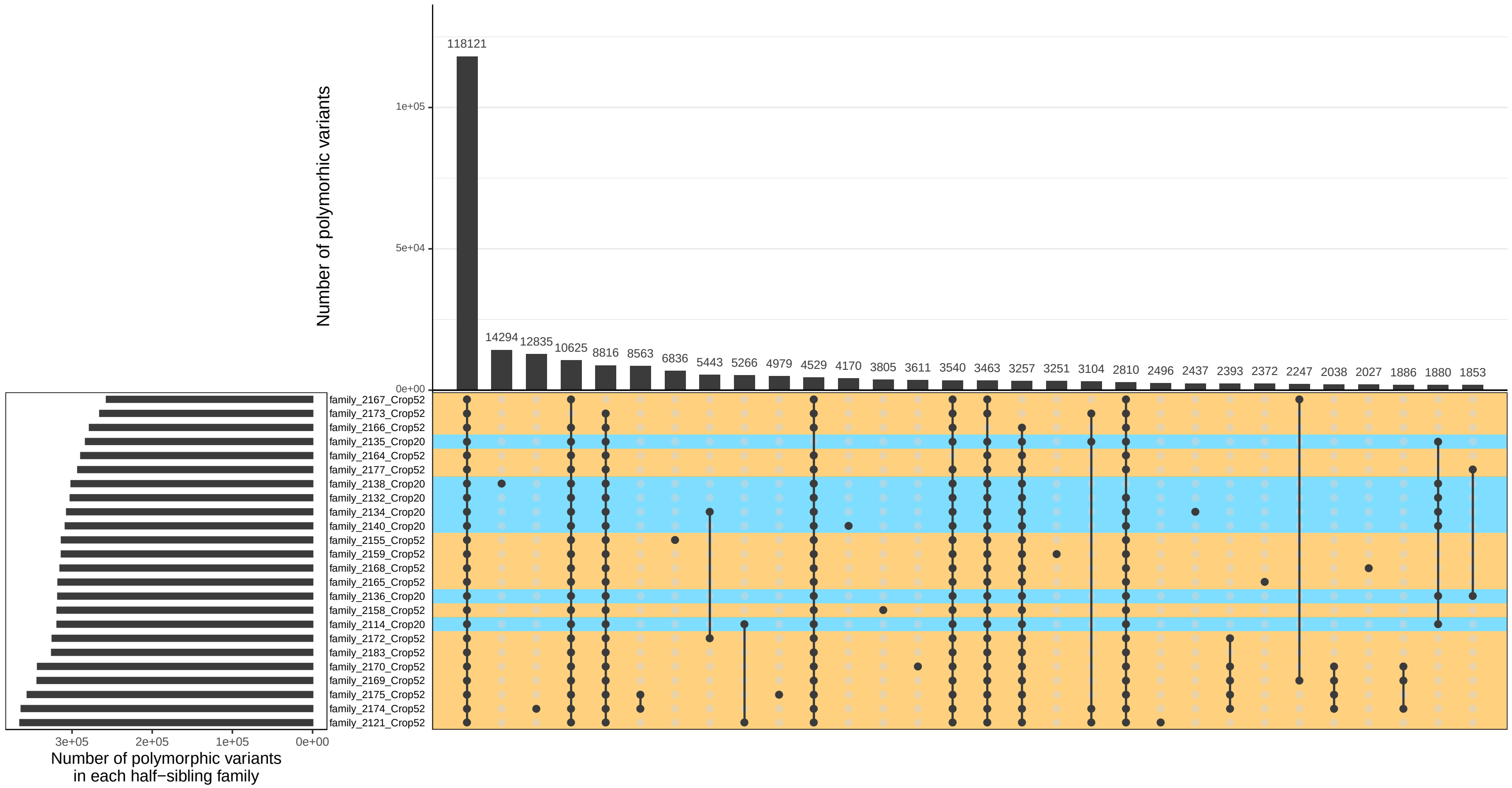


Figure 4.5: Upset plot representing the number of variants that are polymorphic (i.e. for which both alleles are observed) only in specfic subsets of the progeny samples. Each half-sibling family is represented as a row, with the colour indicating whether it involves a cross with Crop52 (gold) or Crop20 (blue). For each column in the graph, the number of variants that are polymorphic only in the progeny of families marked with a dot and not in the progeny of families with no dot is represented by a vertical bar. Only the first 30 intersection sets with the highest number of variants are displayed. We can see that most variants are polymorphic in all families (first column of the plot). We can for example notice that 1,880 variants have both alleles present only in the progeny samples of Crop20 crosses (penultimate column).

progenies of this cross. The second and third principal components (Figure 4.7) separate Crop20 and Crop52 progeny, positioned in the upper right triangle and lower left triangle of the plot, respectfully. All replicates of a same parent are located very closely, which is to be expected as they are replicates of the same variety, with the exception of the three Admiral replicates that are apart in the principal components space. This is probably due to a mislabelling of the samples. Lastly, note that the progeny samples of a cross between two genotyped parents are positioned half-way between the two parents in the PCA plot, which is not surprising as they share half of the genetic material of each parent.

### 4.4.4 Transcriptomics data

RNA measurements were obtained for 100 samples, all progeny of Crop52. The library size of the samples ranged from $6.8 \times 10^6$ to $4.2 \times 10^7$. For most samples, more than 80% of the reads were uniquely mapped to the reference genome, with an average of 76.9% of mapped reads that mapped to a gene in the reference genome. In total, read counts were obtained for 39,028 transcribed genes. Raw read counts were converted to RPKM counts to correct for gene length and library size, and to VST counts using the Variance Stabilizing Transformation (Anders & Huber, 2010), which corrects the mean-variance bias in RNAseq data, for comparison. Transcribed genes were filtered out based on the total raw read counts and RPKM counts across all samples, as well as according to the number of samples with low read count. This led to retaining 25,163 transcribed genes (64.47%). GO annotations were obtained for 17,387 (69.1%) retained transcribed genes.

### 4.4.5 Population structure and individual relatedness

The program STRUCTURE was used to uncover the population structure among the 182 genotyped samples. Five subpopulations were identified as the optimal setting; and five runs of STRUCTURE with K (the number of subpopulations) set to five led to very similar results. The main output of STRUCTURE is the posterior membership probability profile of each sample, i.e. the probability that the sample belongs to each of the K detected subpopulations. This informs about the possible admixture of the samples, in the case where the membership probability of a sample is not zero for more than one subpopulation. The subpopulations identified, that I thereafter denote as K$i, 1 \leq i \leq 5$, are consistent with the known pedigree of the samples. Figure 4.8 displays the estimated posterior membership probabilities of the parent samples for the five subpopulations. Some of the parents are assigned exclusively to one subpopulation: Tutaekuri to K1, V390 to K2, Crop20 to K3, Crop52 to K4 and Dolcevita to K5. This makes sense as these are the most different varieties in the group, except for Crop52 which is a progeny of Crop20. The remaining parents are detected as an admixture of these subpopulations; for example, Moonlight, Karaka and LoneRanger are assigned a posterior membership probability of around 0.5 for subpopulation K2, which is in agreement with the fact that they are the progeny of the V394 crop, which has the same breeding as V390. Note that one of the Admiral replicates has a different membership probability profile than the other two replicates, which provides further evidence of sample mislabelling. Unsurprisingly, the membership probability profile of each progeny sample is an equal mix of these of the corresponding parents (see Supplementary Figure F.3 for two examples): for example the membership probabilities for the progeny of Crop52 and Dolcevita are around 0.5 for K4 (representing the Crop52 genotype) and 0.5 for K5 (representing the Dolcevita genotype). This also permits the detection of outliers progeny samples, that can arise from mislabelling or self of one of the parents, or the contamination by a stray pollen.

The population structure was also analysed using the DAPC algorithm. The analysis revealed four sample clusters as the optimal setting, where the clusters inform us about different subpopulations within the samples. One of the outputs of DAPC is a measure similar to the posterior membership probability computed with STRUCTURE; however it mainly reflects the attribution of the samples to the different clusters and is less informative about possible admixture. I show the membership probabilities of the parent samples in Figure 4.8; the membership probabilities of the parent samples for different numbers of clusters are shown in Supplementary Figure F.4. Similarly to STRUCTURE, DAPC separates Crop20, V390, Tutaekuri and Crop52 into different clusters, but groups the latter with Dolcevita. More generally, each DAPC cluster, that I will denote as C$i, 1 \leq i \leq 4$, corresponds to one of the STRUCTURE subpopulations (or two in the case of cluster C4), and gathers parents whose membership probability for the corresponding STRUCTURE subpopulation is the highest; e.g. samples that have a high posterior membership probability for STRUCTURE subpopulation K2 (blue) were all assigned to cluster C2 (yellow). Note that RedRascal is the only parent sample that has a non-null posterior membership probability for two clusters. While it was originally assigned to cluster C4, it also has a small but non-null posterior membership probability for cluster C1, which is in accordance with the fact that cluster C1 gathers samples with a high STRUCTURE posterior membership probability for subpopulation K3, and RedRascal is in this case. For comparison, the DAPC analysis was also performed with five clusters to see whether the clusters would match the STRUCTURE subpopulations. I found that instead Dolcevita and Crop52 were still grouped together, while V390 and parents descending from V394 were split into two clusters. With DAPC, the progeny samples of each cross are all grouped with one of the parents. Similarly to the results of STRUCTURE, some progeny samples are not clustered with their siblings, which could be an indication that the sample was mislabelled. These samples also have a different STRUCTURE membership probability profile than their siblings. Interestingly, they are also the samples with the highest proportion of problematic variants, i.e. variants for which they carry an allele that is not found in the parents. Ultimately, there is a clear correlation between the samples coordinates in the DAPC space and the membership probability for the STRUCTURE subpopulations corresponding to the clusters that each discriminant function (i.e. axis in the DAPC space) separates. Indeed, the discriminant functions are designed to maximise the variance between clusters, and thus the samples coordinates on each of the DAPC space axis informs about their membership to one or more DAPC clusters. The latter are similar to the STRUCTURE subpopulations and thus samples coordinates also provide information

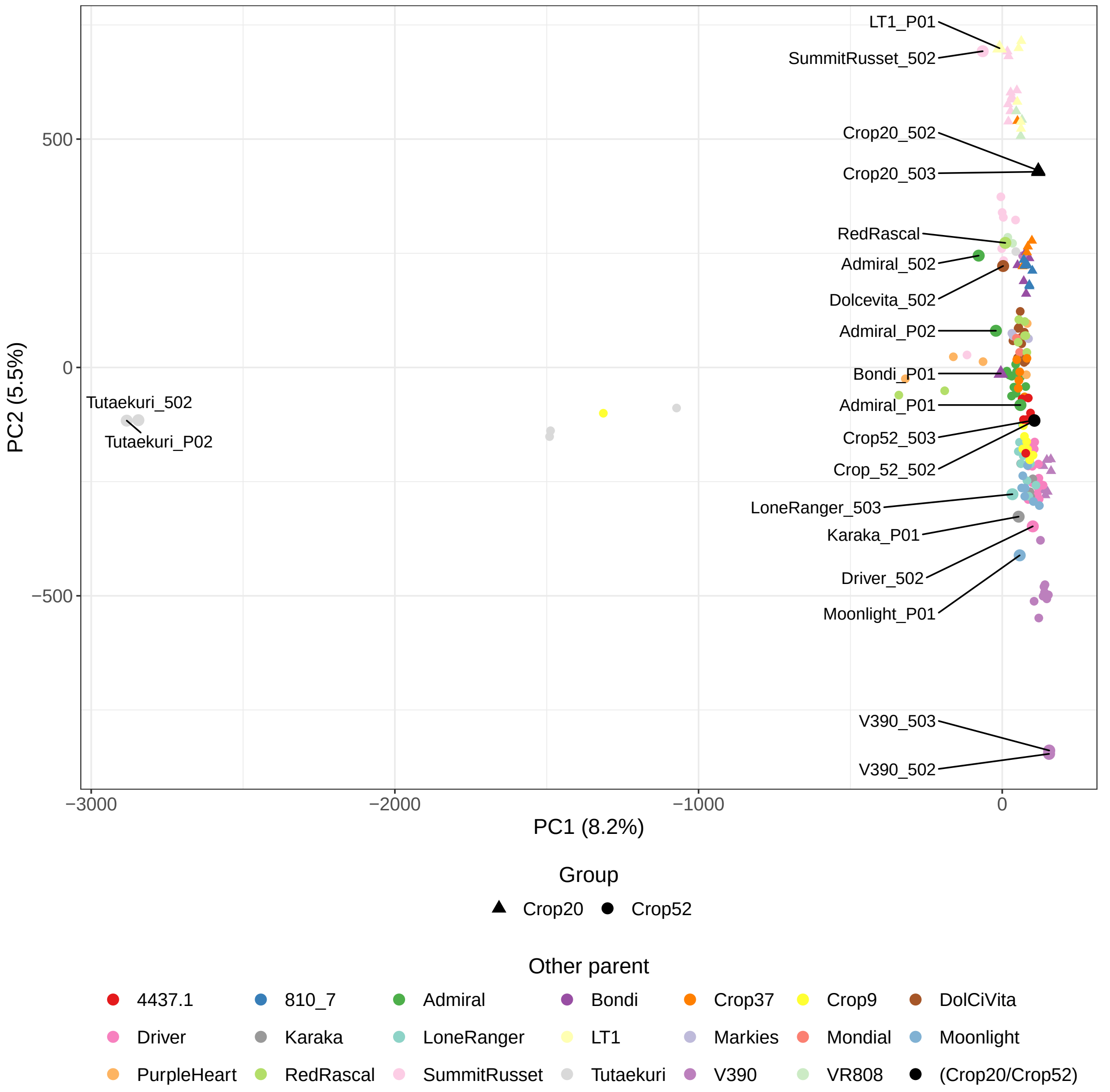


Figure 4.6: PCA plot of the first two components of a PCA applied to the variant dosage of 182 samples. The name of the parent samples (large points) is indicated next to the corresponding point, while progeny samples are indicated with smaller points. For the progeny samples, the shape of the points represent the first parent (i.e. Crop20 or Crop52), and the colour the other parent. For the parent sample, the suffix indicates the batch in which the sample was processed. Samples with suffixes 502 and 503 arise from the same biological sample, and samples with suffixes p01 and p02 arise from a second biological sample.

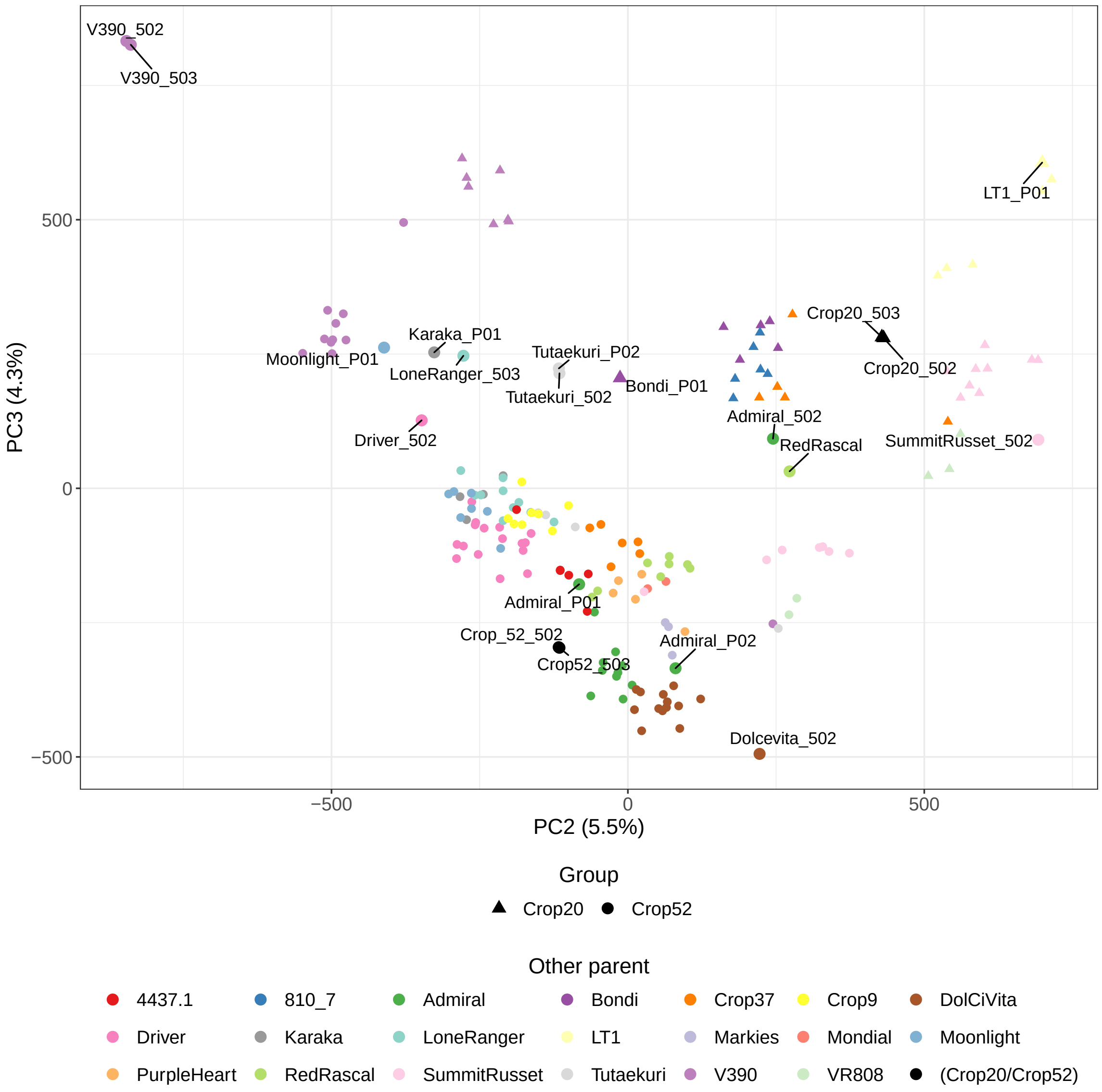


Figure 4.7: PCA plot of the second and third components of a PCA applied to the variant dosage of 182 samples. The name of the parent samples (large points) is indicated next to the corresponding point, while progeny samples are indicated with smaller points. For the progeny samples, the shape of the points represent the first parent (i.e. Crop20 or Crop52), and the colour the other parent. For the parent sample, the suffix indicates the batch in which the sample was processed. Samples with suffixes 502 and 503 arise from the same biological sample, and samples with suffixes p01 and p02 arise from a second biological sample.

about the samples membership probability for the concerned STRUCTURE subpopulations.

While the population structure was studied over all 182 genotyped samples, the kinship matrix, which represents the relationship among samples, was computed over the 142 progeny samples for which I also have phenotype data. This is because in the case of population structure, including the parent samples can help decipher the structure amongst samples. On the contrary for the kinship matrix the relationship between any pair of samples is independent from other samples, and only samples with phenotype data will be considered in the GWAS analysis. The kinship matrix (Figure 4.9) clearly illustrates the half-sibling design of the experiment. Progeny from a same cross (full siblings, along the diagonal) are highly related. They are also related to a lesser extent to progeny from a cross sharing a common parent. On the contrary progeny from different crosses with no common parents show little relatedness. When clustering samples based on the kinship matrix, progeny from a same cross are clustered together, and two super-clusters separate Crop20 progeny from Crop52 progeny, with the exception of the V390 progeny (with both Crop20 and Crop52) that was grouped in a same cluster.

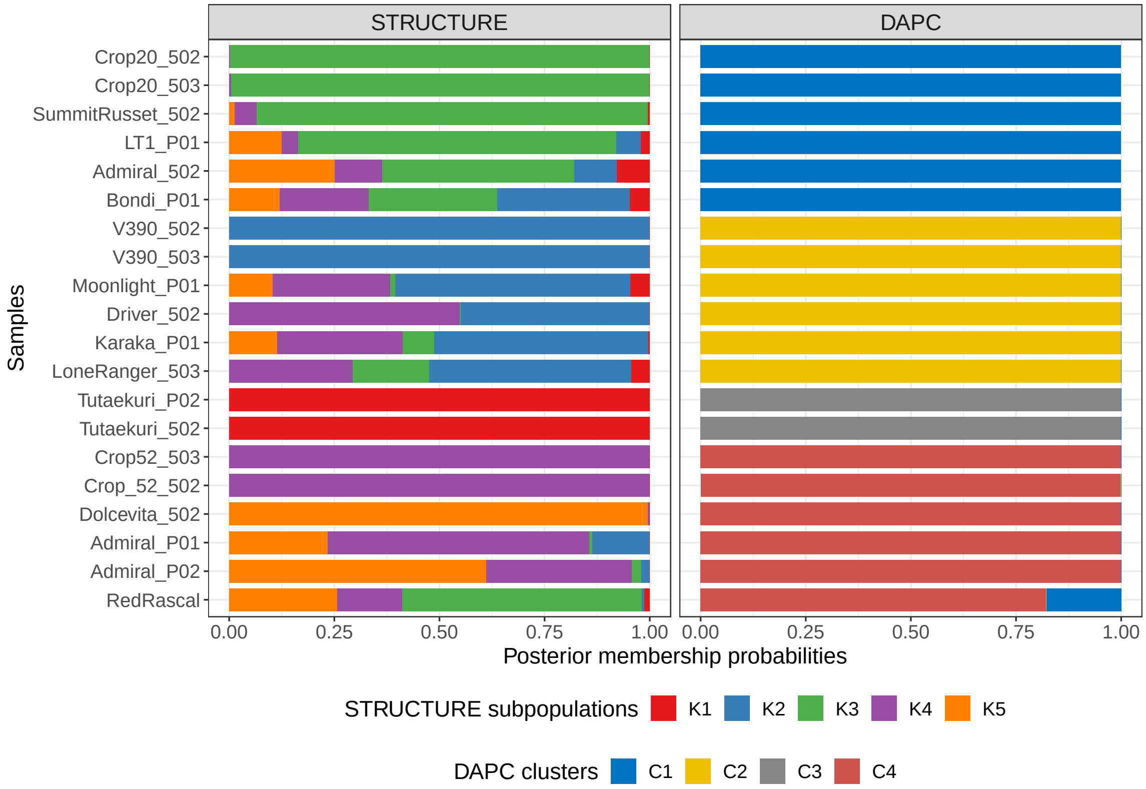


Figure 4.8: Population structure uncovered with STRUCTURE and DAPC for parent samples. The posterior membership probabilities of the parent samples are displayed for the five subpopulations identified with STRUCTURE (left panel) and for the four clusters identified with DAPC (right panel).

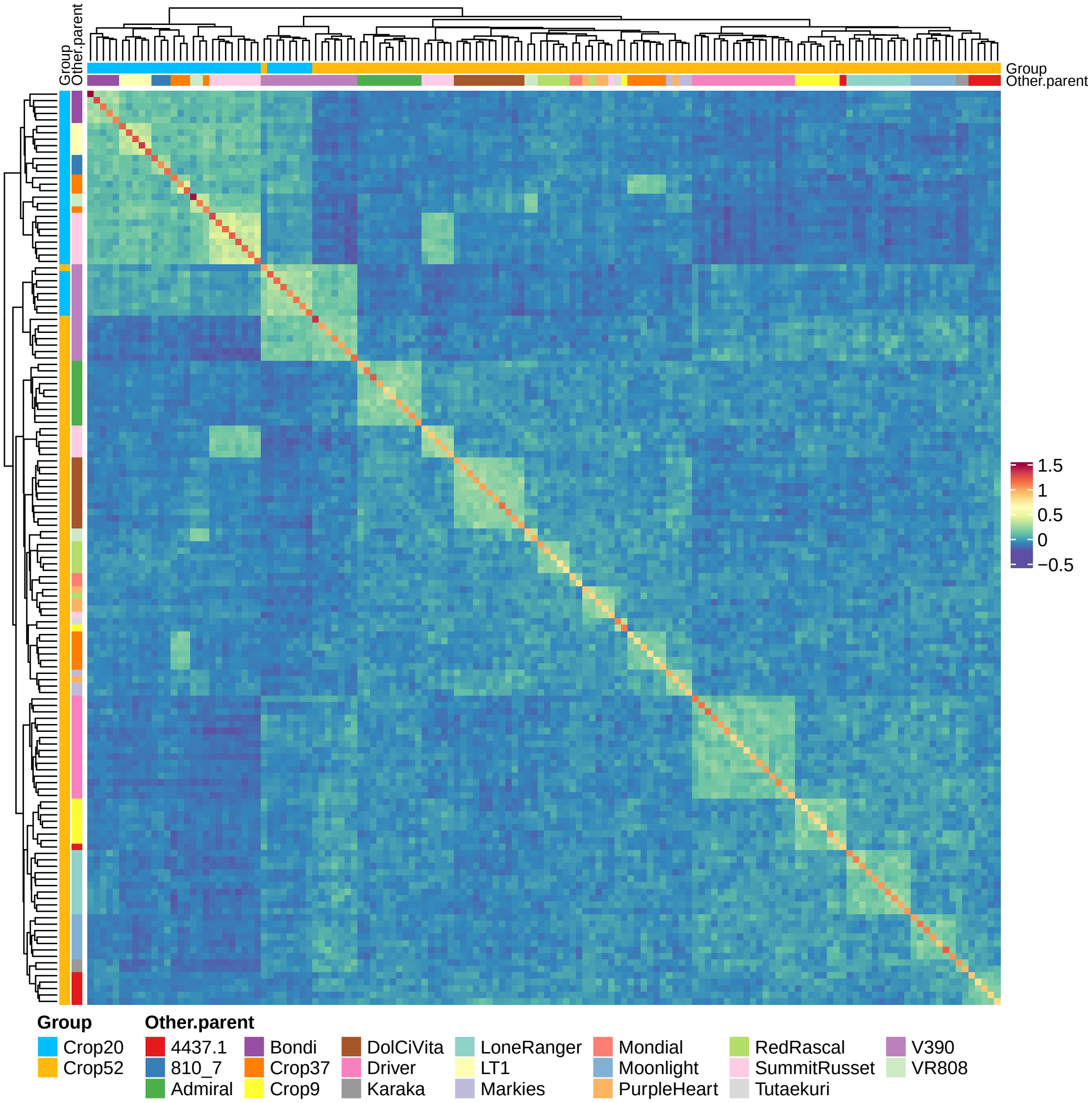


Figure 4.9: Kinship matrix representing the relatedness between 142 progeny samples. Each row and column corresponds to one sample, whose group (i.e. whether it comes from a cross with Crop20 – in blue – or Crop52 – in gold –) and second parent are indicated on the left (for rows) or above (for columns). The colour of each cell indicates the individual relatedness between a pair of samples, with a high value (red) indicating that the two samples are highly related, while a low value (blue) indicates that the samples are not related. Note that the matrix is scaled such that the mean of the diagonal elements is one.

### 4.4.6 Impact of GWAS settings on the marker scores

The GWAS analysis was performed for 142 progeny samples and each of the 13 measured phenotypes. Six different genetic models (describing the effect of a marker's dosage on the phenotype) and six different population settings (correcting for individual relatedness and/or population structure) were considered, which yielded for each phenotype 36 sets of marker scores, each describing the relationship between each marker's dosage and the phenotype according to the corresponding genetic model and population setting. The ability of each GWAS setting (i.e genetic model and population setting) to control for false positive and negative in the resulting scores was evaluated by computing for each an inflation factor (Rosyara et al., 2016). The inflation factor describes the deviation of observed markers scores from their expected values under the null hypothesis of no association with the phenotype. As I anticipate that most markers are not associated with the phenotype, especially in this experiment in which potatoes have undergone selection for some of these traits, I expect that for most markers the observed and expected scores will be very close, which would give an inflation factor close to one. An inflation factor higher than one indicates that many markers are detected as significantly impacting the phenotype, which is not desirable, as it most likely would contain many false positive. Such inflation of false positive may notably be due to a failure of the model to account for relationships between samples (Bazakos et al., 2017). On the contrary, an inflation factor smaller than one indicates that the model fails to detect the relationship between markers and phenotype, which could be caused by over-compensating for population structure (Zhao et al., 2011). The inflation factor is thus an important metric to help us choosing the optimal population setting, which may differ between phenotypes.

Figure 4.10 compares the marker scores obtained with the different genetic models and population settings for the bruising mean score phenotype. We can see that in general the marker scores are more similar when computed with the same genetic model but different population settings than when computed with the same population setting but different genetic models. This is an expected result, as the genetic model defines the relationship between marker dosage and phenotype and thus critically influences the estimated quantitative impact of a marker's dosage on the trait. Note however that the scores obtained with the additive model are positively correlated with those obtained with the duplex dominant models, and more modestly with those obtained with the simplex dominant models. On the contrary the two simplex dominant models return scores that are uncorrelated. For a similar genetic model, the population setting only influences to a small extent the marker scores (high correlation between scores obtained for a same genetic model with different population settings). The naive model yields results that are the most different to other population settings, while settings that account for individual relatedness (i.e. K and K+Q settings) return similar marker scores. Interestingly, it seems that in general the $K_{\text{DAPC}}$ setting yields scores that are more similar to the K setting (correlation of 0.99 for all genetic models) compared to the $K+Q_{\text{STRUCTURE}}$ setting (correlation between 0.94 and 0.96 across the genetic models). This tends to indicate that using the posterior membership probability

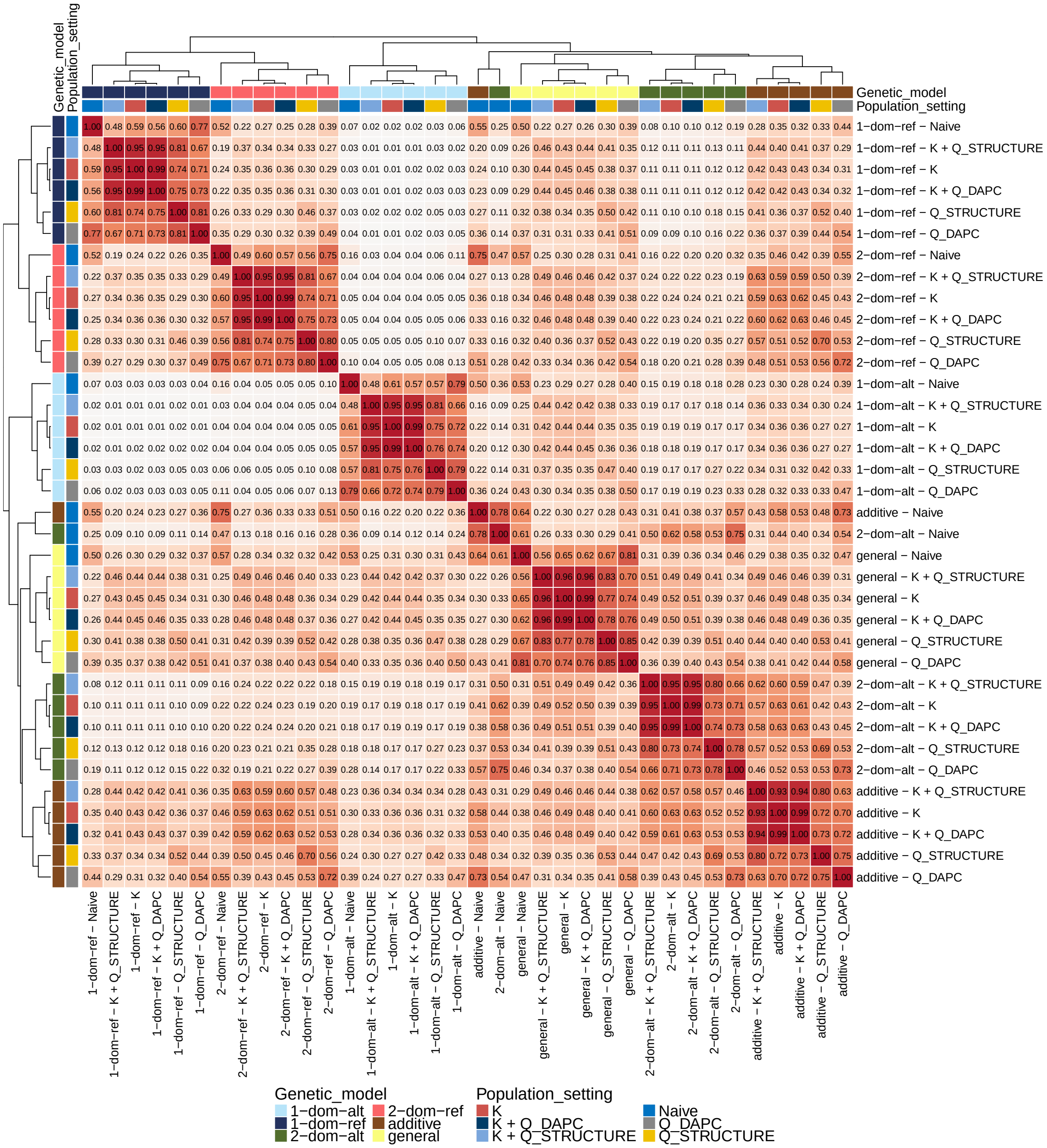


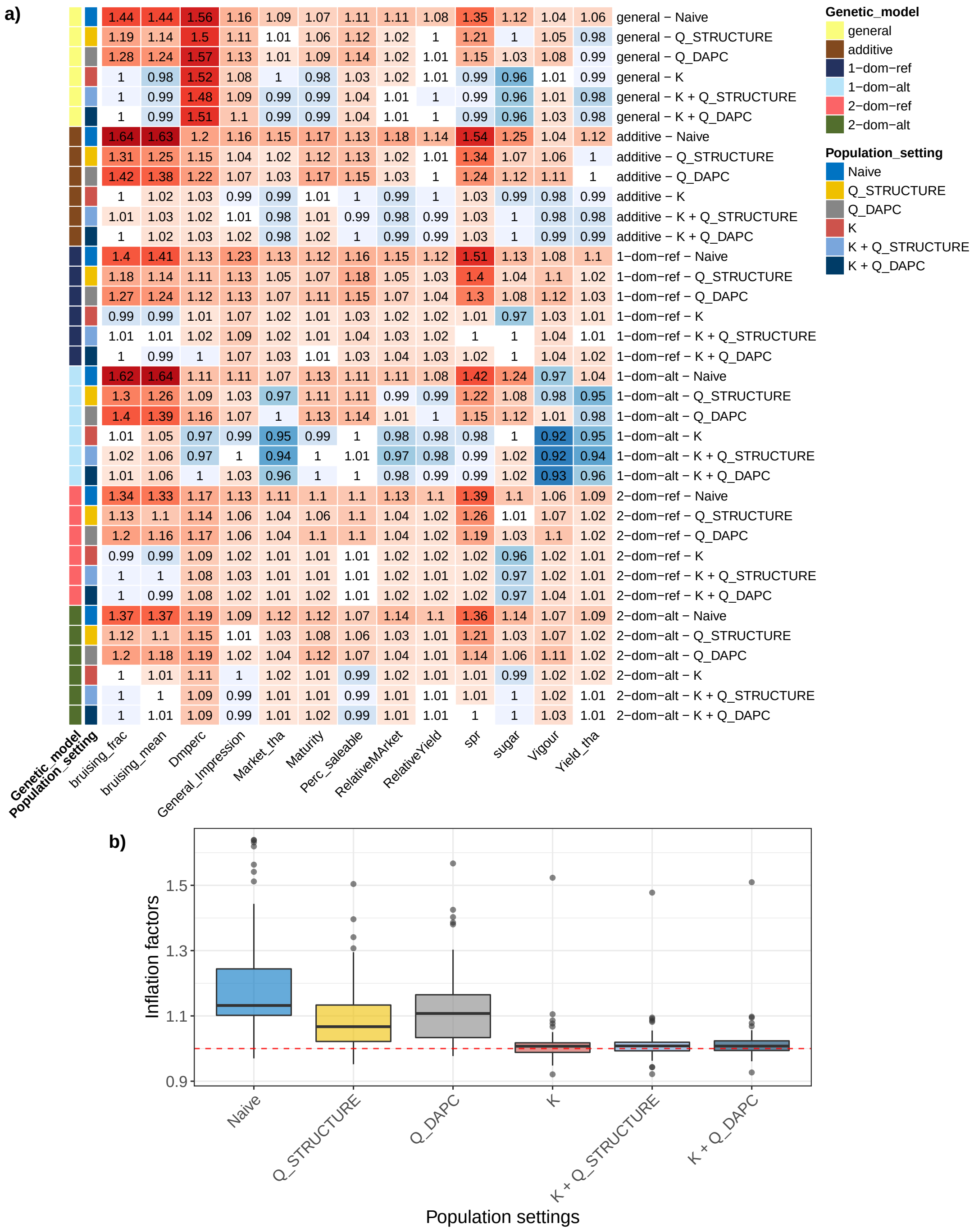
Figure 4.10: Correlation coefficients between the marker scores obtained for the bruising mean score phenotype for different GWAS settings, i.e. different genetic models and population settings.

returned by DAPC is less informative about the structure of the population than the one returned by STRUCTURE. This makes sense as the DAPC output doesn't provide information about the possible admixture of samples, while STRUCTURE does. The same holds when using the samples coordinates in the DAPC space as covariates in the analysis to account for population structure (instead of the posterior membership probability – results not shown). This is in contradiction with the results of Rosyara et al. (2016), who found that the $Q_{\text{DAPC}}$ model outperformed the $Q_{\text{STRUCTURE}}$ model in correcting for population structure. This could be due to the fact that the samples used in this study are more related than the samples used by Rosyara et al. (2016), and thus information about admixture of the samples is more important in this case.

The inflation factors obtained for each phenotype and GWAS setting are presented in Figure 4.11. These results confirm that performing GWAS without accounting for population structure or individual relatedness, i.e. using the naive model, leads to an inflation of high marker scores (inflation factor above one). As observed by Rosyara et al. (2016) and Sharma et al. (2018), correcting only for population structure without accounting for individual relatedness (i.e. Q models) still lead to inflated false positive. Interestingly, the $Q_{\text{STRUCTURE}}$ model, that uses the population structure inferred with STRUCTURE, yields inflation factors closer to one than the model using the population structure inferred with DAPC. Models that account for individual relatedness by including the kinship matrix K yield inflation factors closer to one than the settings that do not include K, indicating that accounting for sample relatedness allows us to better control the false positive inflation. The distribution of inflation factors for the K and K + Q models are very similar, and centred around one, indicating that accounting for the individual relatedness (with K) is sufficient to control markers scores inflation. However, as can be noted in Figure 4.11 a), the optimal population setting (i.e. yielding an inflation factor closest to one) will differ depending on the considered phenotype. For example marker scores obtained for the dry matter content trait are all quite high across all population settings, while those estimated for the yield phenotypes are consistently low. One possible explanation for the latter is that the crops investigated have been selected for this trait and thus the observed variations in phenotypic values are not caused by genetic variations. In addition, yield is a complex trait with many expected QTLs with small effect, and therefore a large population size would be required to be able to detect them.

### 4.4.7 Markers-trait association

To account for multiple testing, the Bonferroni correction was used to set a significance threshold for each GWAS setting and phenotype, with a type-I error ($\alpha$) level of 0.05. As mentioned in the previous section, inflation of marker scores leads to many false positive, so I discarded GWAS settings yielding high inflation factors (above 1.1). GWAS settings yielding low inflation factors (below 0.95) were also discarded, however in the latter case no marker would be considered significant. Indeed, a low inflation factor means that the observed marker scores are lower than their expected values



| Genetic_model – Population_setting | bruising_frac | bruising_mean | Dmperc | General_Impression | Market_tha | Maturity | Perc_saleable | RelativeMArket | RelativeYield | spr | sugar | Vigour | Yield_tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| general − Naive | 1.44 | 1.44 | 1.56 | 1.16 | 1.09 | 1.07 | 1.11 | 1.11 | 1.08 | 1.35 | 1.12 | 1.04 | 1.06 |
| general − Q_STRUCTURE | 1.19 | 1.14 | 1.5 | 1.11 | 1.01 | 1.06 | 1.12 | 1.02 | 1 | 1.21 | 1 | 1.05 | 0.98 |
| general − Q_DAPC | 1.28 | 1.24 | 1.57 | 1.13 | 1.01 | 1.09 | 1.14 | 1.02 | 1.01 | 1.15 | 1.03 | 1.08 | 0.99 |
| general − K | 1 | 0.98 | 1.52 | 1.08 | 1 | 0.98 | 1.03 | 1.02 | 1.01 | 0.99 | 0.96 | 1.01 | 0.99 |
| general − K + Q_STRUCTURE | 1 | 0.99 | 1.48 | 1.09 | 0.99 | 0.99 | 1.04 | 1.01 | 1 | 0.99 | 0.96 | 1.01 | 0.98 |
| general − K + Q_DAPC | 1 | 0.99 | 1.51 | 1.1 | 0.99 | 0.99 | 1.04 | 1.01 | 1 | 0.99 | 0.96 | 1.03 | 0.98 |
| additive − Naive | 1.64 | 1.63 | 1.2 | 1.16 | 1.15 | 1.17 | 1.13 | 1.18 | 1.14 | 1.54 | 1.25 | 1.04 | 1.12 |
| additive − Q_STRUCTURE | 1.31 | 1.25 | 1.15 | 1.04 | 1.02 | 1.12 | 1.13 | 1.02 | 1.01 | 1.34 | 1.07 | 1.06 | 1 |
| additive − Q_DAPC | 1.42 | 1.38 | 1.22 | 1.07 | 1.03 | 1.17 | 1.15 | 1.03 | 1 | 1.24 | 1.12 | 1.11 | 1 |
| additive − K | 1 | 1.02 | 1.03 | 0.99 | 0.99 | 1.01 | 1 | 0.99 | 1 | 1.03 | 0.99 | 0.98 | 0.99 |
| additive − K + Q_STRUCTURE | 1.01 | 1.03 | 1.02 | 1.01 | 0.98 | 1.01 | 0.99 | 0.98 | 0.99 | 1.03 | 1 | 0.98 | 0.98 |
| additive − K + Q_DAPC | 1 | 1.02 | 1.03 | 1.02 | 0.98 | 1.02 | 1 | 0.99 | 0.99 | 1.03 | 1 | 0.99 | 0.99 |
| 1−dom−ref − Naive | 1.4 | 1.41 | 1.13 | 1.23 | 1.13 | 1.12 | 1.16 | 1.15 | 1.12 | 1.51 | 1.13 | 1.08 | 1.1 |
| 1−dom−ref − Q_STRUCTURE | 1.18 | 1.14 | 1.11 | 1.13 | 1.05 | 1.07 | 1.18 | 1.05 | 1.03 | 1.4 | 1.04 | 1.1 | 1.02 |
| 1−dom−ref − Q_DAPC | 1.27 | 1.24 | 1.12 | 1.13 | 1.07 | 1.11 | 1.15 | 1.07 | 1.04 | 1.3 | 1.08 | 1.12 | 1.03 |
| 1−dom−ref − K | 0.99 | 0.99 | 1.01 | 1.07 | 1.02 | 1.01 | 1.03 | 1.02 | 1.02 | 1.01 | 0.97 | 1.03 | 1.01 |
| 1−dom−ref − K + Q_STRUCTURE | 1.01 | 1.01 | 1.02 | 1.09 | 1.02 | 1.01 | 1.04 | 1.03 | 1.02 | 1 | 1 | 1.04 | 1.01 |
| 1−dom−ref − K + Q_DAPC | 1 | 0.99 | 1 | 1.07 | 1.03 | 1.01 | 1.03 | 1.04 | 1.03 | 1.02 | 1 | 1.04 | 1.02 |
| 1−dom−alt − Naive | 1.62 | 1.64 | 1.11 | 1.11 | 1.07 | 1.13 | 1.11 | 1.11 | 1.08 | 1.42 | 1.24 | 0.97 | 1.04 |
| 1−dom−alt − Q_STRUCTURE | 1.3 | 1.26 | 1.09 | 1.03 | 0.97 | 1.11 | 1.11 | 0.99 | 0.99 | 1.22 | 1.08 | 0.98 | 0.95 |
| 1−dom−alt − Q_DAPC | 1.4 | 1.39 | 1.16 | 1.07 | 1 | 1.13 | 1.14 | 1.01 | 1 | 1.15 | 1.12 | 1.01 | 0.98 |
| 1−dom−alt − K | 1.01 | 1.05 | 0.97 | 0.99 | 0.95 | 0.99 | 1 | 0.98 | 0.98 | 0.98 | 1 | 0.92 | 0.95 |
| 1−dom−alt − K + Q_STRUCTURE | 1.02 | 1.06 | 0.97 | 1 | 0.94 | 1 | 1.01 | 0.97 | 0.98 | 0.99 | 1.02 | 0.92 | 0.94 |
| 1−dom−alt − K + Q_DAPC | 1.01 | 1.06 | 1 | 1.03 | 0.96 | 1 | 1 | 0.98 | 0.99 | 0.99 | 1.02 | 0.93 | 0.96 |
| 2−dom−ref − Naive | 1.34 | 1.33 | 1.17 | 1.13 | 1.11 | 1.1 | 1.1 | 1.13 | 1.1 | 1.39 | 1.1 | 1.06 | 1.09 |
| 2−dom−ref − Q_STRUCTURE | 1.13 | 1.1 | 1.14 | 1.06 | 1.04 | 1.06 | 1.1 | 1.04 | 1.02 | 1.26 | 1.01 | 1.07 | 1.02 |
| 2−dom−ref − Q_DAPC | 1.2 | 1.16 | 1.17 | 1.06 | 1.04 | 1.1 | 1.1 | 1.04 | 1.02 | 1.19 | 1.03 | 1.1 | 1.02 |
| 2−dom−ref − K | 0.99 | 0.99 | 1.09 | 1.02 | 1.01 | 1.01 | 1.01 | 1.02 | 1.02 | 1.02 | 0.96 | 1.02 | 1.01 |
| 2−dom−ref − K + Q_STRUCTURE | 1 | 1 | 1.08 | 1.03 | 1.01 | 1.01 | 1.01 | 1.02 | 1.01 | 1.02 | 0.97 | 1.02 | 1.01 |
| 2−dom−ref − K + Q_DAPC | 1 | 0.99 | 1.08 | 1.02 | 1.01 | 1.02 | 1.01 | 1.02 | 1.02 | 1.02 | 0.97 | 1.04 | 1.01 |
| 2−dom−alt − Naive | 1.37 | 1.37 | 1.19 | 1.09 | 1.12 | 1.12 | 1.07 | 1.14 | 1.1 | 1.36 | 1.14 | 1.07 | 1.09 |
| 2−dom−alt − Q_STRUCTURE | 1.12 | 1.1 | 1.15 | 1.01 | 1.03 | 1.08 | 1.06 | 1.03 | 1.01 | 1.21 | 1.03 | 1.07 | 1.02 |
| 2−dom−alt − Q_DAPC | 1.2 | 1.18 | 1.19 | 1.02 | 1.04 | 1.12 | 1.07 | 1.04 | 1.01 | 1.14 | 1.06 | 1.11 | 1.02 |
| 2−dom−alt − K | 1 | 1.01 | 1.11 | 1 | 1.02 | 1.01 | 0.99 | 1.02 | 1.01 | 1.01 | 0.99 | 1.02 | 1.02 |
| 2−dom−alt − K + Q_STRUCTURE | 1 | 1 | 1.09 | 0.99 | 1.01 | 1.01 | 0.99 | 1.01 | 1.01 | 1.01 | 1 | 1.02 | 1.01 |
| 2−dom−alt − K + Q_DAPC | 1 | 1.01 | 1.09 | 0.99 | 1.01 | 1.02 | 0.99 | 1.01 | 1.01 | 1 | 1 | 1.03 | 1.01 |

Figure 4.11: Inflation factors obtained for the different phenotypes and GWAS settings. a) Heatmap of the inflation factors obtained for each GWAS setting (row) and phenotype (column). b) Distribution of inflation factors obtained across phenotypes and genetic models for the different population settings.

under the null hypothesis of no association with the phenotype. A low inflation factor can be caused by an over-correction for population structure. Alternatively, as the studied population underwent selection for some traits of interest, the association between markers and the considered phenotype was consequently reduced. Markers were retained as significantly associated with a phenotype if their score is above the significance threshold in one (or more) GWAS setting whose inflation factor is between 0.95 and 1.1. The list of significant markers are presented in Supplementary Table F.1. I observed eight significant markers for the dry matter phenotype (on chromosomes 1, 2, 4, 11 and 12), four for the general impression score (on chromosomes 6, 8, 11 and 12), one for the percentage saleable phenotype (on chromosome 8), one for the sprouting phenotype (on chromosome 9) and one for the vigour trait (on chromosome 7). No markers were detected as significant for the other phenotypes. This small number is not surprising, as the samples used for this analysis have been selected for many of these traits. Hence, I do not expect a clear and strong signal from the main genomic regions controlling these traits. Instead, I am looking for the cause of more subtle variations in the phenotypes. And indeed, none of the detected significant markers are close to previously detected QTLs for these traits. It is interesting to note that most of these markers were detected as significant with only one genetic model, but several population settings. In such cases, the marker scores and estimated effects of the marker on the phenotype are consistent across the population settings. On the contrary, very few of these markers were detected as significant with several genetic models. This is not surprising as each genetic model searches for different patterns of association between a marker's dosage and the phenotype.

A common visualisation of GWAS results is in the form of a Manhattan plot, in which we represent the markers score along the genome, each chromosome being drawn one after the other. An example is depicted in Figure 4.12 a). With such plot, the genomic regions most associated with the phenotype are easily visualised as peaks or "towers". In this example, higher scores are observed towards the telomeres. This is due to the fact that the baits used to obtain genomics measurements targeted regions of high exonic densities, which occur away from the centromere of the chromosome and thus closer to the ends of the chromosomes (see Supplementary Figure F.2). Therefore, near the centromere, there are fewer genes with genomics measurements and so we can expect that very few of them have a significant association with the phenotype. In the present case, because I am interested in more subtle associations between markers and phenotypes, I also want to observe genomic regions that were assigned high but non-significant scores, i.e. the peaks in the Manhattan plots that are below the significance threshold. In order to compare the results obtained with different genetic models, and because marker scores are highly correlated between different population settings for a same genetic model, the population setting yielding the inflation factor closest to one was selected for each trait and genetic model. The "peaks" from each Manhattan plot were extracted as the markers with a score higher than the $99.99^{\text{th}}$ quantile of the score distribution (for the corresponding phenotype and GWAS setting). I represent the position of the peaks obtained

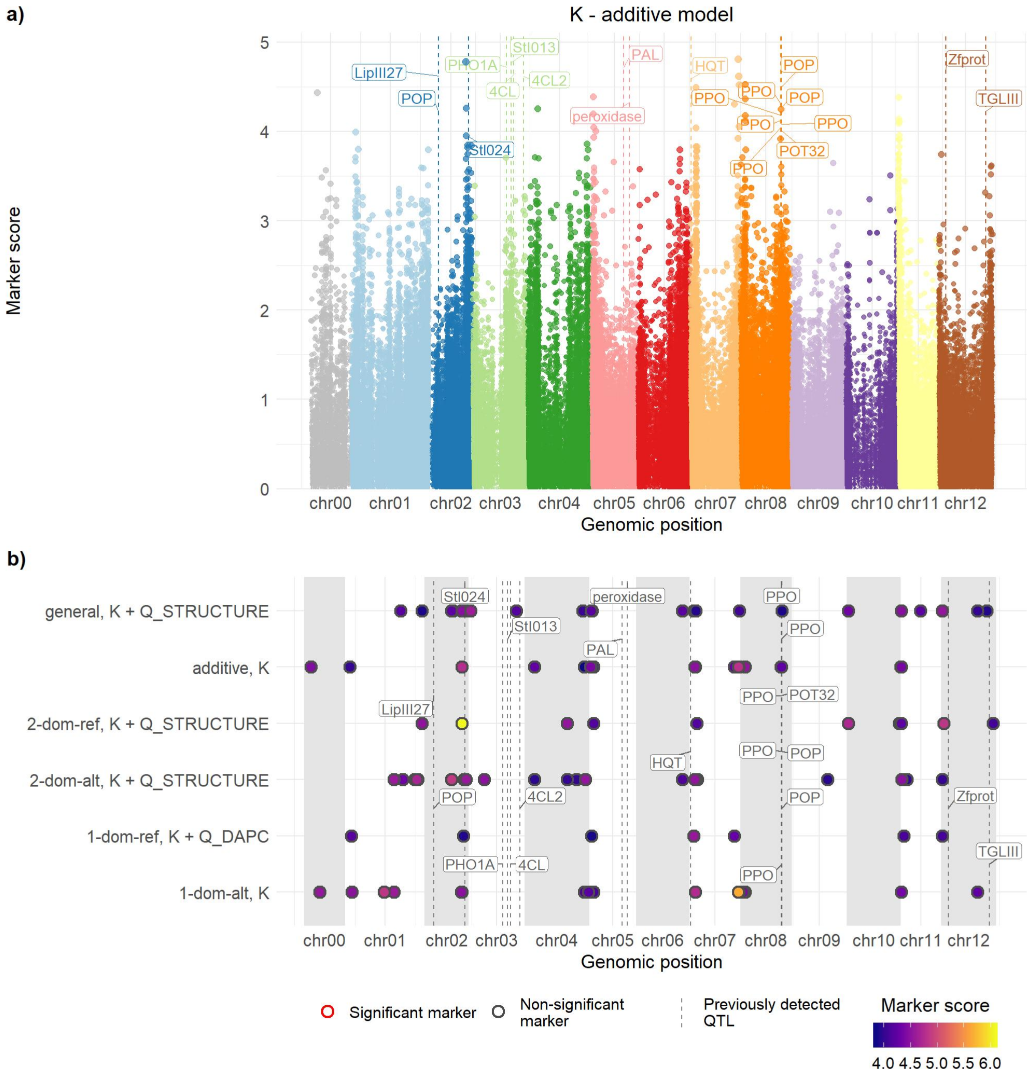


Figure 4.12: GWAS marker scores for the bruising mean score phenotype. a) Manhattan plot for the additive model and K setting. The marker scores (y-axis) are plotted for each marker along the genome (x-axis). b) Genomic positions of the peaks in the Manhattan plots for the six genetic models and corresponding optimal population setting. The colour of the points represents the marker score. In both plots, the position of previously detected QTLs along with the name of the candidate gene are represented with dotted lines.

with the six genetic models for the bruising mean score in Figure 4.12 b). It is interesting to see that some marker peaks are detected in the same region with all six genetic models (e.g. at the beginning of chromosome 7), while some peaks are only detected with one or a few genetic models (e.g. beginning of chromosome 4). Also, it can be noted that some of the markers with high but non-significant scores are located close to previously detected QTLs. For example, with all six genetic models we observe a peak in the GWAS scores on chromosome 2 around 41 to 45Mb, which is close to the StI024 SSR marker (PGSC0003DMG400010074 gene, encoding a hydroxyproline-rich glycoprotein

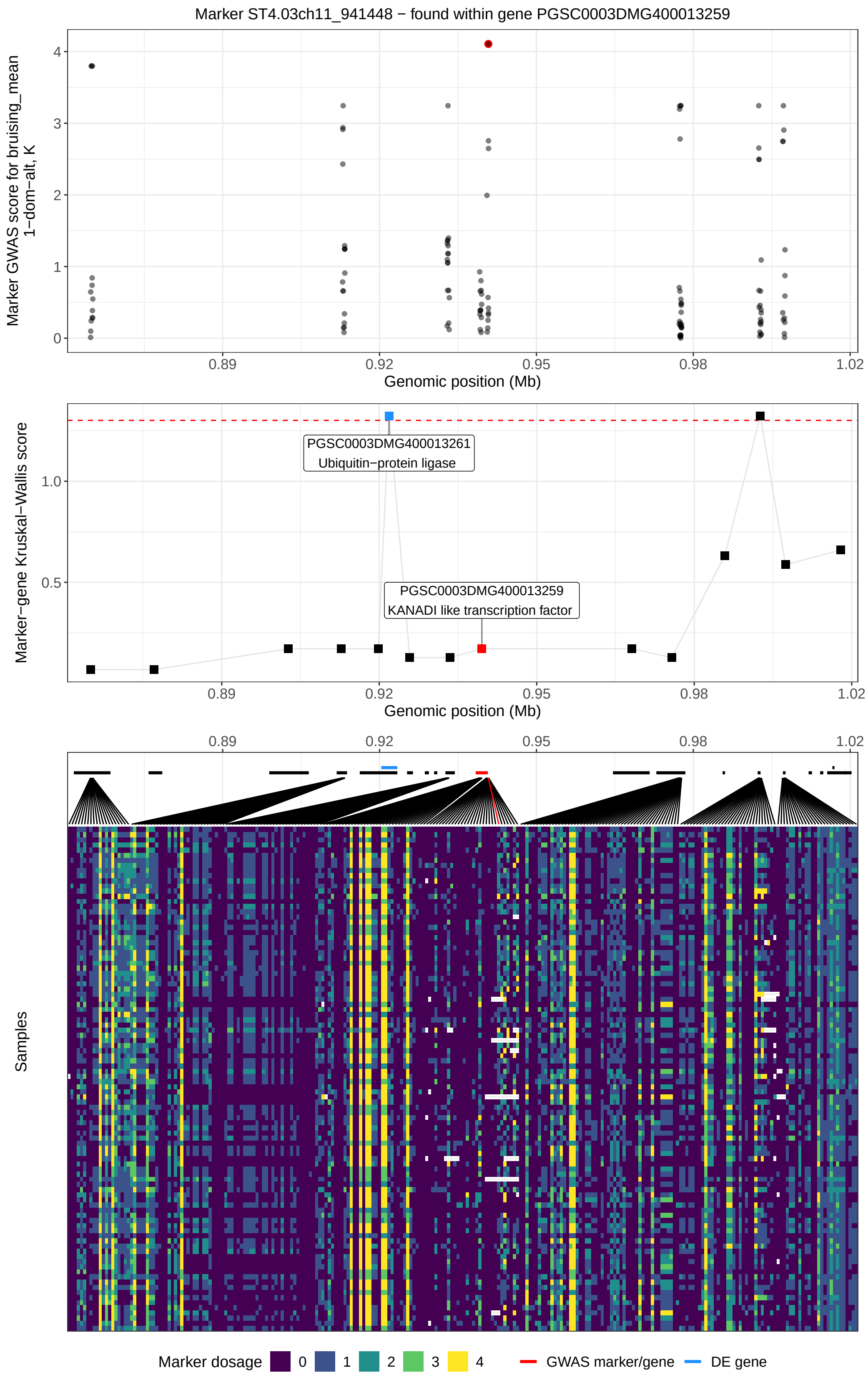


Figure 4.13: (Caption on next page.)

Figure 4.13: (Previous page.) Association between a high-scoring marker dosage and neighbouring transcribed genes' expression. The marker ST4.03ch11_941448, found just after the coding region of gene PGSC0003DMG400013259, was assigned a high score in the GWAS analysis for the bruising mean score trait (top panel - one point per marker). The association of its dosage with neighbouring transcribed genes is depicted in the middle panel, in which each square represents the Kruskal-Wallis score between the high-scoring marker and the expression of a neighbour gene. The bottom panel presents the position of the gene near which the considered marker is found (red rectangle) and neighbouring genes (black rectangles), including a transcribed gene detected as differentially expressed (in blue). The dosage of the observed variants in this region are plotted below, with lines indicating were they are found within the genes.

family protein) that was detected as significantly associated with a bruising phenotype by Urbany et al. (2011) in an association mapping study of tetraploid potatoes. The general and additive models also detected a few markers with high scores on chromosome 8 around 45Mb, within and close to genes encoding polyphenol oxidases or PPOs (PGSC0003DMG4000189{13, 14, 17, 19, 24, 25} and PGSC0003DMG400029576), including the POT32 gene (PGSC0003DMG400018916), which is expressed in potato tubers (Urbany et al., 2011). These enzymes catalyse the oxidation of phenolic compounds, causing tissue discolouration and browning, and were detected as QTLs in several studies (e.g. Werij et al., 2007; Hara-Skrzypiec et al., 2018). Similarly, a non-significant GWAS score peak was found for the maturity trait very close to the StCDF1 gene, which is a major maturity locus (Kloosterman et al., 2013). In general, across the phenotypes, some of the high-scoring variants were found within genes previously detected as QTLs for the considered traits. This highlights the importance of looking at genomic regions with high GWAS scores that are not significant, as these can provide useful biological information.

### 4.4.8 Comparing GWAS results with a differential expression analysis

As transcripts measurements were taken after bruising the tubers, I expect the data to reflect the expression of the genes in response to the bruising. Therefore, 41 samples with the lowest bruising mean scores and 41 samples with the highest bruising mean scores were selected, to perform a differential expression analysis. I obtained 158 significantly differentially expressed genes (DE genes) with an adjusted p-value under 0.05. Amongst them, 92 transcribed genes were found upregulated, and 62 downregulated. I did not use a threshold on the fold-change of the transcribed genes in order to detect genes with small but consistent changes between the groups. Comparison between two different multiple-testing correction methods, namely FDR and IHW, showed that both methods were in very good agreement with respect to the transcribed genes detected as differentially expressed, with only 54 transcribed genes found as significant with one method but not the other. This is to be expected, as with both methods genes with a very small p-value will still be considered significantly differentially expressed after correction. A gene set enrichment analysis was performed to detect GO terms enriched for differentially expressed genes. The GO terms 'protein binding' (`GO:0005515`), 'ATP binding'

(`GO:0005524`) and ‘protein kinase activity’ (`GO:0004672`), all three representing molecular function terms, were significantly enriched for DE genes. Unfortunately, the lack of proper translation of gene IDs into KEGG prevented an over-representation analysis of metabolic pathways. This highlights the fact that proper gene annotation is essential to interpret the results of a differential expression analysis.

From the set of transcribed genes detected as significantly DE, two transcribed genes were also detected as containing a high-scoring marker from the GWAS analysis, for the bruising mean score phenotype. The relationship between one of these high-scoring marker, corresponding gene expression and bruising mean score is shown in Supplementary Figure F.5. The fact that there was no overlap between the GWAS and differential expression results is not surprising. First, the causal variants might control the phenotype without impacting gene expression, for example by affecting post-transcriptional events, e.g. the conformation of a protein. Also, in the case that the causal variants do influence the phenotype through changes in gene expression, I do not expect the GWAS analysis to pinpoint true causal variants, but rather genomic regions of interest. The detected markers may be located close to the true causal variant but on a different gene/regulatory region. Indeed, I observed that for most high-scoring markers there is little to no relationship between their dosage and the expression of the gene in which they are found. Only 19 of the 117 high-scoring markers detected for the bruising mean phenotype were found to have a significant relationship between their dosage and the expression of the genes in which they are found, and none of the corresponding transcribed genes were differentially expressed. One possibility is that these high-scoring markers, via linkage with the true causal variant, are associated with the expression of a neighbouring gene, which does play a role in controlling the trait of interest. However no significant difference was found between the distance to a GWAS high-scoring marker of differentially expressed genes and of non-differentially expressed genes (p-value = 0.13).

I illustrate this for one marker in Figure 4.13. The dosage of the high-scoring marker depicted is not associated with the expression of the gene closest to which it is found, however it is significantly associated with the expression of a nearby gene which has been found to be differentially expressed. This example emphasises the importance of considering the results of a GWAS analysis with caution, and the usefulness of combining several omics layers in order to detect molecular features involved in controlling a trait of interest, rather than relying on significant markers only. Yet another possibility is that the variants affect genes encoding for transcription factors that have been found to occur in small numbers in cells, and could thus be missed by the differential analysis. Lastly, the variants could affect the expression of a gene whose impact on the phenotype would be missed due to the timing of the sampling. Indeed, samples were obtained two hours after bruising, while the bruise was scored after 24 hours.

To summarise the information uncovered with the GWAS and differential expression analysis, I visualised the genomic position of high-scoring GWAS markers for the bruising mean phenotype (across the different valid GWAS settings) alongside the position of significantly differentially expressed genes, as well as candidate genes previously identified in the literature (Figure 4.14). We can see that even though the genes highlighted by the GWAS and differential expression analysis are not identical, there are nevertheless often found clustered in specific genomic regions, for example at the end of chromosome 2, at the beginning and end of chromosome 7, end of chromosome 8 and beginning of chromosome 11, with smaller clusters spread throughout the genome. Some of these clusters coincide with previously reported QTLs, i.e. the Stl024 gene in chromosome 2 or PPO genes in chromosome 8, but others are not located near previously reported QTLs, and could thus provide new insights into the mechanisms of potato bruising, such as for example the end of chromosome 7 or beginning of chromosome 11.

### 4.4.9 Genes co-expression network

One question that arises is whether the genes of interest (genes with high-scoring markers, DE genes or previously uncovered candidate genes) are involved in similar biological pathways. One way to answer this question is to reconstruct the co-expression network among transcribed genes, based on the correlation between their expression. The hypothesis is that transcribed genes that are close in the co-expression network, i.e. with high correlation between their expression, are involved in a common pathway. The VST transformed count data was used to reconstruct a co-expression network among the 25,163 genes with transcriptomics measurements. Modules of highly connected transcribed genes were detected based on the topological overlap between pairs of genes, i.e. the number of common neighbours between two considered genes. I identified 196 modules of size ranging from 10 to 5,295. In addition, 7,410 transcribed genes (29.4%) were not assigned to any module. For each module, its eigengene was computed, that summarises the expression profile of the genes in the module across the measured samples. The eigengenes of 32 modules were detected as significantly correlated with the bruising mean phenotype (14 modules with a significant positive correlation and 18 modules with a significant negative correlation). This suggests that multiple biological processes might be involved in the response to bruising. The GO terms significantly enriched in these modules are presented in Supplementary Table F.2.

I also tested whether the different uncovered modules were enriched for genes containing GWAS high-scoring markers, DE genes, or previously identified candidate genes. The correlation p-value of each module's eigengene for the bruising mean score phenotype as well as the different enrichment scores are depicted in Figure 4.15. The 301 genes of interest (i.e. either containing high-scoring markers, differentially expressed or previously detected as candidate QTLs) for which we had transcriptomics measurements were scattered across 56 modules, and 40 (13.3%) were not assigned to any module. Six out of the 32 modules whose eigengene was significantly correlated with the

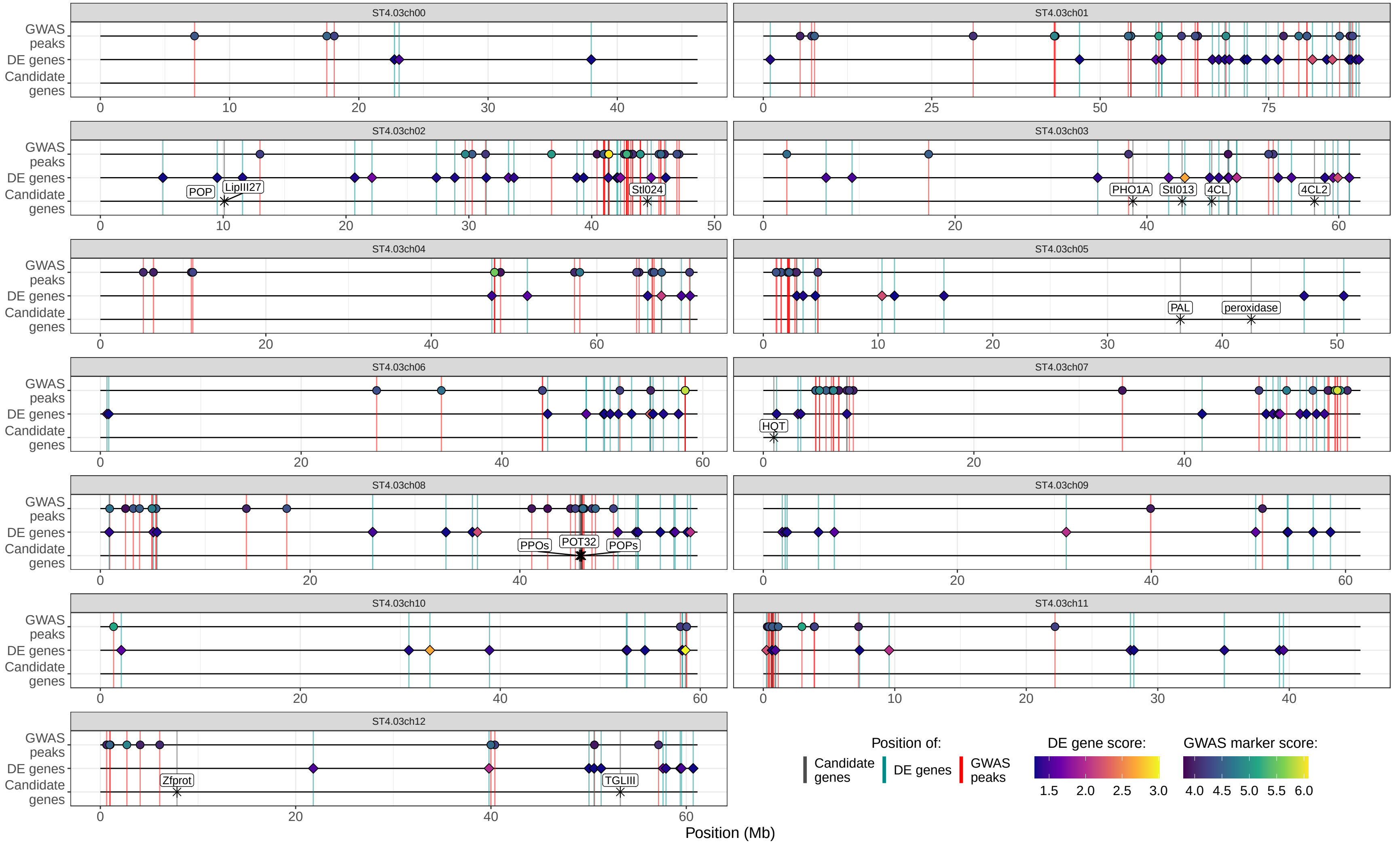


Figure 4.14: Genomic position along the 13 chromosomes of the GWAS high-scoring markers for the bruising mean score trait (circles, red lines), differentially expressed genes (diamonds, blue lines) and candidate genes previously uncovered (stars, grey lines).

phenotype were also found enriched for DE genes. This is to be expected as the eigengene represents the expression of transcribed genes within the module, so a high correlation of the eigengene with the trait implies that the expression of at least some genes within the module is also correlated with the trait. Genes with high-scoring markers were found in modules that were not enriched for DE genes nor associated with the trait, and previously detected candidate genes also found in modules not enriched for DE genes nor for GWAS genes. Interestingly, when comparing the adjacency matrix restricted to the genes of interest, I observed that a large fraction of the DE genes shared high adjacency values between them, while most genes containing high-scoring markers had smaller adjacency values with other interesting genes. This could be due to the fact that some genes containing high-scoring markers are unrelated to the trait of interest but only physical neighbours of causal variants, as discussed in the previous section. Taken together, this co-expression analysis provide further information about genes possibly involved in controlling tuber bruising. Genes co-expressed with differentially genes can be further investigated to detect biological pathways involved in tuber bruising. Another explanation could be that the causal variants affect the phenotype through other means than via modification of gene expression.

## 4.5 Concluding remarks

Genome wide association studies are an invaluable tool to understand the genomic component of variation in traits of interest. In this study, I demonstrated how partial genomics measurements, together with transcriptomics data, can be used to uncover interesting genomic regions and candidate genes associated with a considered trait in a tetraploid organism with complex population structure. I focused on tuber bruising as well as several agronomic traits of the autotetraploid potato. Despite the plants used for the study being selected for some of these traits, several significant markers associated with the phenotypes were uncovered. As I anticipate that the selection process removed variants in major QTLs for the traits, these significant markers likely provide clues about genetic components associated with more subtle variations in the phenotypes. This is an important result, as it demonstrates that breeding populations that were not specifically designed for association analysis can still be used to investigate relationships between genotype and phenotype.

Although the significant markers uncovered did no co-localise with previously reported QTLs, a careful investigation of the high-scoring markers, i.e. markers with high GWAS scores but below the significance threshold, revealed peaks close to or within genes previously found related to the traits of interest. This highlights the importance of considering non-significant results in association studies, as they can be biologically relevant. Also, the choice of a specific multiple testing correction method influences greatly which markers are retained as significantly associated with the phenotype. In this study, I chose to use the Bonferroni correction when selecting the significance threshold

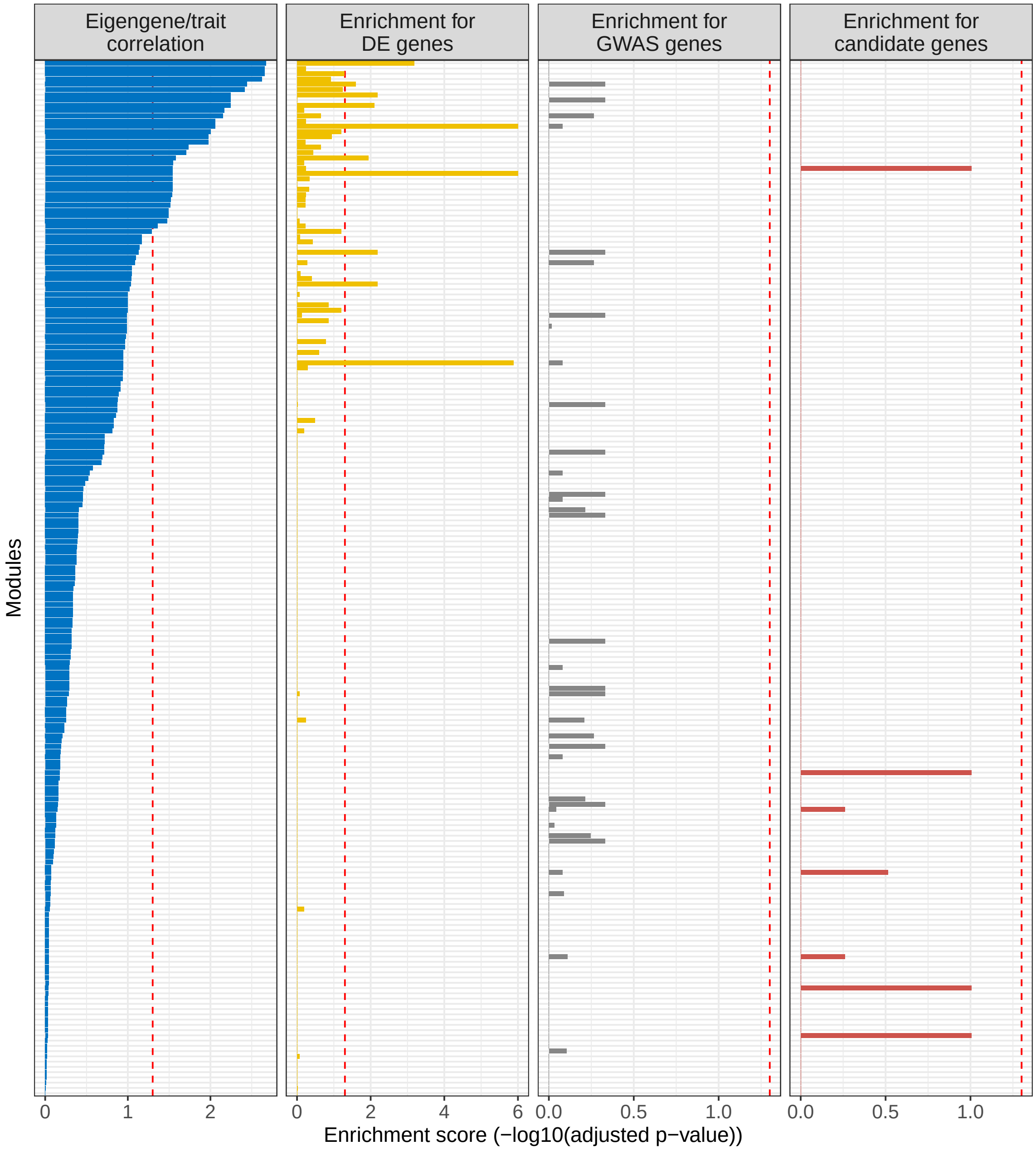


Figure 4.15: Correlation score between the eigengene of the different modules detected in the co-expression network and the bruising mean score (left panel), and enrichment score of the modules for (from left to right) differentially expressed genes, genes containing high-scoring markers, and candidate QTLs from the literature. Scores are computed as -log10 of the corresponding adjusted p-value. Scores higher than six were set to six for ease of visualisation. The 0.05 p-value threshold is indicated with a red dotted line.

for markers' scores. I considered the Bonferroni correction more appropriate than a FDR-based approach in this case as, due to the individuals being subject to selection for some traits of interest, I do not expect major causal QTLs to be detected. Therefore, a FDR-based correction would have resulted in many false positives for a relatively small increase in statistical power. By performing a differential expression analysis on samples with extreme phenotypes, I showed that markers with high GWAS scores and significantly differentially expressed genes where located in clusters throughout the genome, thus providing evidence that these regions are of interest for the investigated traits. This comparison between differentially expressed genes and high-scoring markers provides a way to complement capture sequencing as it allows to pin-point possible causal genes near which the true causal variants might be located, even though no genomics data was recorded for these genes. Moreover, it offers some intuition about possible mechanisms by which genetic variations affect the trait of interest through the expression of important genes. Note that except for the GWAS analysis, the FDR correction was used to correct for multiple testing, as it is less conservative than the Bonferroni correction and thus allowed to detect more signal from the datasets. Using FDR-based corrections that account for useful covariates has been shown to be useful in differential expression analyses (Korthauer, Kimes, et al., 2019) and was done to detect significantly differentially expressed genes from the transcriptomics dataset.

For crops such as potato, it is important to account for the fact that the cultivars used in the association panel are probably related to some extent, and adjust accordingly the statistical model used to perform the analysis. In this study, I demonstrated the importance of including information about individual relatedness as well as samples admixture in the model in the case where the samples are highly related. Even though STRUCTURE and DAPC detected a similar clustering of the samples into subpopulations, the former was more effective in reducing the effect of population stratification, as it informed about samples admixture. This result is likely to change for experiments with a different population structure. I expect that for panels with more unrelated samples, the DAPC posterior membership probabilities will be sufficient to account for population stratification, which has been the case for Rosyara et al. (2016). As previously mentioned by Rosyara et al. (2016) and Sharma et al. (2018), inflation factors provide a useful guide for assessing the effect of population structure on the results of a GWAS analysis. Moreover, these results show the usefulness of investigating different scenarios linking the markers' dosage to the phenotypic value (i.e. different genetic models), a feature offered by the package GWASpoly. Different genetic models highlight the contribution of different markers and provide complementary outcomes.

The findings from this association study can be used to focus the search for interesting markers for breeding selection in the genomic regions highlighted. In addition, I am planning to go one step further and combine genomics, transcriptomics, metabolomics and phenotypic data to reconstruct the causal flow of information from genotype to phenotype.

## 4.6 Acknowledgements

Thank you to Plant and Food Research for providing the data used in this chapter. In particular, I am grateful for the help of Dr Samantha Baldwin, Dr Susan Thomson, Dr Rebecca Bloomer and Katrina Monaghan for generating and pre-processing the genomics, transcriptomics and phenotypic datasets. This work was supported by The New Zealand Institute for Plant and Food Research Limited Strategic Science Investment Fund (SSIF) from the New Zealand Ministry of Business, Innovation and Employment (MBIE).

## Chapter 5

# Uncovering causal relationships from genotype to phenotype using multi-omics data in autotetraploid potatoes

### 5.1 Introduction

Integrating measurements obtained at different cellular scales, i.e. different "omics" datasets, offers the potential to reconstruct an unprecedented picture of the molecular mechanisms at play within cells. The central dogma of molecular biology requires extensions: there is extensive interaction and feedback between different molecule types, i.e. DNA, RNA, proteins and metabolites (Angelin-Bonnet et al., 2019). Variations in the DNA can affect the expression of genes, or the activity of the encoded proteins (Albert & Kruglyak, 2015). Non-coding RNAs can also control the different steps of gene expression (see for example Geisler & Coller, 2013). Enzymes catalyse metabolic reactions, and metabolites can effect a feedback on the expression or lifetime of target proteins (e.g. Serganov & Patel, 2012). Thus, focusing on one of these omics layers only amounts to overlooking the complexity of regulatory interactions at play and prevents us from reconstructing a complete overview of the biological processes involved. As technology progresses, and the monitoring of different omics becomes routine, including on the same set of samples, more datasets are generated that are ideally suited to omics data integration. A number of statistical methods have been proposed to this end, with different objectives. Methods based on correlation (Peng et al., 2018) or clustering (Acharjee et al., 2016) aim at detecting from the different omics datasets groups of co-regulated features, potentially involved in similar biological networks. Some algorithms seek instead to perform dimension reduction, to summarise the information contained in the different datasets and identify groups of features driving the variation amongst the observations or samples. Examples include multiple coinertia analysis (MCIA – Meng et al., 2014), orthogonal projections to latent structure for n matrices (OnPLS – Srivastava et al., 2013), Generalised Canonical Correlation Analysis (Tenenhaus et al., 2017) or block sparse PLS-DA, known as the DIABLO algorithm (Singh et al., 2016). Some of these tools make use of regularisation – such as LASSO (Tibshirani, 1996), Elastic Net (Zou & Hastie, 2005) – to perform feature selection, in order to retain features from the different datasets that are best associated with a phenotypic outcome of interest (e.g. Lê Cao et al., 2008; Cai & Huo, 2020; Singh et al., 2016).

In the present work, I am interested in methods that focus on reconstructing the flow of information from genotype to phenotype, specifically in the context of tuber bruising in autotetraploid potatoes. To combine genomics, transcriptomics and phenotypic data, several studies have made use of uncovered QTLs as causal anchors in order to reconstruct causal relationships between different molecular features. For example, given an intermediate phenotype (e.g. gene, protein or metabolite) associated with a genomic variant that coincides with a QTL region for a trait of interest, Millstein et al. (2009) proposed a Causal Inference Test that assesses the causal status of the intermediate phenotype as a mediator between the QTL and the phenotype. They show that this test can be used to reconstruct a causal transcription regulatory network for yeast. Zhu et al. (2012) used Bayesian networks to model causal regulatory relationships between causal variants, transcripts and metabolites. Recently, Qiu et al. (2020) used a combination of differential analysis, QTL mapping and Mendelian randomisation to detect biomarkers from genomics, methylomics, transcriptomics and metabolomics datasets with a causal effect on bone mineral density. Mendelian randomisation uses SNPs as instrumental variables to test for causal relationships between variables. Causal inference is indeed a desirable way to integrate different omics datasets, as it allows us to move beyond mere co-regulation and to uncover the biological mechanisms at play. However, causal inference is still a field under active research, and its use in systems biology is not yet widespread. One of the reasons is that a number of assumptions must be fulfilled in order to detect true causation, which are hard to check or even violated in biological systems. Nevertheless, causal inference methods have been used on biological datasets and have provided interesting insights, as they were able to detect previously reported associations as well as detect new relationships between molecular actors and phenotypes (Neto et al., 2008; Peñagaricano et al., 2015a). In Chapter 3, I have shown that, when applied to small simulated transcriptomics datasets, causal inference methods are able to detect true relationships between the different gene products. In this analysis, I aim at expanding their use in an effort to bring together genomics, transcriptomics and metabolomics measurements.

I expand here the analysis undertaken in Chapter 4, in which I study the molecular mechanisms of tuber bruising in tetraploid potatoes. In Chapter 4, I focused on the genetic component of tuber bruising, by investigating the associations of genomic variants with the phenotype. In the present study, I complement these results by turning my attention to biological mechanisms that mediate this genotype-phenotype association, by analysing metabolomics measurements obtained from the same half-sibling population. I perform the integration of the genomics, transcriptomics, metabolomics and phenotypic datasets in two steps. The first step aims at reducing the dimension of the datasets by means of feature selection. I use the DIABLO algorithm, implemented in the `mixomics` R package, to retain co-regulated features associated with the phenotype. For the second step, I explore the use of a number of state-of-the-art causal inference methods on the selected features. This permits the construction of a multi-omics causal network relating to potato tuber bruising. I contrast the results obtained with those from single-omics analyses performed on each of the datasets. In addition, I make use of a number of causal queries, that seek different types of causal relationships between the molecular features (e.g. parental or ancestral relationships), to interpret the inferred graphs. These causal queries provide an efficient and intuitive way to extract information from the reconstructed graphs.

## 5.2 Contribution

The metabolomics data acquisition and pre-processing (step M1 in Figure 5.1) were performed by the Plant and Food Research metabolomics team, Martin Shaw and Nigel Joyce.

## 5.3 Materials and methods

A schema of the analysis workflow used throughout this chapter is presented in Figure 5.1.

### 5.3.1 Plant materials and phenotyping

The potato material used for this analysis is presented in Section 4.3 of Chapter 4. Briefly, several cultivars were crossed in a half-sibling design, and the resulting progeny lines underwent two rounds of selection. Retained progeny lines were cultivated and subsequently subjected to a bruising experiment. For each progeny line, three tubers were harvested for each of two biological replicate plants, and each tuber was bruised on two sides by dropping a lead weight from controlled height. The site of bruising was recorded by inking the weight before bruising. After 24 hours, tubers were sliced at the site of bruising, and an image was taken of the bruising for the three tubers of each biological replicate per genotype. A bruising score was attributed to each tuber upon visual inspection of the images, from zero (no visible bruising) to five (extensive dark bruise), in accordance to a predefined visual scale. For each biological replicate, the bruising mean score was obtained as the average of the bruising score attributed to each of the three tubers. In addition, a bruising mean score was computed for each genotype as the average bruising score of the six tubers (i.e from both biological replicates).

### 5.3.2 Metabolomics dataset

#### Data acquisition

Two hours after bruising, samples were taken from the bruising site on each tuber used for the bruising experiment, and the samples obtained from the three tubers collected from a same biological replicate

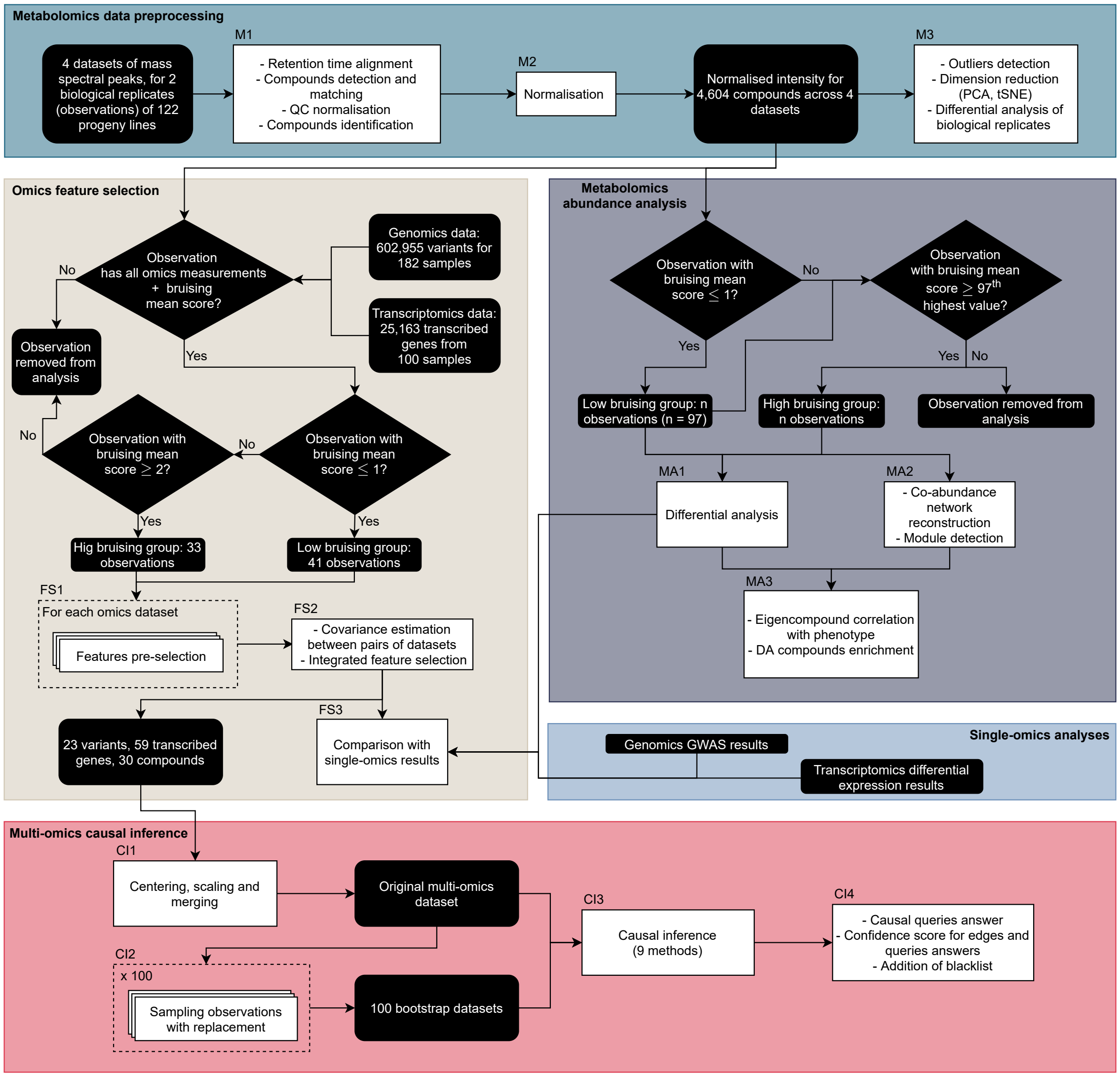

Metabolomics data preprocessing
4 datasets of mass spectral peaks, for 2 biological replicates (observations) of 122 progeny lines
M1
- Retention time alignment
- Compounds detection and matching
- QC normalisation
- Compounds identification
M2
Normalisation
Normalised intensity for 4,604 compounds across 4 datasets
M3
- Outliers detection
- Dimension reduction (PCA, tSNE)
- Differential analysis of biological replicates
Omics feature selection
Observation has all omics measurements + bruising mean score?
No
Yes
Genomics data: 602,955 variants for 182 samples
Transcriptomics data: 25,163 transcribed genes from 100 samples
Observation removed from analysis
No
Observation with bruising mean score ≥ 2?
No
Observation with bruising mean score ≤ 1?
Yes
Yes
Hig bruising group: 33 observations
Low bruising group: 41 observations
FS1
For each omics dataset
Features pre-selection
FS2
- Covariance estimation between pairs of datasets
- Integrated feature selection
FS3
23 variants, 59 transcribed genes, 30 compounds
Comparison with single-omics results
Metabolomics abundance analysis
Observation with bruising mean score ≤ 1?
No
Observation with bruising mean score ≥ 97th highest value?
Yes
Yes
No
Low bruising group: n observations (n = 97)
High bruising group: n observations
Observation removed from analysis
MA1
Differential analysis
MA2
- Co-abundance network reconstruction
- Module detection
MA3
- Eigencompound correlation with phenotype
- DA compounds enrichment
Single-omics analyses
Genomics GWAS results
Transcriptomics differential expression results
Multi-omics causal inference
CI1
Centering, scaling and merging
Original multi-omics dataset
CI2
x 100
Sampling observations with replacement
100 bootstrap datasets
CI3
Causal inference (9 methods)
CI4
- Causal queries answer
- Confidence score for edges and queries answers
- Addition of blacklist

Figure 5.1: Schema of the analysis workflow used throughout the chapter. Input/output datasets and endpoints are presented in black rounded boxes and analysis steps are shown in white rectangle boxes. The coloured rectangles outline different themes in the analysis.

were pooled and snap-frozen. Samples were collected for each of the two biological replicates for 122 progeny lines, all arising from a cross involving Crop52. Untargeted metabolomics measurements were then obtained for the 244 observations by liquid chromatography-mass spectrometry; a copy of the protocol for samples processing and analysis can be found in Appendix G, Section G.1. Measurements were taken under aqueous normal phase conditions (H column) as well as reverse phase conditions (C column), and the eluent from each column was scanned with in both the negative (n) and positive (p) ion mode. This resulted in four different analyses or modes per sample, denoted as Hn, Hp, Cn and Cp to indicate both the column and ion mode used. As two biological replicates were used for each progeny line, the samples labelled as “Replicate 1” across the genotype lines were processed first, and all samples labelled as “Replicate 2” were processed in a second time.

The resulting mass spectral scans were processed with the Compound Discoverer software (box M1 in Figure 5.1). Following a first exploratory analysis of the data showing an effect of the samples processing order on the compounds retention time, retention times were aligned to compensate for drifts occurring throughout sample processing. Detected compounds with a molecular weight different of less than five parts per million and a retention time difference of less than 0.4 minutes were matched. Peak areas were extracted and normalised using the QC samples measured throughout sample processing, to correct for batch effects. Several internal databases (including a snapshot of the mzCloud database) were then consulted to extract the name, formula and structure of the compounds. Alternatively, the chemical formula of unknown compounds was predicted from the spectral data. Ultimately, normalised intensities were obtained for 953 compounds with the Cn mode, 2,570 compounds with the Cp mode, 229 compounds with the Hn mode and 853 compounds with the Hp mode (4,604 compounds in total).

The normalised area intensities of the compounds for each of the four modes (Cp, Cn, Hp, Hn), thereafter referred to as the Cp, Cn, Hp and Hn datasets, respectively, were extracted and further processed with the R package `MetaboDiff` (Mock et al., 2018). In addition, the four datasets were merged to obtain a combined dataset with all 4,604 identified compounds. Subsequent analyses were performed on each of the four datasets separately as well as on the combined dataset, for comparison. The area intensities were normalised using the Variance Stabilising Transformation (VST) (Huber et al., 2002) in order to mitigate the mean-variance bias observed (box M2 in Figure 5.1). The following exploratory analyses were performed on these normalised intensities (box M3 in Figure 5.1). The `tsne` R package (Donaldson, 2016) was used to apply the t-distributed Stochastic Neighbour Embedding (tSNE) algorithm (Van Der Maaten & Hinton, 2008) on the samples, a non-linear dimension reduction technique aiming at projecting samples onto a two-dimensional space. In this reduced space, similar samples (as defined by a Euclidean distance metric) are positioned closely with high probability, while dissimilar samples are placed far apart with high probability. A principal component analysis (PCA) was performed to assess the impact of samples processing order on the resulting intensities. In addition, the `metaboDiff` package was used to perform a differential analysis between the first biological replicate of each sample (processed first in the experiment) and their second biological replicate (processed last), via a two-sided two-sample t-test with equal variance. The p-values obtained were corrected for multiple testing using the FDR correction (Benjamini & Hochberg, 1995) (see Section 4.1 for a discussion on multiple testing correction methods). Compounds with an adjusted p-value below 0.05 and a log2 fold-change above 1.5 were considered significantly differentially abundant. The tSNE analysis was then repeated without the compounds detected as differentially expressed, to test whether they were the only drivers of the difference between biological replicates.

### Differential analysis

All biological replicates with a bruising mean score of one or less were classified as the “low bruising” metabolomics group. The same number of biological replicates with the highest mean bruising scores were then selected for the “high bruising” metabolomics group. This yielded two groups of 97 observations each (where an observation is one biological replicate of a progeny sample), which were used for the differential expression analysis. The low bruising group gathered 67 different genotypes, 30 of them with both observations in the low bruising group. In the high bruising group, 70 different genotypes were gathered, 27 of them having both replicates in the high bruising group. In addition, 23 genotypes had one replicate clustered in the low bruising group and the other in the high bruising group. Observations that did not group in the low nor the high bruising group were discarded from further analyses. The `MetaboDiff` package was used to perform the differential analysis between the two phenotype groups. The compounds p-values obtained were automatically corrected for multiple testing using the FDR procedure (cf Section 4.1). Compounds were considered significantly differentially abundant if their adjusted p-value was below 0.01.

### Network co-abundance reconstruction

The `WGCNA` package (Langfelder & Horvath, 2008) was used to reconstruct the co-abundance network between the compounds, using the VST-transformed area intensity datasets (box MA2 in Figure 5.1). The algorithm starts by computing a similarity matrix for all compounds, where the similarity between any pair of compounds corresponds to the correlation between their normalised intensity across the observations. The adjacency matrix is then elevated to a soft thresholding power $\beta$, in order to remove noise from the correlations. For each dataset, the soft-thresholding power was set as follows: three for the Hp and combined datasets, four for the Cn and Hn datasets, and 14 for the Cp dataset. These values were selected by varying for each dataset the soft-thresholding power, and selecting the smallest value yielding both a scale-free network and a small mean connectivity coefficient. The resulting adjacency matrix is used to compute a topological overlap matrix (TOM) describing for any pair of compounds the overlap between their neighbours in the adjacency graph. The opposite of the TOM matrix (i.e. 1 - TOM) is used as input for performing a hierarchical clustering on the observations. The resulting dendrogram is split using the dynamic tree cut method, in order to extract modules of highly connected compounds from the adjacency graph. The `deepSplit` parameter was set to two and the minimum cluster size to 10. Both values were chosen to detect small densely connected modules rather than large and more loosely connected modules. The value for `deepSplit` was kept to two, contrary to the transcriptomics analysis in which it was set to three, as the number of compounds in the dataset was smaller than the number of transcribed genes measured in the transcriptomics dataset (4,604 compounds vs 25,163 transcribed genes). Modules with similar intensity profiles were merged. For each resulting module, its eigencompound was computed, as the samples coordinates for the first principal component resulting from a PCA applied to the observations based only on the compounds in the module (box MA3 in Figure 5.1). These eigencompounds provide a summary of the intensity profile distribution of the module across the observations. A correlation test was used to assess the association between each module eigencompound and the bruising mean score across the observations. In addition, an enrichment test of each module for differentially abundant compounds was performed using the R package `gage` (Luo et al., 2009), which compares the mean compounds' differential score (-log10 of their differential abundance adjusted p-value) for compounds within the module to a same number of random compounds outside of the module via a prototype two-sample t-test. The resulting p-values were adjusted for multiple testing with the FDR correction (cf Section 4.1). Modules with an adjusted p-value below 0.05 were considered as significantly correlated with the bruising mean score.

### 5.3.3 Genomics dataset

#### Data acquisition

Genomics data was obtained from young leaf tissue material for the progeny samples as well as some of the parents by capture-bait sequencing. Details of the data collection and preprocessing are presented in Section 4.3.2. Briefly, the reads obtained were used to call genomic variants. The resulting variants were then filtered according to their quality and fraction of missing values. In addition, samples were filtered according to the fraction of missing values, to the status of the cultivar (parent or progeny) and to their inclusion in the bruising experiment. The resulting dataset provides information about the dosage of 602,955 genomic variants across 13 chromosomes, for 182 samples (including 20 samples corresponding to parent genotypes).

#### Genome-wide association study

The genotype data (biallelic SNP and dosage) was used to assess the population structure among the genotyped lines (see Section 4.3.4). This information was subsequently used to perform a genome-wide association study (GWAS - Section 4.3.5), in order to detect genomic regions whose variation is associated with traits of interest, notably the bruising mean score. Even though statistical significance was not reached when correcting for multiple testing, markers with high GWAS scores were labelled as GWAS high-scoring markers and retained for further analysis.

### 5.3.4 Transcriptomics dataset

#### Data acquisition

Two hours after the bruising experiment, samples were taken from one bruised side of each tuber in order to collect transcriptomics measurements. Details of the data collection, preprocessing and filtering are presented in Section 4.3.3. Note that only one biological replicate for each progeny line was used for the transcriptomics measurements, yielding RNA levels for 25,163 transcribed genes across 100 unique progeny samples.

#### Differential analysis

Samples with RNA measurements and with a bruising score of one or lower were grouped as the "low bruising" transcriptomics group (as presented in Section 4.3.6). The same number of samples (with RNA measurements) and with the highest bruising scores were then selected as the "high bruising" transcriptomics group. This yielded two groups of 41 samples each. The difference in the number of observations between the low and high bruising transcriptomics groups and the low and high bruising metabolomics groups is due to the fact that (i) only one biological replicate of a given progeny line was used for the transcriptomics measurements and (ii) some progeny lines were used for the metabolomics measurements but not the transcriptomics measurements. A differential expression analysis was performed to compare the expression of the transcribed genes between these two groups, using the `DESeq2` R package (Love et al., 2014). Two different methods were compared to correct the p-values for multiple testing: the traditional FDR correction, and independent hypothesis weighting (Ignatiadis et al., 2016) that accounts for the total read counts of the transcribed genes to improve the power of detection. Transcribed genes with either corrected p-value below 0.05 were retained as significantly differentially expressed.

#### Co-expression network reconstruction

Similarly to the metabolomics analysis, the `WGCNA` package was used to reconstruct a co-expression network among the transcribed genes (see Section 4.3.7). Because of the large number of transcribed genes in the dataset, the co-expression network detection could not be run on the entire dataset at once. Instead, the transcribed genes were first clustered into four main groups (using projective k-means clustering), and the module detection procedure presented in Section 5.3.2 was applied separately to each group. The soft-thresholding power $\beta$ was set to six, the `deepSplit` parameter to three and the minimum cluster size to 10.

### 5.3.5 Multi-omics data integration

In order to integrate the multi-omics datasets, 98 observations (an observation is a biological replicate of a progeny sample) for which genomics, transcriptomics and metabolomics measurements as well as a bruising mean score were available were retained. From these, 41 observations with a bruising mean score of one or less were selected, and were classified as the "low bruising" group, and 33 observations with a bruising mean score of two or more were selected, which constituted the "high bruising" group. The remaining samples were removed from the datasets. For all subsequent analyses, the combined metabolomics dataset (i.e. with compounds identified with any of the four modes) was used.

Because of the large size of the omics datasets (602,955 genomic variants, 25,163 transcribed genes and 4,604 compounds), a feature pre-selection step was first performed separately on each dataset using the sparse Partial Least Squares-Discriminant Analysis (sPLS-DA) algorithm (Lê Cao et al., 2011) implemented in the R package `mixOmics` (Rohart et al., 2017 – box FS1 in Figure 5.1). The algorithm jointly performs feature selection and dimension reduction, by seeking linear combinations of subsets of variables, termed latent components, that best discriminate the outcome groups among the observations. In this case, the outcome groups correspond to the high and low bruising groups constructed using the bruising mean score of the observations. The algorithm requires as input the number of latent components to compute as well as the number of features to retain for each latent component. It is recommended to use cross-validation in order to set optimal values for both parameters; however this option is not available if the dataset contains missing values, which is the case for the present genomics dataset. This does not have too much impact for this step of the analysis as the goal is to reduce the size of the datasets by removing features (i.e. genomic variants, transcribed genes or compounds) with no association with the phenotype, rather than selecting only relevant features. To this end, for the genomics and transcriptomics dataset, two latent components were constructed, each retaining 1,000 features from the corresponding dataset. For the metabolomics dataset, three latent components were constructed, each retaining 200 compounds. These values were chosen to reduce the size of each dataset while retaining enough features to ensure that no feature of interest is discarded. Only the features retained for the latent components of each dataset were kept for further analysis. In the genomics dataset, variants that were identified as high-scoring markers in the GWAS analysis were also retained for further analysis; and in the transcriptomics and metabolomics datasets, features detected as differentially expressed in the single-omics analysis were also retained. This lead to retaining 2,106 genomic variants, 1,985 transcribed genes and 601 compounds.

Next, a Partial Least Squares regression (Wold, 1966) (PLS - implemented in the `mixOmics` package) was used to assess the covariance between pairs of omics datasets (box FS3 in Figure 5.1). The algorithm seeks linear combinations of features from each of the two datasets that maximise the covariance between the datasets. The covariance between the datasets is then estimated as the correlation between the first latent component computed for each dataset.

Finally, the DIABLO algorithm (Singh et al., 2016), also termed block sPLS-DA and implemented in the `mixOmics` package, was used to integrate the three omics datasets together, with respect to the phenotype groups (box FS3 in Figure 5.1). Similarly to sPLS-DA, DIABLO performs both feature selection and dimension reduction on each dataset. The algorithm constructs for each dataset successive latent components, which are linear combinations of subsets of the original features, such that the latent components (i) best discriminate the outcome groups (here the bruising groups) and (ii) maximise the covariance between datasets (i.e. retain features across the datasets that co-vary). The relative importance of these two objectives is set with a design matrix. In the design matrix, each row and column represents one of the input datasets, and the value in a given cell (outside the diagonal) indicates whether the emphasis should be put on maximising the correlation between the two corresponding datasets (value close to one) or differentiating between the outcome groups (value close to zero). For this analysis, the non-diagonal elements in the design matrix were set to 0.1, in what is referred to as a weighted full design matrix. As only a subset of all features from each dataset are retained to construct the latent components, DIABLO only select features that are associated with the phenotype of interest and are correlated. The samples are then projected for each dataset onto the reduced space generated by the latent components. In addition, a weighted consensus reduced space was constructed from these latent components. The coordinate of a given sample in the $i^{th}$ dimension of this weighted consensus reduced space corresponds to the weighted average of its coordinates for the $i^{th}$ latent component of each dataset. The contribution of each dataset is weighted by the correlation between the corresponding latent component and the outcome groups, in order to give more weight to datasets that better discriminate the outcome groups. In this weighted consensus reduced space, a silhouette score was calculated for each sample, which measures how close the sample is located to samples from the same outcome group. A sample is attributed a silhouette score close to one if it is located close to samples from the same group, close to zero if it is located between samples from different groups, and close to -1 if it is located among samples from another group. The average silhouette of a group of samples therefore informs about whether the samples from the group are closely clustered together in the reduced space.

The DIABLO algorithm requires as input the number of latent components to construct for each dataset, as well as the number of features from each dataset to retain for each of the latent components. Again, it is advised to use a cross-validation scheme to tune these values, however it is not applicable if the datasets contain missing values, which is the case for this genomics dataset. Therefore, the number of latent components to be computed was set to two, which corresponds to the number of outcome groups among the observations, and is the recommended value by the authors of the package. In order to estimate the optimal number of features to retain for the first latent component of each dataset, DIABLO was ran with several configurations. The selection of 1%, 3%, 5%, 10%, 15%, 20% or 100% of the features independently in each dataset for the first latent component was tested, while retaining all features for the second latent component of each dataset. This yielded 343 ($7^3$) different configurations. Then, the average silhouette of both bruising groups in the resulting weighted consensus reduced space was computed for each configuration . An appropriate number of features to retain for the first latent component of each dataset was retained based on the resulting group average silhouettes, as well as considering the need to retain a small number of features for later analyses. This implied selecting among the configurations retaining at most 60 features per dataset the one yielding the highest average silhouette for both phenotype groups. In addition, features were ranked in each configuration, independently for each dataset, according to the absolute value of their loading for the first latent component. The loading of a given feature for a latent component, i.e. its coefficient in the linear combination, informs about the contribution of the feature to the latent component. The rank one was attributed to the feature with the highest absolute loading. Features not selected in a given configuration were assigned the rank $p$ ($p$ being the total number of features in the dataset). A mean rank score was then computed for each feature as the average rank of the feature across all configurations. With the number of features to retain for the first component of each dataset set, DIABLO was ran with several percentages of features to retain for the second latent component (again 1%, 3%, 5%, 10%, 15%, 20% or 100% of the features in the dataset), and evaluated the different configurations based on the resulting groups silhouette average. The mean rank score of each feature was also computed for the second latent component, as described previously, but using the features loading for the second latent component. Ultimately, this led to retaining for each latent component 21 variants, 59 transcribed genes and 60 compounds. This analysis will be referred to thereafter as the full DIABLO analysis. In addition, the analysis was repeated while restricting the genomics dataset to only the variants detected as high-scoring markers in the GWAS analysis (GWAS-only DIABLO analysis). In this case, 23 variants, 59 transcribed genes and 30 compounds were retained for the first latent component, and for the second latent component three variants, 59 transcribed genes and six compounds (i.e. 180 features in total).

### 5.3.6 Causal inference

The next step of the analysis is to perform causal inference on the features selected with DIABLO. The 74 samples constituting the low and high bruising groups were retained. Each individual omics dataset restricted to the features selected with DIABLO was centred and scaled, before combining them to obtain a multi-omics matrix with 180 features measured across 74 observations (box CI1 in Figure 5.1). In addition, the centred and scaled bruising mean score of each observation was added to the multi-omics matrix as an additional variable to account for in the causal inference. Twelve missing values for genomic variants were replaced by zero (i.e. the mean across the dataset as the data was centred). Seven different causal inference methods along with two network inference methods were tested (box CI3 in Figure 5.1):

- PC-stable (Colombo & Maathuis, 2014) – constraint-based causal inference method (`bnlearn` package – Nagarajan et al. (2013));
- FCI (Spirtes et al., 1999) – constraint-based causal inference method (`pcalg` package – (Kalisch et al., 2012));
- FCI+ (Claassen et al., 2013) – constraint-based causal inference method (`pcalg` package);
- GES (Chickering, 2003) – score-based causal inference method (`pcalg` package);
- FGES (Ramsey et al., 2017) – score-based causal inference method (`rcausal` package – (Wongchokprasitti, 2019));
- MMHC (Tsamardinos et al., 2006) – hybrid causal inference method (`bnlearn` package);
- ARGES (Nandy et al., 2018) – hybrid causal inference method (`pcalg` package);

- ARACNE (Margolin et al., 2006) – network inference method (`minet` package – (Meyer et al., 2008));
- GENIE3 (Huynh-Thu et al., 2010) – network inference method (`GENIE3` package – (Huynh-Thu et al., 2010)).

Each method is described in more detail in Section 3.2.3. Contrary to the analyses performed in Chapter 3, the version of the PC algorithm implemented in the `bnlearn` package was used, and this implementation is referred to thereafter as PC-stable. Simple comparisons highlighted that the implementations from the `pcalg` package and the `bnlearn` package return inferred graphs with identical skeletons, but differ in the orientation of problematic v-structures. This could be solved with appropriate setting of the parameters for the `pcalg` implementation, but was not attempted for this analysis. Each causal or network inference method depends on one or more tuning parameters, whose value must be set by the user. Again, these tuning parameters and their meaning are presented in Section 3.2.5. Briefly, I focus on the three following tuning parameters:

- $\alpha$: the constraint-based causal inference methods rely on conditional independence tests to reconstruct the causal graph. They consider two variables to be independent conditionally on a set of other variables if the p-value of the corresponding conditional independence test is lower than the threshold $\alpha$. Consequently, smaller values of $\alpha$ yield sparser inferred graphs.

- Penalty: the score-based methods assess the fit of a candidate causal graph to the observed data using some model selection criterion. The latter balances the fit of the model to the data with the complexity of the model, i.e. the number of parameters. The penalty terms dictates the impact of the model complexity in the resulting graph score. Higher penalty values highly penalise model complexity, i.e. the number of edges in the candidate graph, and thus yield sparser inferred graphs.

- Threshold quantile: the two investigated network inference methods return a weighted adjacency matrix that indicates the confidence in the existence of an edge between any two features. In order to obtain a non weighted inferred graph, a threshold is required, and any edge with a weight below this threshold is removed from the final graph. In this experiment, the threshold is selected as a quantile of the observed weight distribution in the inferred weighted adjacency matrix. This way, a threshold quantile of 0.5 signifies that the top 50% of the edges (i.e. with the highest weights) are retained in the final graph.

Each method was run for different values of the tuning parameters. For each method, the value of the tuning parameter(s) yielding a number of inferred edges (regardless of their orientation) as close to 150 as possible was selected, as across the range of parameter values tested, each method was able to infer a graph with approximately 150 edges (see Figure 5.8). A bootstrapping scheme was used to quantify the confidence in the inferred causal relationships. One hundred bootstrap datasets were generated from the original multi-omics matrix, each generated by sampling with replacement 74 observations (i.e. the number of observations in the original dataset) from the original matrix (box CI2 in Figure 5.1). The different causal and network inference methods were applied to each of the 100 bootstrap datasets, with the selected values of the tuning parameters. For each method, the skeletons of the graphs inferred from the bootstrap datasets were extracted, and a confidence score was computed for each (undirected) edge as the fraction of bootstrap datasets for which the edge was also inferred (box CI4 in Figure 5.1).

The similarity between the skeletons of the graphs inferred by each method with the original dataset was quantified as follows. Given an inferred graph $\mathcal{G}_A$ with $N$ nodes, I define its skeleton adjacency matrix $S_A = \{S^A_{ij}\}_{1 \leq i,j \leq N}$ with $S^A_{ij} = S^A_{ji} = 1$ if there is an edge between the nodes $i$ and $j$ in $\mathcal{G}_A$, and 0 otherwise. Thus, for two graphs $\mathcal{G}_A$ and $\mathcal{G}_B$, both with the same $N$ nodes, the similarity between their skeleton is computed as:

$$Sk = 1 - \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} |S^A_{ij} - S^B_{ij}|}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} S^A_{ij} + \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} S^B_{ij}}$$

Additionally, the orientation of inferred causal relationships was compared across methods by answering, for each inferred graph, several causal queries (Heinze-Deml et al., 2018 – box CI4 in Figure 5.1). The queries considered assess the ability of the methods to detect different types of causal relationships between the features. In particular, each inferred graph was used to answer the following queries for any pair of features $(A, B)$:

- the parent query: is $A$ a causal parent of $B$? The answer is “yes” if there is a direct causal link from $A$ to $B$. As the FCI and FCI+ algorithms only infer ancestral relationships, they cannot answer this causal query. Moreover ARACNe, which returns undirected graphs, cannot answer this query either.

- the potential parent query: is $A$ a potential causal parent of $B$? The answer is “yes” if there is a direct causal link from $A$ to $B$ or if the orientation of the causal edge between $A$ and $B$ could not be determined. Parents of a given variable are also considered as potential parents of the variable.

- the ancestor query: is $A$ an ancestor of $B$? The answer is “yes” if there is a direct or indirect causal path directed from $A$ to $B$. ARACNe, returning undirected graphs, cannot answer this query.

- The potential ancestor query: is $A$ a potential ancestor of $B$? The answer is “yes” if there is a direct or indirect causal path from $A$ to $B$, possibly including causal edges for which the orientation could not be inferred.

For a given inferred graph $\mathcal{G}$, the answer to a particular causal query comes in the form of a $p \times p$ matrix $Q^{\mathcal{G}}$, where $p$ is the number of features in the dataset, with element $Q^{\mathcal{G}}_{i,j} = 1$ $(1 \leq i, j \leq p)$ if the answer is positive for the (ordered) pair of variables $(i, j)$ and 0 if the answer is negative. Similarly to the edges of the inferred graphs skeleton, a confidence score can be computed for each positive answer for a given method as the fraction of bootstrap datasets for which the answer was also positive with the corresponding method.

Lastly, the impact of removing edges directed towards genomic variants on the reconstructed causal graphs was also evaluated (box CI4 in Figure 5.1). This can be done by providing to the causal or network inference method a blacklist of directed edges to be excluded from the reconstruction process. Only some implementations of the investigated methods can process such a blacklist, namely PC-stable, FGES, MMHC and GENIE3. The causal inference task was therefore repeated with these four methods (using for their tuning parameters the values selected before) on the original dataset as well as on the bootstrap datasets, and providing as a blacklist the list of all possible edges pointing towards any of the genomic variants. This prevents the method from inferring edges that are directed towards a genomic variant.

## 5.4 Results and Discussion

### 5.4.1 Metabolomics

#### Exploratory analysis

Untargeted metabolomic profiling was performed on potato tubers previously subjected to mechanical bruising. This yielded measurements of 4,604 compounds in total (2,570 from the Cp mode, 953 from the Cn mode, 853 from the Hp mode and 228 from the Hp mode), across two biological replicates of 122 progeny samples (i.e. 244 observations). A strong positive correlation was observed between the variance of the compounds' area intensities and their mean. Therefore, the area intensities were normalised using a Variance Stabilising Transformation, which is similar to the transformation commonly used to correct the mean-variance bias in transcriptomics datasets. Samples correlation and clustering revealed an influence of the samples processing order on the profile of some compounds, especially for the Cn and Hp datasets. This is most likely due to shifts in retention time occurring throughout the analysis. This bias was however minimised by a correction of retention time differences between compounds during the data pre-processing, which corrected most of the variation due to processing order. The t-distributed Stochastic Neighbouring Embedding (tSNE) algorithm was applied to each dataset separately as well as to the combined dataset, in order to observe the relationships between samples. The result for the combined dataset is presented in Figure 5.2 (left panel). While for some samples both biological replicates are closely located on the reduced space, in some cases

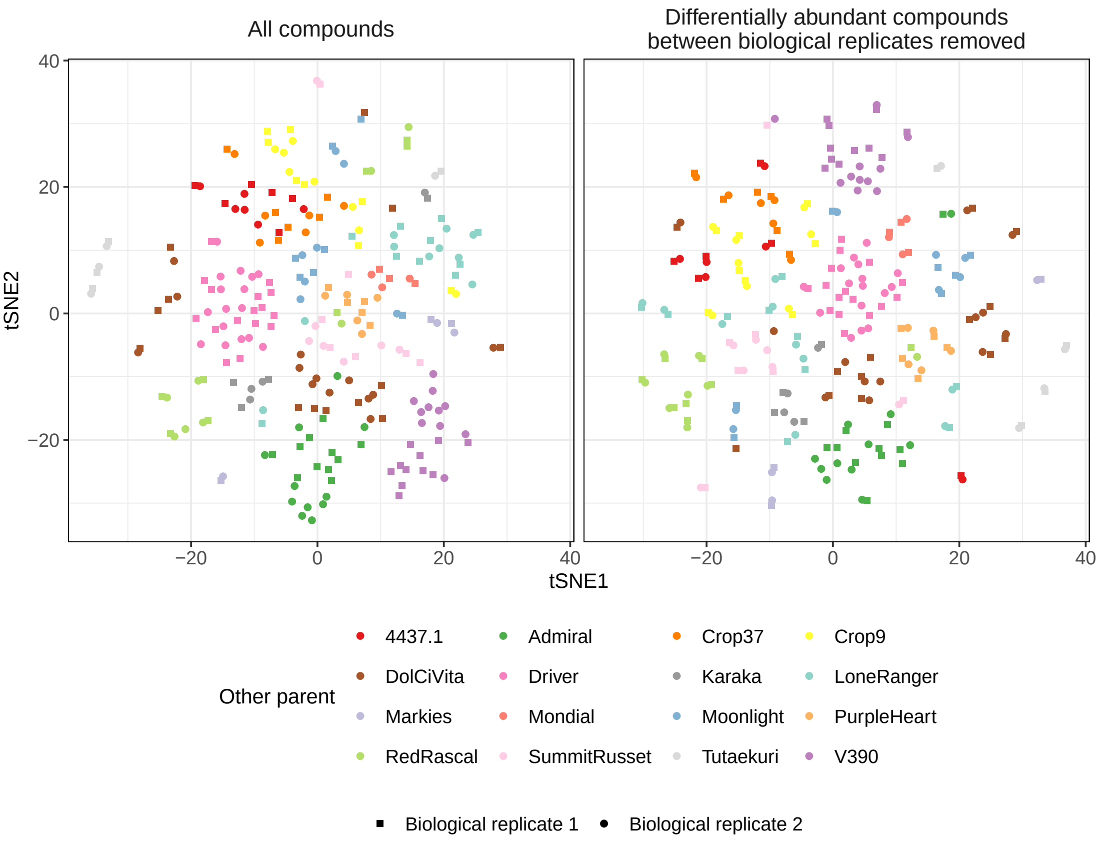


Figure 5.2: Coordinates of the samples in the tSNE reduced space obtained on the combined metabolomics dataset with all compounds (left panel) or with only compounds that were not found to be differentially expressed between the two biological replicates (right panel). Each point represents one observation, its shape describing whether it is labelled as the first or second biological replicate of the sample, and its colour representing the parent (other than Crop52) of the sample.

they lie far apart. This could be caused by the order in which the samples were processed: the first biological replicate of all samples was processed first, and the second biological replicates was processed in a second batch. Alternately, this difference could have arisen from a difference in the environment in which the two biological replicates were grown. A differential analysis was therefore performed to compare the intensities of the compounds between the first and second biological replicates of all samples. From the combined dataset, 557 compounds (12.1%) were found to be differentially abundant, mainly compounds identified with the Hp (39%) and Cp (32.9%), and flagged for later analyses. The identified differentially abundant compounds included a number of amino acids, several steroidal alkaloids and glycoalkaloids found notably in potatoes, and ascorbic acid, amongst others. In a number of cases, two different compounds found to be differentially abundant were identified as the same metabolite, but their retention time was different. This confirmed that differences between the biological replicates of a same samples arose in part from their processing order. Consequently, the list of compounds found to be differentially expressed was recorded for comparison with subsequent results. The tSNE analysis was repeated without these differentially abundant compounds for comparison, and the result for the combined dataset is presented in Figure 5.2 (right panel). As expected, most progeny samples arising from a same cross were located close, even though some outliers can be observed, which might be due to a mislabelling of the samples, or to differences in growing conditions.

### Differential expression

A differential analysis was performed to compare the compounds intensities between two groups of 97 observations each (where an observation corresponds to one biological replicate of a progeny sample) with low and high bruising scores respectively. In order to detect compounds with small but consistent shifts between the two phenotype groups, no threshold was used on the fold-change of the compounds when assessing significance. From the combined dataset, 107 compounds were found to be differentially abundant (63 downregulated, 44 upregulated). The results of the differential analysis on the combined dataset were in good agreement with those from the analysis performed on each dataset separately. Moreover, only 17 of these differentially abundant compounds were also found to be differentially abundant when comparing the two groups of biological replicates. This is reassuring, as it means that the differences observed between the two phenotype groups are not due to the processing order of the samples or difference in growing conditions. Amongst the differentially abundant compounds, identified compounds include derivatives of spermine and spermidine, two polyamines involved in tuber stress response (Lulai et al., 2015) and notably mandarin bruising (Lamikanra et al., 2005), the reduced form of L-glutathione, an important antioxidant with multiple roles in stress response, citrulline (a precursor of arginine), nucleobases such as uracil and guanine, as well as guanosine, fatty acids such as corchorifatty acid F (which plays an antioxidant role in plants) or 10,12-9-hydroxy-10,12-octadecadienoic acid, both involved in linoleic acid metabolism. These results are consistent with an activation of stress-response pathways in the tubers following mechanical damage.

### Co-abundance network

Next, the `WGCNA` package was used to reconstruct a co-abundance network between the compounds. Among the reconstructed network obtained with the combined dataset, 47 modules of highly co-abundant compounds were detected, ranging in size from 10 to 1,108 compounds. Interestingly, all compounds were assigned to a module. This stands in contrast with the co-expression network reconstructed from the transcriptomics dataset, for which 44.2% of the transcribed genes were not assigned to a module. For each module, an eigencompound was computed, which provides a summary of the intensity profile of the compounds within the module for each observation. For six of the 47 identified modules (12%), their eigencompound was found significantly correlated to the bruising

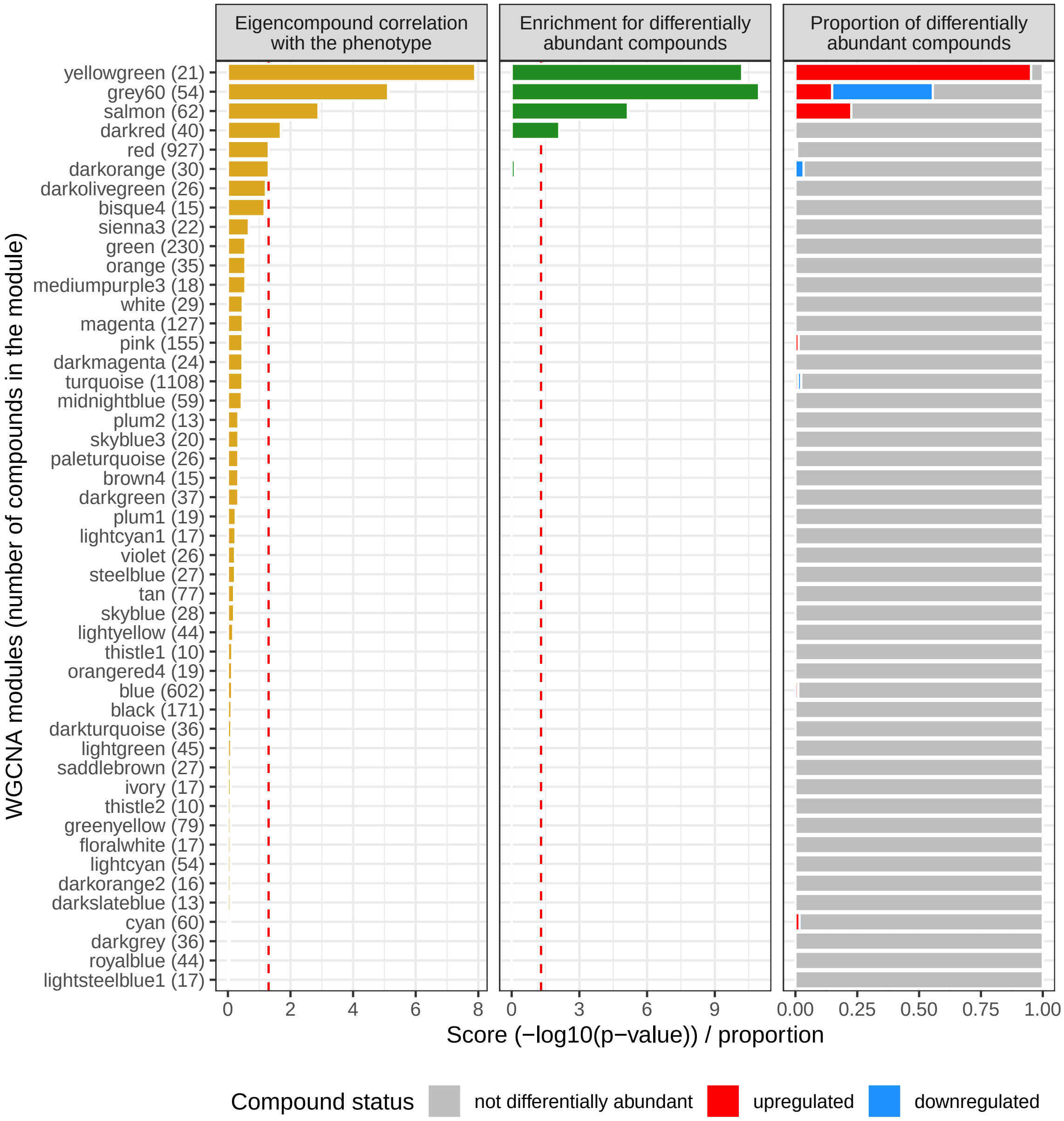


Figure 5.3: Association of the WGCNA modules with the bruising mean score. Left panel: adjusted correlation score (i.e. -log10(p-value)) between the modules' eigencompound and the bruising mean score. Middle panel: adjusted GSEA enrichment score of the modules for differentially abundant compounds. Right panel: proportion of upregulated (red), downregulated (blue) and not differentially abundant (grey) compounds in each module. For the left and middle panels, the dotted red lines represent the 0.05 p-value significance threshold.

mean score of the observations (see Figure 5.3). Among them, four were detected as significantly enriched in differentially abundant compounds, including one module (WGCNA module “darkred” in Figure 5.3) with no differentially expressed compounds. In the latter case, the significant enrichment is due to the fact that a majority of the compounds in the module have differential abundance scores close to the significance threshold. The differentially abundant compounds were clustered into nine modules in total. Again, this differs from the results from the transcriptomics co-expression network reconstruction, in which the differentially expressed genes were found scattered across many modules. This could indicate that with metabolomics data, WGCNA is able to gather metabolites involved in a common metabolic pathway in a same module, and the differential activation of the pathway between the phenotype groups can be visualised as most compounds involved are found to be differentially abundant.

As an example, two modules identified are presented in Figure 5.4. In this figure, the yellow-green module includes compounds that are related to the linoleic acid metabolism, for example corchorifatty acid F and 9,10,13-tihydroxy-11-octadecenoic acid. The grey-60 module clusters compounds involved in glutathione biosynthesis, notably parent compounds of spermine and spermidine. It also includes four distinct compounds all identified as L-glutathione. This again reflects the large shift in retention time between samples, as all four compounds have the same molecular weight (307.08 g/mol) and identified formula, but their retention time ranges from 1.0 to 1.16 minutes. These four compounds are assigned a high adjacency score with WGCNA. Note however that two derivatives of putrescine (feruloylputrescine and caffeoylputrescine), also involved in glutathione metabolism as precursors of spermidine, are assigned to a different module. Seven additional modules were investigated to see if they gathered compounds involved in common metabolic pathways. Over these seven modules, three did not contain any identified compounds. In the remaining four, most of the identified compounds could be linked to a common metabolic pathway.

### 5.4.2 Multi-omics data integration with DIABLO

#### The DIABLO algorithm

In order to gain a better understanding of the molecular mechanisms underlying potato tuber bruising, I aimed at integrating measurements obtained at different omics levels (namely genomics, transcriptomics and metabolomics). The first step involved extracting from these omics datasets features (i.e. genomic variants, transcribed genes and metabolites) that (i) are involved in the biological processes of tuber bruising and (ii) interact with one another. Ninety-eight samples with measurements from all three omics datasets as well as with a measured bruising mean score were used for this purpose (one biological replicate per sample). From them, 41 samples with a bruising mean score of one or less were used to create the low bruising group, while 33 samples with a bruising mean score of two or more were retained as the high bruising group. The remaining samples were removed from

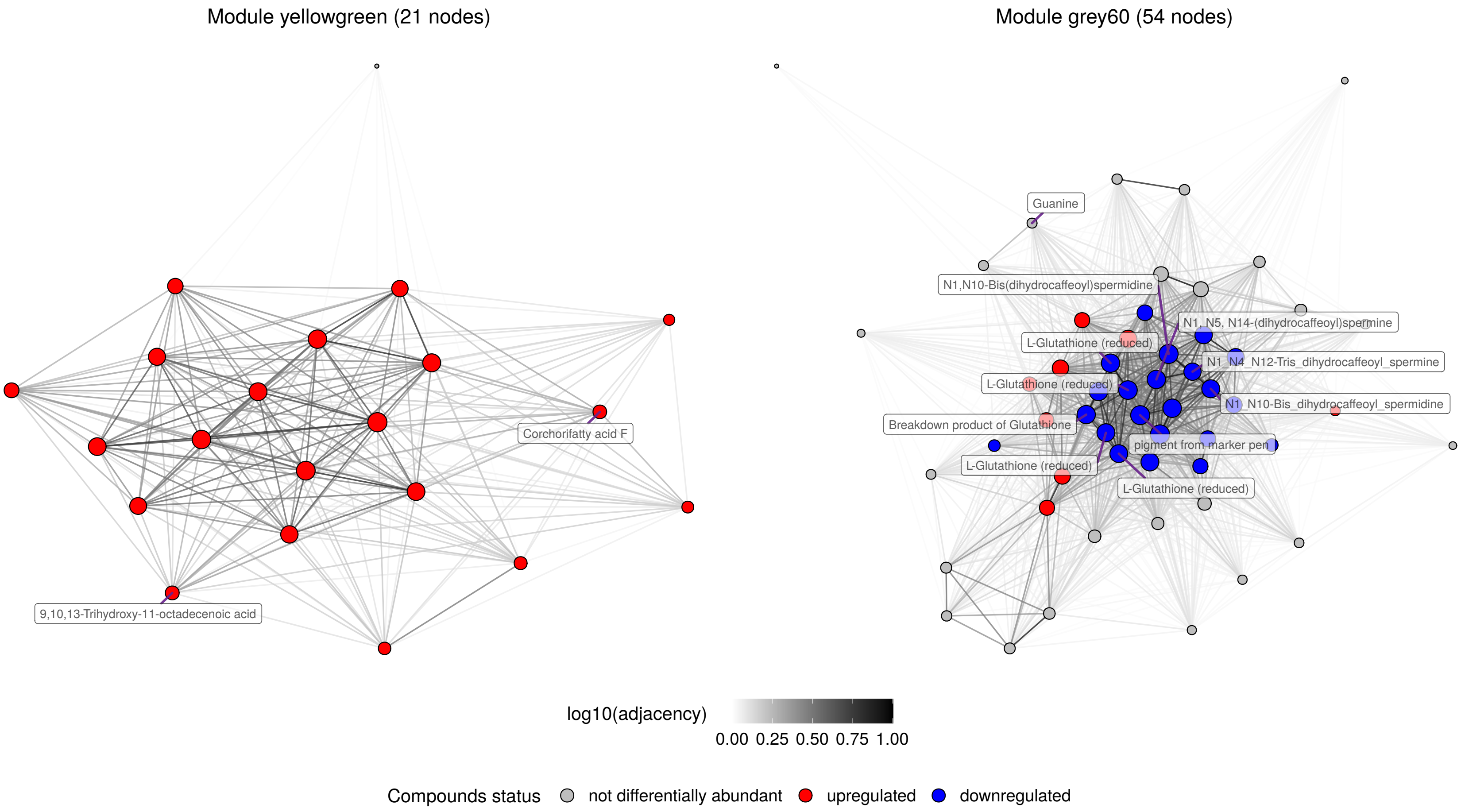

Module yellowgreen (21 nodes)
Module grey60 (54 nodes)
Corchorifatty acid F
9,10,13-Trihydroxy-11-octadecenoic acid
Guanine
N1,N10-Bis(dihydrocaffeoyl)spermidine
N1, N5, N14-(dihydrocaffeoyl)spermine
L-Glutathione (reduced)
N1_N4_N12-Tris_dihydrocaffeoyl_spermine
L-Glutathione (reduced)
N1_N10-Bis_dihydrocaffeoyl_spermidine
Breakdown product of Glutathione
pigment from marker pen
L-Glutathione (reduced)
L-Glutathione (reduced)
log10(adjacency)
0.00 0.25 0.50 0.75 1.00
Compounds status
not differentially abundant
upregulated
downregulated

Figure 5.4: Two example modules from the metabolomics co-abundance network. The nodes in each module represent compounds, and the edges between them represent the adjacency between pairs of compounds. The colour of the edges indicates the adjacency score attributed to each pair of compounds, with values below 0.01 set to zero for visualisation sake. The size of the nodes shows the weighted degree of the nodes, and their colour the status of the compound in the differential analysis (grey if not differentially abundant, red if upregulated and blue if downregulated). Identified compounds are labelled.

the analysis. Initially, a coarse feature pre-selection was performed independently for each dataset, in order to reduce the datasets dimension by removing features with no association to the phenotype. Using a $l_1$-penalised Partial Least Square regression, 2,000 genomic variants, 2,000 transcribed genes and 600 compounds were retained as potentially associated with the phenotype. In addition, features detected as associated with the phenotype in the single-omics analyses (i.e. high-scoring markers from the GWAS analysis, and significantly differentially expressed genes and differentially abundant compounds) were also retained. yielding a total of 2,106 genomic variants, 1,985 transcribed genes (as 38 transcribed genes were selected for both latent components) and 601 compounds.

Then, the DIABLO algorithm was used to select only features that best discriminated the two phenotype groups, and that co-vary. DIABLO creates for each dataset a number of latent components (here, two), which are linear combinations of the features that best separate the phenotype groups, and such that the covariance between the $i^{th}$ latent component of each dataset is maximised. Note that the different latent components are not explicitly constructed to be orthogonal. In order to perform feature selection, only a subset of all features is used to create any given latent component, by setting the loading of remaining features (i.e. their contribution to the latent component) to zero, using a $l_1$ penalisation. For this analysis, the feature selection was performed with the additional goal of retaining a reasonable number of features for the causal inference step, as causal inference can become computationally intensive when applied to a large set of variables. For this, it is possible to tune the degree to which the covariance between the latent components is maximised, in the form of a design matrix. The design matrix informs about the balance between the two goals of DIABLO: discriminating the outcome groups (off-diagonal elements of the matrix close to 0) and maximising the covariance between the datasets (off-diagonal elements of the matrix close to 1). This choice can be guided by performing a pairwise comparison of the omics dataset. Using a partial least square analysis on each pair of datasets, I found a moderate to high correlation between the three pairs of datasets (from 0.61 between the transcriptomics and metabolomics dataset to 0.72 between the genomics and metabolomics dataset). This justified the use of a weighted full design matrix (i.e. with off-diagonal elements of the matrix set to 0.1), that strikes a balance between the two goals.

#### Features selection with DIABLO

It is recommended to use cross-validation to inform the choice of number of features to be selected for each of the omics dataset. This was however not an option in this case due to the presence of missing values in the genomics dataset. Instead, I compared how selecting different numbers of features from each dataset for both latent components affected the clustering of samples from each phenotype group in the resulting reduced space (see Materials and Methods). For the first latent component, retaining a small number of variants and a large number of transcribed genes yielded the best average silhouette (quantifying how closely the samples from a same group are located in the reduced space) for both groups. The number of metabolites retained did not impact much the resulting average silhouettes. For the second latent component however, a large number of variants and a smaller number of transcribed genes resulted in better clustering of the two groups. Again, the number of metabolites included did not affect much the clustering of the groups.

For each latent component, the mean rank score of the features was also computed. It quantifies the importance of each feature for the corresponding latent component. A small mean rank score (close to one) indicates that the feature is consistently attributed a high absolute loading for the latent component, while a high rank score indicates that the feature is consistently not selected for the latent component. As presented in Figure 5.5, the mean rank score of transcribed genes and compounds for the first latent component is correlated with the results of the differential analysis performed on each dataset. Most of the features with small mean rank scores were found to be differentially expressed, especially in the case of metabolic compounds. This is not surprising as we can expect features that are differentially expressed between the two groups to be able to efficiently discriminate the two sample groups. On the contrary, the second latent component selected mostly features that are not differentially expressed. Again, this makes sense, as we expect the different latent components to uncover uncorrelated trends in features levels that differentiate the phenotype groups. Thus, if the first latent component retains differentially expressed features, the second latent component will seek features less correlated with the bruising phenotype but that also play a role in differentiating the groups. Contrary to what is observed with the transcriptomics and metabolomics datasets, there seems to be little correlation between the mean rank score of the genomic variants, for either the first or the second latent component, and the results of the GWAS analysis. Indeed, the high-scoring markers did not obtain a smaller mean rank than variants that were not high-scoring in the GWAS analysis. On the contrary, they were found in the middle of the list. This phenomenon is most likely due to the impact of population structure, with markers associated with the phenotype only through population structure rather than true causal effect obtaining lower mean rank. Indeed, the variants mean rank scores seem to be correlated to some extent to the variants scores obtained when performing the GWAS analysis without correcting for the effect of population structure. It comes as no surprise, since no information about population structure was included in the DIABLO analysis. I therefore expect DIABLO to be subject to the same biases as the GWAS analysis, including spurious associations between variants and phenotype due to stratification amongst samples.

Ultimately, the number of features to be used for constructing each latent component was set to 21 variants, 59 transcribed genes and 60 compounds, a configuration that yielded good phenotype groups clustering (Figure 5.6) while retaining a reasonable number of features in each dataset. This result is thereafter referred to as the full DIABLO analysis. However, two potential problems were noted regarding the selected variants. First, as mentioned above, the association between the selected variants, and especially those retained for the first latent component, and the phenotype groups likely arose from the population structure among the samples. The score of these variants was

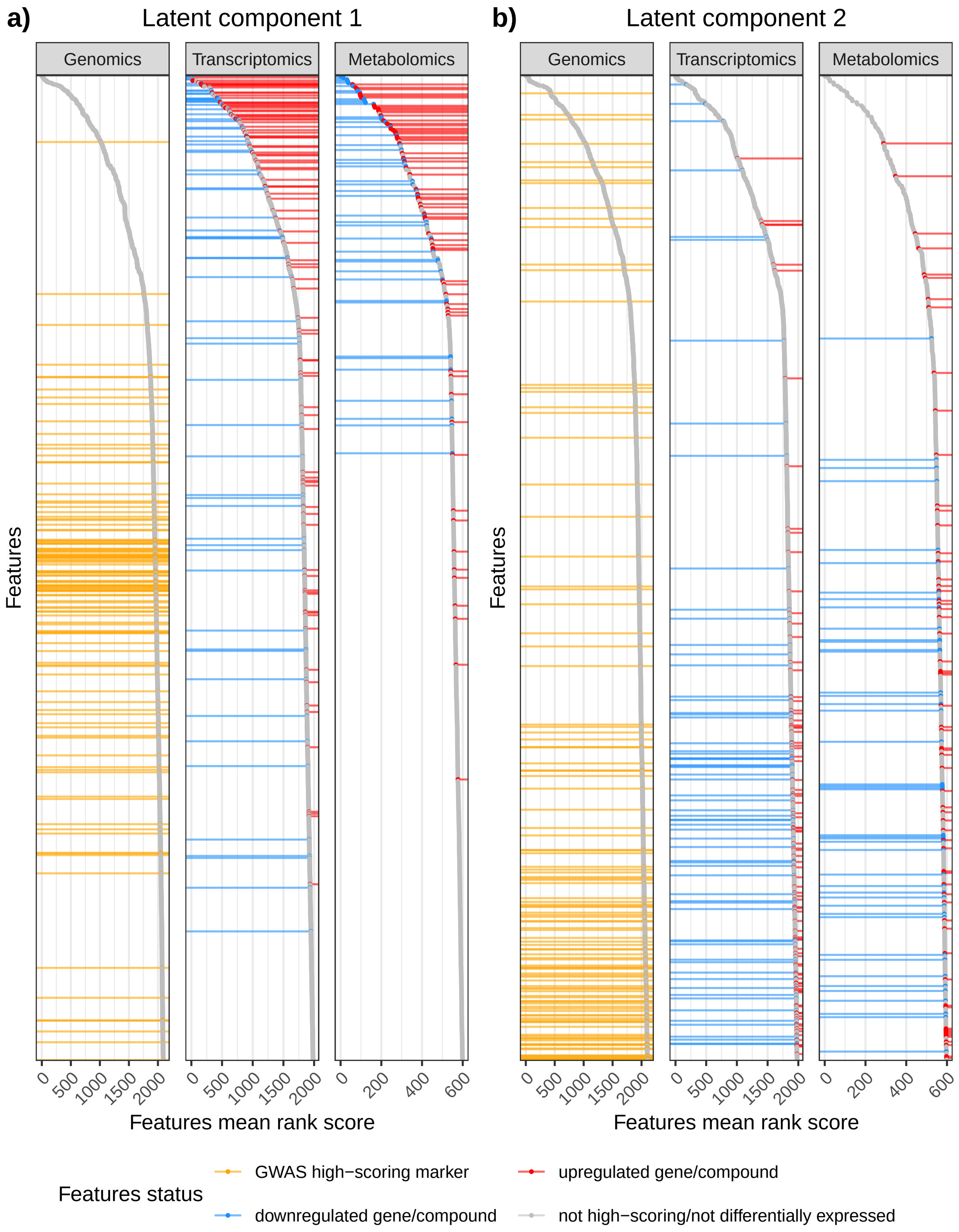


Figure 5.5: Features mean rank score for each dataset a) for the first latent component and b) for the second latent component. High-scoring markers from the GWAS analysis are coloured in orange, while upregulated transcribed genes and compounds are drawn in red and the downregulated transcribed genes or compounds in blue. Horizontal lines highlight the position of these interesting features.

higher with a GWAS analysis that ignored samples stratification, than with an analysis in which population structure effects were accounted for (Supplementary Figure G.1). The impact of this can be visualised in the weighted consensus reduced space when featuring the crosses from which the samples arose (Supplementary Figure G.2). In addition to separating the two phenotype groups, the reduced space also clusters progeny from crosses involving LoneRanger and V390 away from other samples, highlighting the population structure amongst the samples. Moreover, 13 (out of 21) variants that were selected for the first latent component, were located between 54.3Mb and 54.8Mb on chromosome 8, while 19 (out of 21) variants that were retained for the second latent component were positioned between 84.2Mb and 86Mb on chromosome 1. The closely co-localised variants were also highly correlated, most certainly due to linkage disequilibrium. This implies that the importance of these genomic regions in discriminating the sample groups is exaggerated, as the effect of the correlated variants are summed in the latent component. Thus, even though several variants are retained from these regions, they likely reflect only one source of variation amongst the samples. Therefore, in order to correct for this bias and the fact that the selected features seem to arise from population structure, the DIABLO analysis was repeated by restricting the genomics dataset to only those variants detected as high-scoring markers with the GWAS analysis (GWAS-only DIABLO). This filtering should increase the power to detect features of interest, by limiting the effect of population structure bias. This time, the final configuration involved the selection 23 variants, 59 transcribed genes and 30 metabolites for the first latent component, and three variants, 59 transcribed genes and six compounds for the second latent component. The resulting consensus reduced space is shown in Figure 5.6. The genomic position of the selected variants with both the full and GWAS-only analyses are presented in Supplementary Figure G.3.

### Retained features

Interestingly, restraining the genomics dataset to only the high-scoring markers did not influence much the features retained from the transcriptomics and metabolomics datasets. Indeed, 88 out of the 118 (74.6%) transcribed genes and all 36 compounds selected in the GWAS-only DIABLO were also retained in the full DIABLO analysis. A number of Gene Ontology (GO) terms were associated with the selected transcribed genes, including protein phosphorylation (eight transcribed genes), regulation of transcription (six transcribed genes), oxidation-reduction process (six transcribed genes), primary metabolic process (five transcribed genes) or response to stress (two transcribed genes). From the metabolomics dataset, a large fraction of the selected compounds could not be identified. However, several compounds involved in glutathione metabolism were found, including the four compounds identified as L-glutathione in its reduced form, dihydrocaffeoyl spermine and bis-dihydrocaffeoyl spermidine. Other identified compounds are linked to the linoleic or $\alpha$ linoleic acid pathway: 10,12-9-hydroxy-10,12-octadecadienoic acid, 9-hydroxy-10,12,15-octadecatrienoic acid (9-HOTrE) and 9,10,13-trihydroxy-11-octadecenoic acid (Kimuta & Yokota, 2004). Several other compounds were recognised as breakdown products of metabolites, and could have been

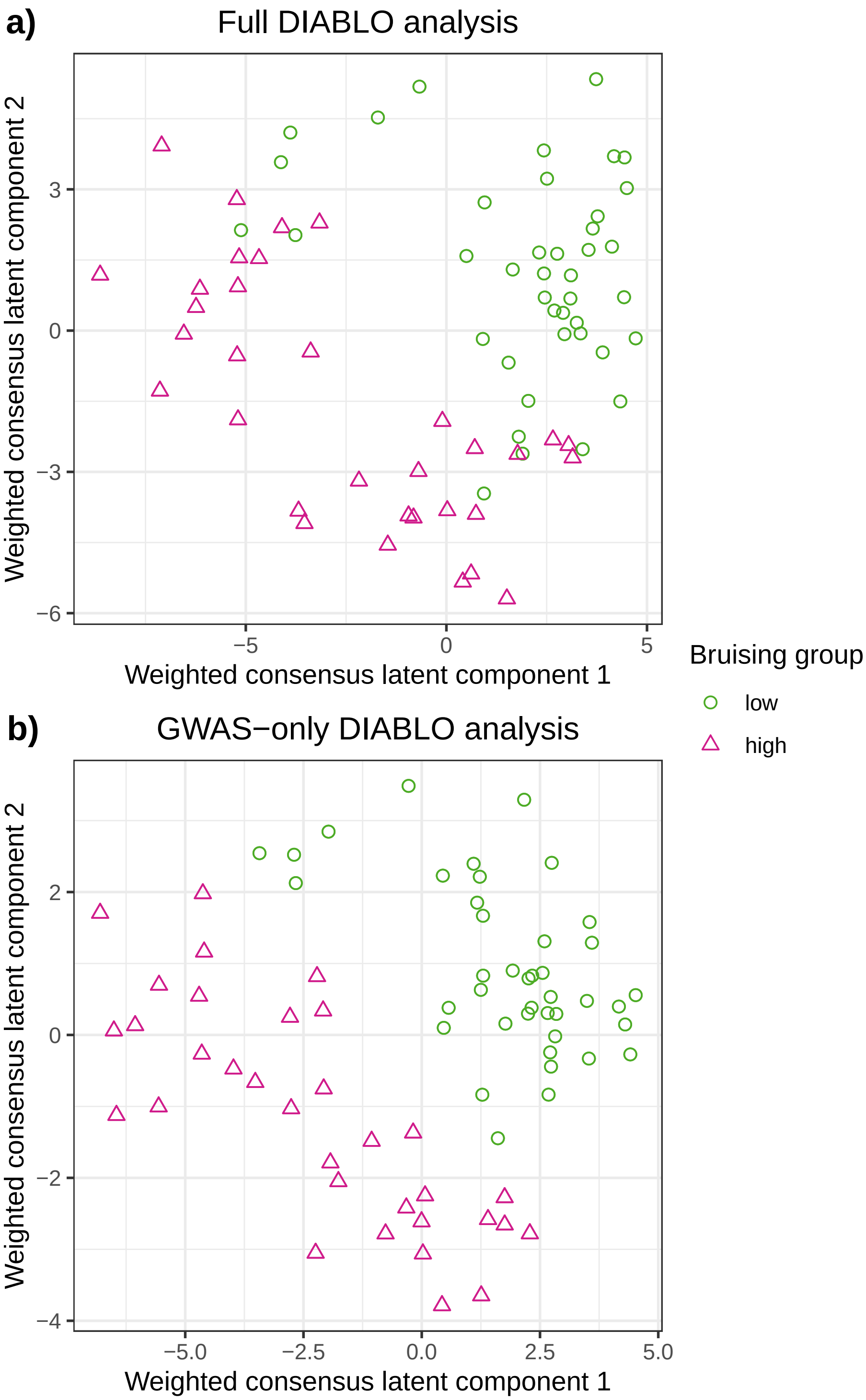


Figure 5.6: Weighted consensus reduced space of a) the full DIABLO analysis and b) the GWAS-only DIABLO analysis. Each point represents one observation (i.e. a biological replicate of one sample), with their shape and colour indicating whether they belong to the low bruising group (green circles) or the high bruising group (pink triangles).

selected as the original metabolite is associated with the phenotype. Interestingly, a number of transcribed genes and compounds that were either selected with DIABLO or found to be differentially expressed/abundant are involved in common biological processes. For example, in addition to retaining glutathione-related compounds, DIABLO also selected a transcribed gene encoding for the glutathione-S-transferase protein (ID of the transcribed gene in the genome annotation PGSC-DM v4.03: PGCS0003DMG400029626). This protein is involved in cell detoxification by catalysing glutathione conjugation to protect cells from oxidative damage (Hernández Estévez & Rodríguez Hernández, 2020). The glutathione-S-transferase protein concentration was found to be increased by high concentrations of glutathione in *Arabidopsis* mutant lines (Kumar & Chattopadhyay, 2018). Other common biological processes include notably the purine metabolism, pyrimidine metabolism, cysteine and methionine metabolism and sulphur metabolism. Finally, it can be noted that most but not all genomic variants selected are found close to the physical position of transcribed genes retained by DIABLO. This points out to potential cis-mutations affecting these genes' expression. Alternately, it could be due to linkage disequilibrium.

The correlation between the selected features and the latent components of the corresponding dataset is presented in Figure 5.7. This figure highlights sets of genomic variants, transcribed genes and metabolites that co-vary and drive the difference between the two bruising groups. Selected transcribed genes and compounds that were found to be differentially expressed/abundant heavily influence latent component 1, with up- and down-regulated features sitting at opposite ends of the correlation plots, as can be expected. A number of non-differentially expressed genes are clustered with them, pointing out features potentially involved in similar biological processes but that would have been missed by the differential analysis performed on the transcriptomics dataset. On the contrary, the second latent component is mostly driven by non-differentially expressed/abundant features. When assessing the correlation between features from different datasets, many of the highest absolute correlations involve glutathione or related compounds. In particular, a high positive correlation (0.7) was found between one of the compounds identified as L-glutathione and a transcribed gene encoding a heat shock factor (PGCS0003DMG400028414). Heat shock proteins have been found involved in response to stress in plants, and their expression was shown to be modulated by glutathione levels (Kumar & Chattopadhyay, 2018).

### 5.4.3 Causal inference

The next step in multi-omics data integration is to uncover the causal relationships between the features that were found linked to the phenotype of interest. This can be done by applying causal inference methods to the omics dataset. They seek to detect causal relationships among a given set of variables, using observations of the values taken by these variables. One straightforward way to combine the omics datasets when performing causal inference is to treat each feature as a variable in the causal inference task. This is achieved by first centering and scaling each omics dataset

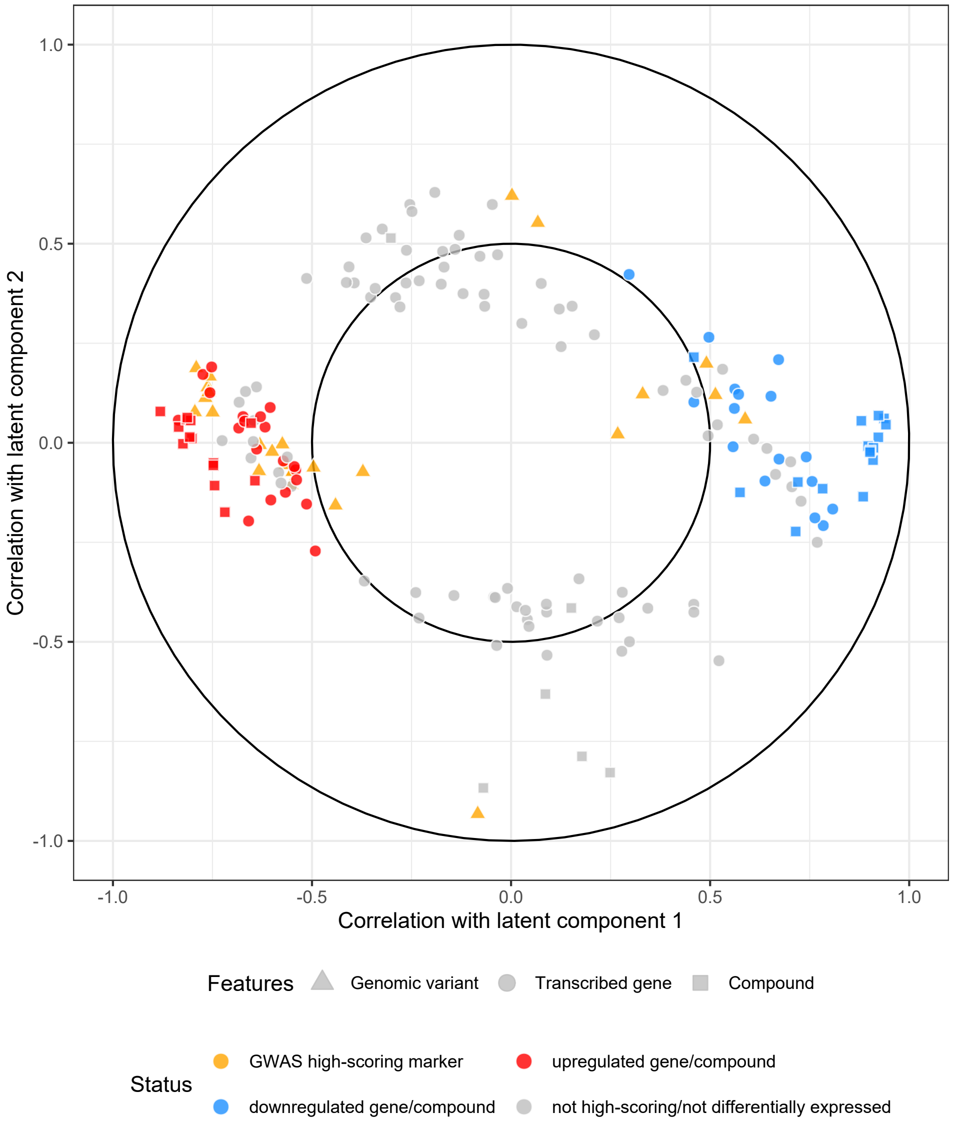


Figure 5.7: Correlation between the features selected with the GWAS-only DIABLO analysis and the latent components of each corresponding dataset. The shape of the points indicates whether the corresponding feature is a genomic variant (triangle), and transcribed gene (circle) or a compound (square). The colour of the points represents the status of the features in the single omics analyses.

independently (restricted to the features selected with DIABLO), and then combining them to obtain one multi-omics matrix describing the intensity of all features across the different observations. As I seek to understand the impact of these features on the phenotype, the bruising mean score of each observation was added to this multi-omics matrix as an additional variable for the causal inference task. Rather that relying on a single statistical method to infer causal relationships between the selected features, the causal graphs inferred by seven state-of-the-art causal methods as well as two widely used network inference methods were compared (see Section 3.2.3). Although the latter do not infer causality but association between molecular features, comparing their inference allows to assess the added benefit of seeking causal relationships rather than association in order to bridge the gap between genotype and phenotype.

One of the first challenges when performing causal inference is to choose a suitable value for the different tuning parameters of the methods. These tuning parameters dictate the sparsity of the resulting inferred causal graph. In order to obtain inferred graphs that are comparable across the investigated methods, a range of values for their tuning parameters were first tested for each method (Table 5.1). The number of edges in the skeleton of the graphs inferred by each method as a function of the values of tuning parameters is presented in Figure 5.8. It ranges from around 30 with a high threshold for ARACNe, to more than 600 for GENIE3 with a low threshold value. For each method, the value of the tuning parameter(s) yielding an inferred graph with a number of edges as close to 150 as possible was selected. This value was chosen as it corresponds to a level of sparsity that can be achieved by all methods within the range of values tested for the different tuning parameters, providing plausible and interpretable results. For the rest of the analysis, the selected values of tuning parameters were used for each method (see Table 5.1). With these settings, the number of edges in the inferred causal graphs ranges from 146 with ARACNe to 155 for PC-stable and FCI+.

### Inferred causal relationships across the methods

As the different causal and network inference methods rely on different computational strategies to extract relationships between features, I expect the inferred causal graphs to not be all identical. I however expect the methods to detect significant true signal from the data. I focus in the first instance in comparing the skeleton of the graphs inferred by the different methods, i.e. ignoring the orientation of the causal relationships detected by the methods. The skeleton similarity score between

Table 5.1: For each causal or network inference method, range of values tested for each relevant tuning parameter, selected value and number of edges obtained in the inferred causal graph with the selected value(s) of the tuning parameter(s).

| Method | Tuning parameter | Range of values tested | Selected value | Number of edges with the selected value |
|---|---|---|---|---|
| PC-stable | $\alpha$ | 0.01 - 0.15 | 0.047 | 155 |
| FCI | $\alpha$ | 0.01 - 0.15 | 0.047 | 154 |
| FCI+ | $\alpha$ | 0.01 - 0.15 | 0.047 | 155 |
| GES | Penalty | 1 - 3 | 2.260 | 154 |
| FGES | Penalty | 1 - 3 | 1.000 | 153 |
| MMHC | $\alpha$ | 0.01 - 0.15 | 0.054 | 150 |
| MMHC | Penalty | 1 - 3 | 1.000 | 150 |
| ARGES | Penalty | 1 - 3 | 2.050 | 151 |
| ARACNe | Threshold quantile | 0.01 - 0.9 | 0.570 | 146 |
| GENIE3 | Threshold quantile | 0.97 - 0.995 | 0.992 | 154 |

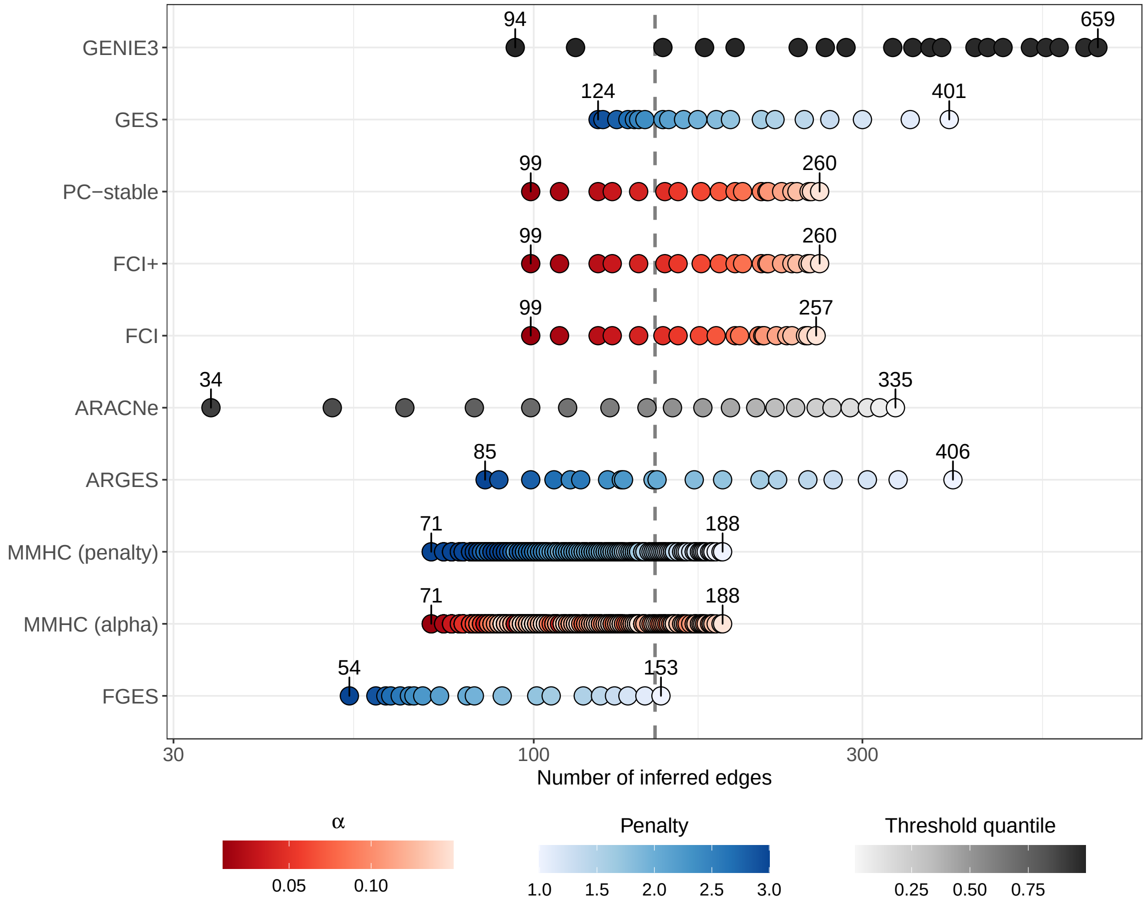


Figure 5.8: Number of edges in the graphs inferred with each causal or network inference method for different values of the tuning parameters. The colour of the dots indicates the corresponding tuning parameter value, with darker shades representing values yielding sparer graphs, while lighter shades represent values yielding denser graphs. Values of the $\alpha$ parameter are shown in shades of red, the penalty parameter in shades of blue, and the threshold quantile parameter in shades of grey.

the different methods is presented in Figure 5.9, with a score of one indicating identical skeletons while a score of zero represents completely dissimilar skeletons (with no edges in common). Similar to what was observed with single-omics simulated datasets in Chapter 3, it is to be noted that the skeleton graphs inferred by the three constraint-based methods (namely PC-stable, FCI and FCI+) are almost identical. MMHC also returns a skeleton similar to those of the constraint-based methods. Likewise, the results of both score-based methods (GES and FGES) are in very good agreement. However, contrary to the observations made with the simulated datasets, ARGES returns a causal graph whose topology is markedly different from the GES result, and from any other inferred graph. The highest skeleton similarity involving ARGES is with GENIE3, and even then it is quite low (0.23). From the two network inference methods, ARACNe returns a graph whose skeleton is most similar to those inferred by the causal inference methods. In particular, its topology is closest to the one inferred with GES (similarity score of 0.63). In general, the similarity scores obtained between

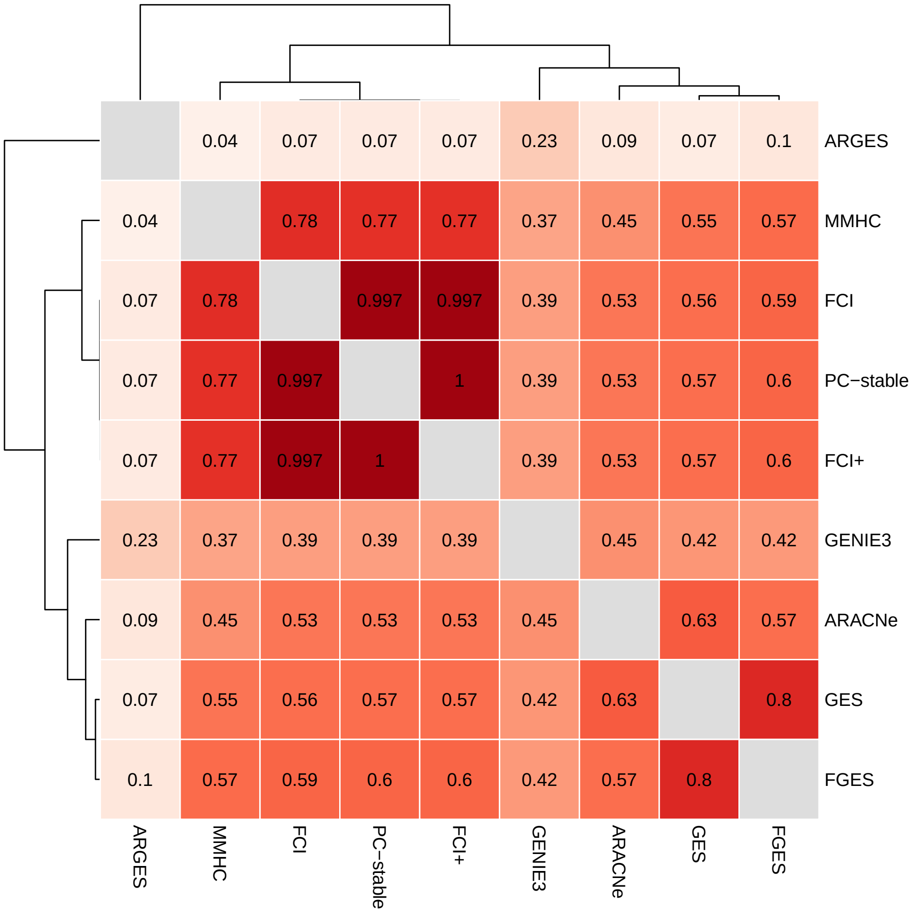


Figure 5.9: Skeleton similarity scores between all causal and network inference methods. A score close to one indicates a high similarity between the skeleton of the inferred graphs. A score close to zero indicates that the two inferred graphs have very different skeletons.

the methods were lower than what was observed in Chapter 3. This is mainly due to the difference in sample size between the two settings. The larger sample size in Chapter 3 resulted in more signal in the data, and as a consequence the different methods were able to detect the causal relationships regardless of their differences in methodology. In the present case on the contrary, the smaller sample size resulted in less signal in the data, and thus the differences in methodology used to detect the causal relationships strongly impacted the resulting networks.

It is interesting to note that a large fraction of the edges inferred by each method link features from a same omics dataset (Figure 5.10). Most inferred relationships are between two transcribed genes, except for GENIE3 which infers mostly edges between metabolites. ARGES is the method that infers the most edges between features from different datasets, while GENIE3 infers almost none. Among the edges inferred across datasets, a large majority links transcribed genes and metabolites, and then genomic variants and transcribed genes. On the contrary, there are very few edges inferred between genomic variants and metabolites. Altogether, these proportions are not surprising, as we expect the

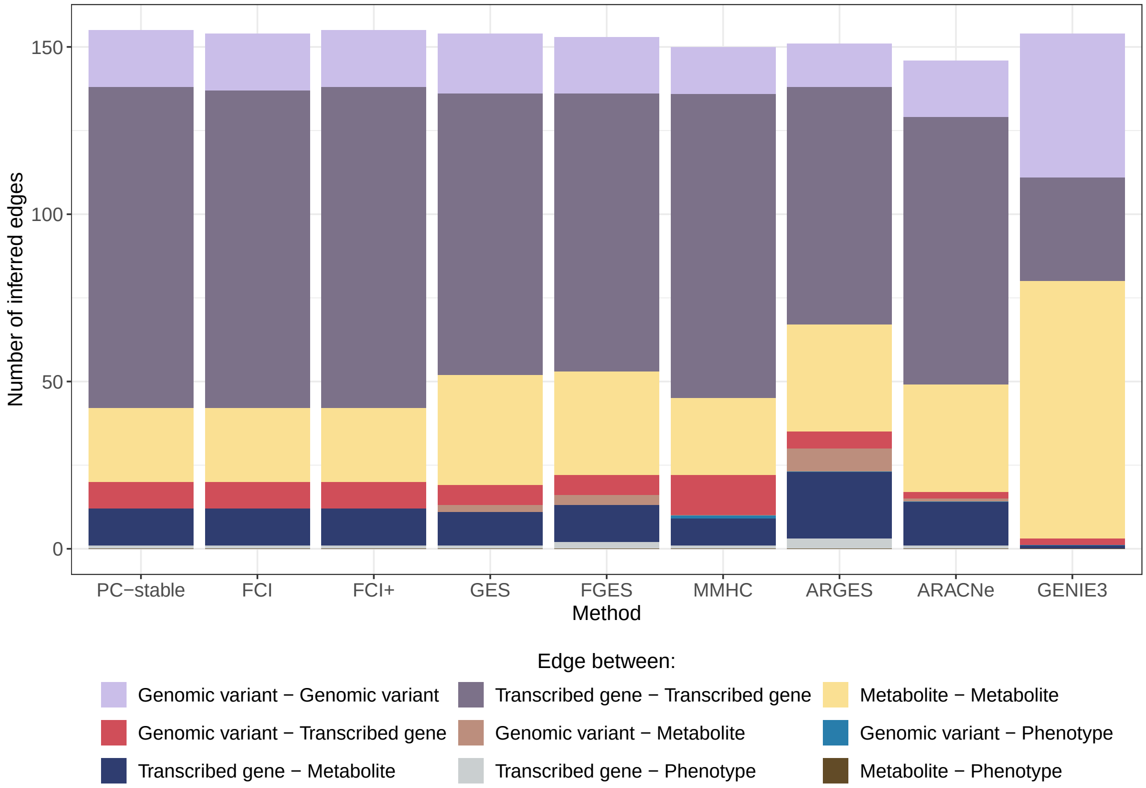


Figure 5.10: Type of edges inferred by each causal and network inference method, i.e. which type(s) of features are linked.

omics datasets to reflect the activity of gene regulatory networks and metabolic pathways, with a few molecules acting as bridges between the different molecular layers. We can notably expect genomic variants to impact gene expression more than metabolic reactions directly, as mRNAs are direct products of DNA transcription. In addition, relationships between the RNA levels of genes encoding enzymes and corresponding compounds might be hindered by post-transcriptional regulations. The effect of post-transcriptional regulations on the detection of gene-gene relationships by causal inference methods has been demonstrated on simulated gene expression data in Chapter 3. Most methods only infer one edge between the bruising mean score and a transcribed gene. Interestingly, for most methods this edge involve either a transcribed gene encoding a LOB domain-containing protein (PGSC0003DMG400020562), potentially involved in starch metabolism (Van Harsselaar et al., 2017) or a transcribed gene coding for a F-box family protein (PGSC0003DMG400005853), with a potential role in defence response (Van Den Burg et al., 2008). Unfortunately, these annotations did not allow me to find a link between the genes' function and tuber bruising in the literature. In addition, MMHC identified an edge between a genomic variant and the bruising mean score, while ARGES inferred three edges involving the bruising mean score and different transcribed genes. An in-depth study of the function of these genes will be required to assess the biological relevance of these inferred relationships.

In order to assess the accuracy of the inferred relationships, a bootstrapping scheme was used to compute for each inferred edge a confidence score. This is done by generating 100 bootstrap datasets, each obtained by sampling with replacement the same number of samples as in the original dataset. The causal inference process is repeated for each bootstrap dataset, and the fraction of occurrence of a given edge over all bootstrap datasets is used as its confidence score. The consensus graph skeleton, i.e. the undirected graph in which the weight of an edge between any two features corresponds to the mean confidence score of the edge across all investigated methods, is presented in Figure 5.11. As mentioned previously, we can see that most edges occur between features of the same dataset. In particular, genomic variants arising from a same genomic region form tightly connected clusters, due to their strong correlation because of physical proximity. Moreover, most of the metabolites found upregulated and downregulated in the high bruising group (compared to the low bruising group) form two distinct modules within the graph, while other non-differentially abundant compounds are scattered in the graph (Supplementary Figure G.4 a)). These two modules most probably correspond to distinct pathways involved in tuber bruising. Indeed, among the cluster of upregulated compounds, several are identified as involved in the linoleic or $\alpha$-linoleic metabolism, while a number of transcribed genes in the downregulated modules are associated with the glutathione pathway. Alternatively, they could correspond to pathways that are activated in response to tuber bruising. More generally, it can be shown that all features selected for the first latent component of DIABLO are found on one side of the graph, while the features retained for the second latent component are found on the other side of the graph (Supplementary Figure G.4 b)). It is not surprising, as DIABLO constructs each latent component with co-varying features, that we can thus expect to be causally related. Interestingly, five transcribed genes were not found causally related to any other features. One possibility to explain this result is that these transcribed genes are associated with the phenotype through additional features not retained by DIABLO.

Amongst the edges with the highest mean confidence score across the methods (above 0.7 - see Table 5.2), there is a link between the compounds 9-HOTrE and 10,12-9-hydroxy-10,12-octadecadienoic acid, both involved in the linoleic acid metabolism; an edge between two transcribed genes encoding for a peptide transporter, one located on chromosome 0 (PGSC0003DMG400022107) and the other on chromosome 6 (PGSC0003DMG400006606); a link between two transcribed genes encoding proteins potentially involved in plant disease resistance (PGSC0003DMG400010887 and PGSC0003DMG402011427); several edges between transcribed genes and a nearby genomic marker, which probably informs about a cis-acting mutation near the corresponding genes; and a link between the compounds bis-dihydrocaffeoyl spermidine and dihydrocaffeoyl spermine (the former being a precursor of the latter). In general, we can observe that the edges with the highest confidence scores are found between two features from a same dataset, or between a transcribed gene and a nearby genomic variant.

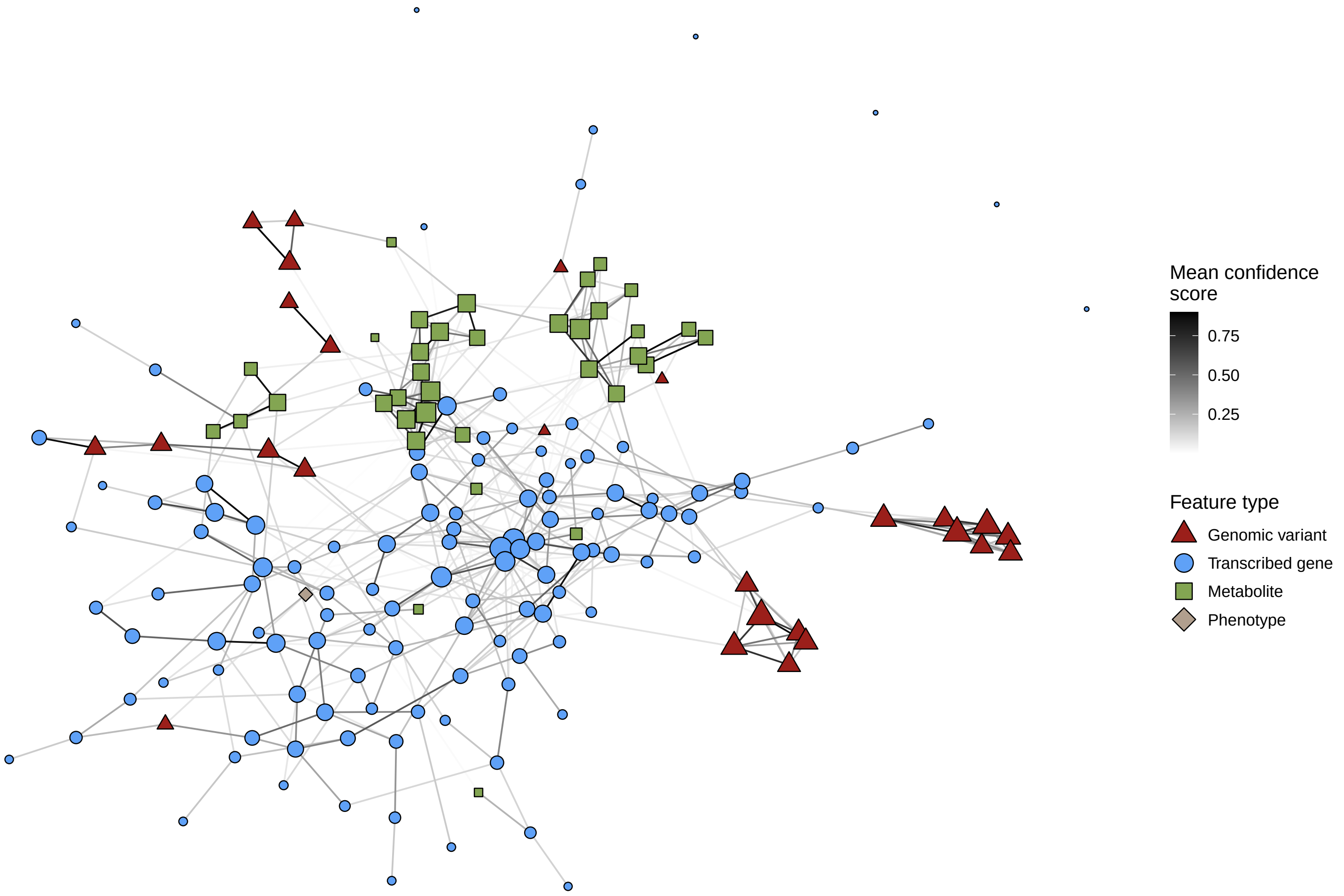


Figure 5.11: Consensus skeleton of the graphs inferred with the nine causal and network inference methods. The nodes represent the features selected with DIABLO (red triangles: genomic variants, blue circles: transcribed genes, green squares: compounds, beige diamond: bruising mean score). Their size indicates the strength (i.e. weighted degree) of the nodes. The colour of the edges represents their mean confidence score over all nine methods.

One interesting question is to assess the overlap between the causal relationships inferred between features of a same dataset and the associations found between transcribed genes or between compounds with WGCNA. As WGCNA returns for each pair of features from a single omics dataset an association score (the topological overlap score), the distribution of the association scores for pairs of features found causally related in the causal graphs was compared to the scores of pairs of features with no edges between them in the causal graphs (Supplementary Figures G.5 and G.6). Without surprise, the topological overlap scores were higher for the pairs of features (either transcribed genes or compounds) found related in the inferred causal graphs. The difference was clearer when looking at the co-abundance network computed for the metabolomics dataset. In the case of the transcriptomics dataset, GENIE3 obtained the highest mean topological overlap score for causally related pairs of features. This implies that features found co-expressed will likely be also found to be causally related.

### Causal queries answers

The interest of going beyond association and inferring causality is to extract information about the directionality of the relationships between features. However, comparing directed causal graphs is not straightforward, as the investigated causal inference methods return different types of causal graphs that have different interpretations. Therefore, the comparison between inferred graphs was performed by comparing their answers to a number of causal queries (see Material and Methods Section 5.3.6). These causal queries search, for each variable in a causal graph, other variables that are related to it via a certain type of causal relationship. The answers will depend on the orientation of the edges within the causal graph as well as the nature of the graph. One can, for example, investigate all features found to be a direct cause of an investigated feature, called parents of the variable. In addition to providing a way to compare causal graphs, the causal queries allow to summarise biologically relevant information from the causal graphs in a way that is intuitive to interpret. For example, obtaining a list of ancestors of a transcribed gene (or compound) provides information about all molecules on which exerting a change could impact the considered feature, i.e. located upstream in the regulatory pathways.

The number of positive answers to the different causal queries with each method is presented in Figure 5.12. Due to the nature of the returned causal graph, FCI, FCI+ and ARACNe cannot answer the parent query. For the same query, FGES obtains only two positive answers, meaning that amongst the 154 edges it inferred, only two were directed. On the contrary, GENIE3 inferred 261 positive answers for this parent query, which indicates that most of the edges it inferred are bidirected. This raises the question of the nature of the graph returned by GENIE3. While GENIE3 returns a list of regulators for each feature, implying a directionality in the relationship, bidirected edges could instead be interpreted as an inability of the method to solve the direction of the regulatory relationship. In such case, GENIE3 is also unable to infer the orientation of most edges, as out of the

Table 5.2: Edges in the consensus skeleton of inferred graphs with a high average confidence score across the causal and network inference methods. Genomic variants are represented as follows: *chromosome, genomic position*; transcribed genes as: *description - chromosome, genomic position (Ensembl ID)*; metabolic compounds as: *description if identified - formula if identified, molecular weight*. Note that most chemical formulas have been automatically generated by the metabolomics analysis software.

| Edge between | And | Edge mean confidence score |
|---|---|---|
| 9-HOTrE (9-hydroxy-10E,12Z,15Z-octadecatrienoic acid) - $C_{18}H_{30}O_3$, 294.22 g/mol | (10E,12Z)-9-Hydroxy-10,12-octadecadienoic acid - $C_{18}H_{32}O_3$, 296.23 g/mol | 0.90 |
| $C_{17}H_{20}N_{10}O_2S$, 428.15 g/mol | 431.15 g/mol | 0.89 |
| $C_{18}H_{26}N_4O_3S$, 378.17 g/mol | $C_{20}H_{30}O_3P_2$, 380.17 g/mol | 0.89 |
| ST4.03ch07, 6,662,497bp | ST4.03ch07, 6,662,506bp | 0.89 |
| $C_{20}H_{28}O_3P_2$, 378.15 g/mol | $C_9H_{23}N_{11}O_2P_2$, 379.15 g/mol | 0.89 |
| pigment from marker pen - 265.16 g/mol | 265.66 g/mol | 0.89 |
| Leucine-rich repeat - ST4.03ch00, 23.1Mb (PGSC0003DMG400010887) | Cc-nbs-lrr resistance protein - ST4.03ch10, 59Mb (PGSC0003DMG402011427) | 0.89 |
| Peptide transporter - ST4.03ch06, 56.1Mb (PGSC0003DMG400006606) | Peptide transporter - ST4.03ch00, 38Mb (PGSC0003DMG400022107) | 0.89 |
| ST4.03ch05, 2,149,643bp | ST4.03ch05, 2,261,796bp | 0.88 |
| Non-symbiotic hemoglobin - ST4.03ch01, 85.5Mb (PGSC0003DMG400025176) | Enoyl-CoA-hydratase - ST4.03ch01, 85.1Mb (PGSC0003DMG403025826) | 0.88 |
| Breakdown product of Glutathione - $C_{13}H_6O$, 178.04 g/mol | $C_{13}H_{17}N_3O_2P_2$, 309.08 g/mol | 0.87 |
| Multidrug resistance pump - ST4.03ch08, 2.9Mb (PGSC0003DMG400004474) | ST4.03ch08, 3,137,398bp | 0.87 |
| ST4.03ch07, 54,289,955bp | ST4.03ch07, 54,813,343bp | 0.87 |
| $C_9H_8O_2$, 148.05 g/mol | Possible breakdown product - $C_{10}H_8O_3$, 176.05 g/mol | 0.87 |
| ST4.03ch08, 4,926,541bp | ST4.03ch08, 4,926,550bp | 0.86 |

Table 5.2: Edges in the consensus skeleton of inferred graphs with a high average confidence score across the causal and network inference methods. Genomic variants are represented as follows: *chromosome, genomic position*; transcribed genes as: *description - chromosome, genomic position (Ensembl ID)*; metabolic compounds as: *description if identified - formula if identified, molecular weight*. Note that most chemical formulas have been automatically generated by the metabolomics analysis software. *(continued)*

| Edge between | And | Edge mean confidence score |
|---|---|---|
| Possible breakdown product - $C_{10}H_8O_3$, 176.05 g/mol | $C_8H_{15}N_{11}$, 265.15 g/mol | 0.86 |
| BHLH domain class transcription factor - ST4.03ch03, 58Mb (PGSC0003DMG400014246) | DNA-directed RNA polymerase II 19 kD polypeptide rpb7 - ST4.03ch03, 58Mb (PGSC0003DMG400014251) | 0.86 |
| Heat shock protein binding protein - ST4.03ch09, 3.8Mb (PGSC0003DMG400002680) | TVLP1 - ST4.03ch07, 51.9Mb (PGSC0003DMG400027646) | 0.85 |
| L-Glutathione (reduced) - $C_{10}H_{17}N_3O_6S$, 307.08 g/mol | L-Glutathione (reduced) - $C_{10}H_{17}N_3O_6S$, 307.08 g/mol | 0.85 |
| 366.65 g/mol | N1, N5, N14-(dihydrocaffeoyl)spermine - $C_{37}H_{50}N_4O_9$, 694.36 g/mol | 0.83 |
| N1,N10-Bis(dihydrocaffeoyl)spermidine - $C_{25}H_{35}N_3O_6$, 473.25 g/mol | N1, N5, N14-(dihydrocaffeoyl)spermine - $C_{37}H_{50}N_4O_9$, 694.36 g/mol | 0.80 |
| ST4.03ch11, 597,680bp | ST4.03ch11, 703,484bp | 0.75 |
| ST4.03ch11, 597,680bp | ST4.03ch11, 644,490bp | 0.73 |
| ST4.03ch07, 54,526,301bp | ST4.03ch07, 54,526,505bp | 0.72 |
| ST4.03ch11, 644,490bp | ST4.03ch11, 941,448bp | 0.71 |
| $C_8H_{21}N_{18}PS$, 432.17 g/mol | $C_5H_{13}N_5S$, 175.09 g/mol | 0.70 |

154 inferred edges, only a third were not bidirected. On the contrary, PC-stable, GES and ARGES were able to orient most of their edges. MMHC is not considered for this comparison as it returns by default a fully directed graph. FGES obtains the highest number of positive answers for both the potential parent and potential ancestor queries. This is due to the fact that it was not able to orient a majority of its edges. Similarly to previous observations, most positive answers to the different causal queries are found between features from a same dataset. This can be visualised in Figure 5.13, which presents the number of positive answers obtained for each causal query between different types of features (i.e. genomic variant, transcribed gene, metabolite or phenotype). It clearly shows that for each causal query, more positive answers are found between two features from a same dataset than between features from different datasets. In addition, most positive queries targeting the phenotype arise from transcribed genes (no genomic variants or metabolites are found to be parents or potential parents of the phenotype).

Similarly to the skeleton similarity metric, a score quantifying the similarity between the methods' answers to a given query can be computed. While the topology (i.e. skeleton) of the inferred graphs can be really similar across the investigated methods, the orientation of the edges, and consequently the answer to the causal queries, are different between the methods. For example, the maximum similarity obtained between the answers to the parent query is 0.35, obtained when comparing the answers of GES and MMHC. For the potential parent query, which also accounts for edges with uncertain orientation in the graph, the similarity between the answers of the constraint-based methods (PC-stable, FCI and FCI+) does not exceed 0.54, and the similarity between the answers of GES and FGES is found to be 0.61. The answers are even more different when considering the ancestor and potential ancestor queries, which comes as no surprise as these take into account paths in the causal graph. The only exception is the relatively high similarity between the answers of FGES and ARACNe for the potential ancestor query (0.56). Taken together, these results indicate that while the methods can detect similar interactions between the different features, they do not often agree on the causal orientation of these interactions. Therefore, the presence of a directed edge inferred by all the methods can be considered as a strong evidence in favour of the presence of this relationship in the biological system.

Also, as can be seen in Supplementary Figure G.7, the confidence score of the answers to the different queries can be consistently low for some methods. For example, PC-stable infers parental relationships with confidence scores not exceeding 0.6, and for FCI+ the confidence scores of its answers to the ancestor query are all below 0.2. This shows that some of the methods, in particular PC-stable, FCI and FCI+, are not consistent when inferring the orientation of causal relationships. On the contrary, ARACNe and GENIE3 return high confidence scores for their queries answers, especially for the potential parent query, indicating that they consistently infer a similar topology and orientation across the bootstrap datasets. Note however than because the ancestor query depends on

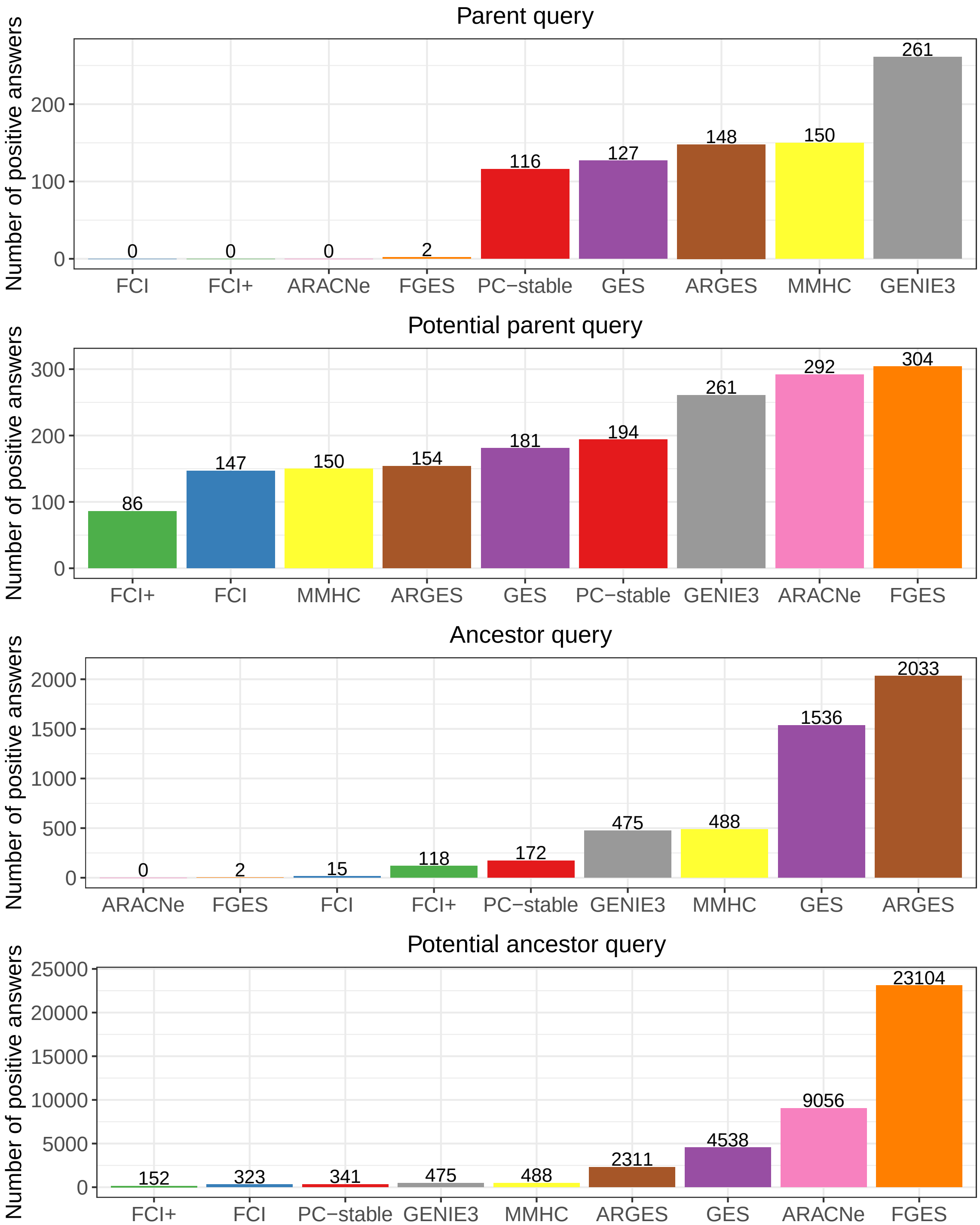
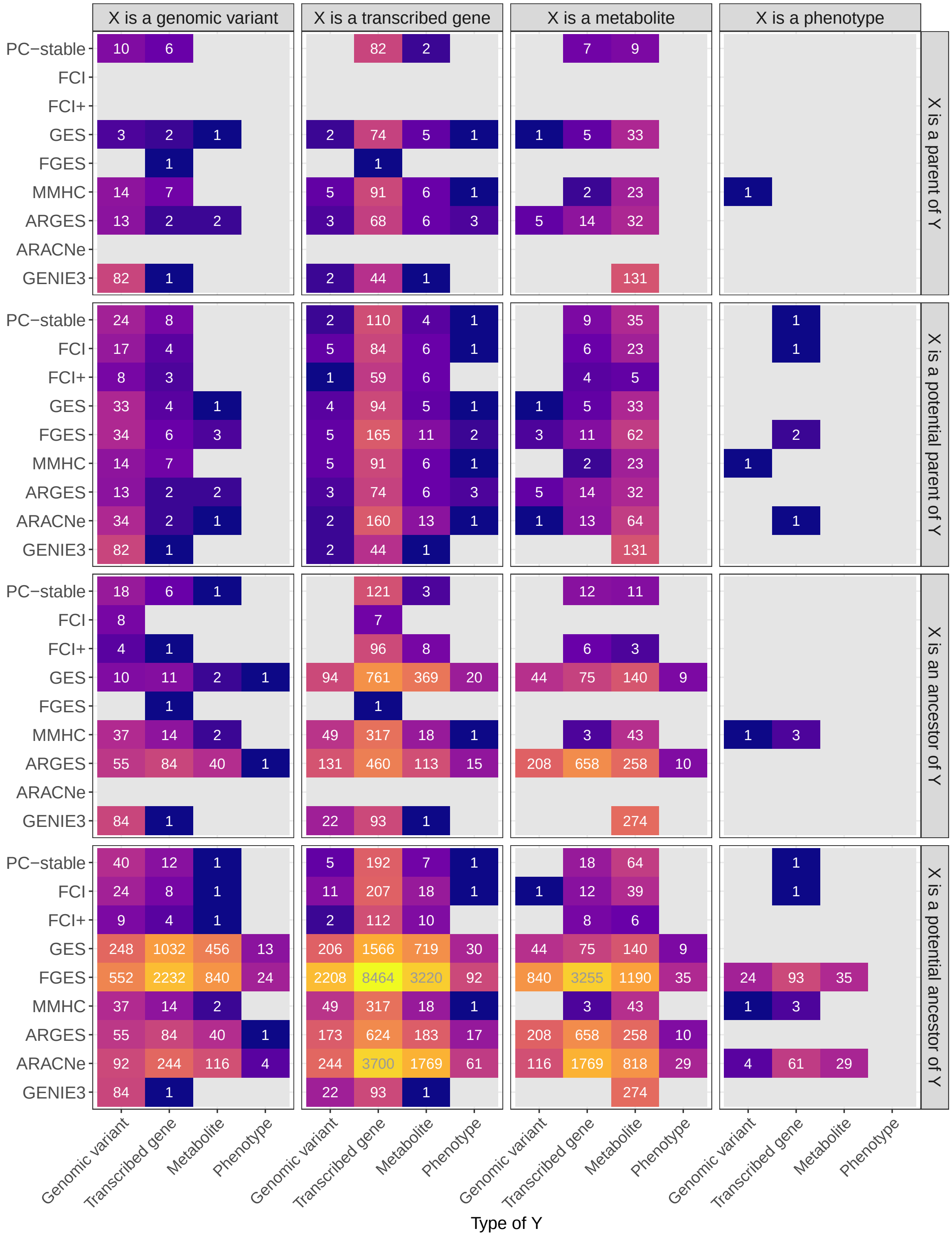


Figure 5.12: Number of positive answers to each causal query for the graphs inferred with the different causal and network inference methods.



X is a parent of Y

| | X is a genomic variant | | | | X is a transcribed gene | | | | X is a metabolite | | | | X is a phenotype | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type of Y | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype |
| PC−stable | 10 | 6 | | | | 82 | 2 | | | 7 | 9 | | | | | |
| FCI | | | | | | | | | | | | | | | | |
| FCI+ | | | | | | | | | | | | | | | | |
| GES | 3 | 2 | 1 | | 2 | 74 | 5 | 1 | 1 | 5 | 33 | | | | | |
| FGES | | 1 | | | | 1 | | | | | | | | | | |
| MMHC | 14 | 7 | | | 5 | 91 | 6 | 1 | | 2 | 23 | | 1 | | | |
| ARGES | 13 | 2 | 2 | | 3 | 68 | 6 | 3 | 5 | 14 | 32 | | | | | |
| ARACNe | | | | | | | | | | | | | | | | |
| GENIE3 | 82 | 1 | | | 2 | 44 | 1 | | | | 131 | | | | | |

X is a potential parent of Y

| | X is a genomic variant | | | | X is a transcribed gene | | | | X is a metabolite | | | | X is a phenotype | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type of Y | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype |
| PC−stable | 24 | 8 | | | 2 | 110 | 4 | 1 | | 9 | 35 | | | 1 | | |
| FCI | 17 | 4 | | | 5 | 84 | 6 | 1 | | 6 | 23 | | | 1 | | |
| FCI+ | 8 | 3 | | | 1 | 59 | 6 | | | 4 | 5 | | | | | |
| GES | 33 | 4 | 1 | | 4 | 94 | 5 | 1 | 1 | 5 | 33 | | | | | |
| FGES | 34 | 6 | 3 | | 5 | 165 | 11 | 2 | 3 | 11 | 62 | | | 2 | | |
| MMHC | 14 | 7 | | | 5 | 91 | 6 | 1 | | 2 | 23 | | 1 | | | |
| ARGES | 13 | 2 | 2 | | 3 | 74 | 6 | 3 | 5 | 14 | 32 | | | | | |
| ARACNe | 34 | 2 | 1 | | 2 | 160 | 13 | 1 | 1 | 13 | 64 | | | 1 | | |
| GENIE3 | 82 | 1 | | | 2 | 44 | 1 | | | | 131 | | | | | |

X is an ancestor of Y

| | X is a genomic variant | | | | X is a transcribed gene | | | | X is a metabolite | | | | X is a phenotype | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type of Y | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype |
| PC−stable | 18 | 6 | 1 | | | 121 | 3 | | | 12 | 11 | | | | | |
| FCI | 8 | | | | | 7 | | | | | | | | | | |
| FCI+ | 4 | 1 | | | | 96 | 8 | | | 6 | 3 | | | | | |
| GES | 10 | 11 | 2 | 1 | 94 | 761 | 369 | 20 | 44 | 75 | 140 | 9 | | | | |
| FGES | | 1 | | | | 1 | | | | | | | | | | |
| MMHC | 37 | 14 | 2 | | 49 | 317 | 18 | 1 | | 3 | 43 | | 1 | 3 | | |
| ARGES | 55 | 84 | 40 | 1 | 131 | 460 | 113 | 15 | 208 | 658 | 258 | 10 | | | | |
| ARACNe | | | | | | | | | | | | | | | | |
| GENIE3 | 84 | 1 | | | 22 | 93 | 1 | | | | 274 | | | | | |

X is a potential ancestor of Y

| | X is a genomic variant | | | | X is a transcribed gene | | | | X is a metabolite | | | | X is a phenotype | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type of Y | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype | Genomic variant | Transcribed gene | Metabolite | Phenotype |
| PC−stable | 40 | 12 | 1 | | 5 | 192 | 7 | 1 | | 18 | 64 | | | 1 | | |
| FCI | 24 | 8 | 1 | | 11 | 207 | 18 | 1 | 1 | 12 | 39 | | | 1 | | |
| FCI+ | 9 | 4 | 1 | | 2 | 112 | 10 | | | 8 | 6 | | | | | |
| GES | 248 | 1032 | 456 | 13 | 206 | 1566 | 719 | 30 | 44 | 75 | 140 | 9 | | | | |
| FGES | 552 | 2232 | 840 | 24 | 2208 | 8464 | 3220 | 92 | 840 | 3255 | 1190 | 35 | 24 | 93 | 35 | |
| MMHC | 37 | 14 | 2 | | 49 | 317 | 18 | 1 | | 3 | 43 | | 1 | 3 | | |
| ARGES | 55 | 84 | 40 | 1 | 173 | 624 | 183 | 17 | 208 | 658 | 258 | 10 | | | | |
| ARACNe | 92 | 244 | 116 | 4 | 244 | 3700 | 1769 | 61 | 116 | 1769 | 818 | 29 | 4 | 61 | 29 | |
| GENIE3 | 84 | 1 | | | 22 | 93 | 1 | | | | 274 | | | | | |

Figure 5.13: Number of positive answers to each causal query (row panels) depending on the type of features involved, for the different causal and network inference methods.

the paths in the inferred graphs, it is natural that the confidence scores obtained for this query are lower than the ones obtained for the parent query. Consequently, the bootstrap confidence scores obtained for parental queries and ancestral queries should be compared only with caution, as they likely do not lie in the same scale.

Among the positive answers with high mean confidence score across all methods, I found the compound 9-HOTrE to be a parent of 10,12-9-hydroxy-octadecadienoic acid, with a mean confidence score of 0.41, and a potential parent of the latter with a mean confidence score of 0.74. It is interesting as they are both octadecanoids, but they do not seem involved in the same metabolic reaction or pathway. Also, a genomic variant at position 31.Mb on chromosome 8 is found to be a direct parent of a multidrug resistant pump-encoding gene (PGSC0003DMG400004474), located at 2.9Mb on the same chromosome (mean confidence score: 0.36, mean confidence score for the potential parent query: 0.64). This probably points to a cis-mutation affecting the expression of the gene. Alternatively, it could be due to linkage disequilibrium. It is encouraging that the methods detect this edge as oriented from the genotype to the transcribed gene's expression and not the opposite. Other high confidence-answers involve unidentified compounds or genomic compounds in linkage disequilibrium. Several parent-child pairs are found in both directions, but one with a higher confidence score than the other. For example, a transcribed gene encoding a DNA-directed RNA polymerase II subunit (PGSC0003DMG400014251) is found to be a parent of a BHLH domain class transcription factor (PGSC0003DMG400014246) with mean confidence score across all methods, 0.3, while the opposite relationship is found with a confidence score of 0.25.

The positive answers targeting the bruising mean score received in general low confidence score. A transcribed gene coding for a LOB domain-containing protein (PGSC0003DMG400020562) was found to be a parent and potential parent of the phenotype with mean confidence score of 0.07 and 0.11, respectively. Similarly, an F-bloc family protein-coding gene (PGSC0003DMG400005853) was found to be a parent of the bruising mean score with mean confidence score of 0.04 and a potential parent at 0.08. Two other parents of the bruising mean score are a conserved gene of unknown function (PGSC0003DMG400017523 - mean confidence score of 0.04) and a transcribed gene encoding a leucine-rich repeat protein (PGSC0003DMG400010887 - mean confidence score of 0.01). Amongst the inferred ancestors of the bruising mean score, several genes encoding proteins can be found, such as a big map kinase (PGSC0003DMG400002452), a heat shock factor (PGSC0003DMG400028414), a N-acylneuraminate-9-phosphatase (PGSC0003DMG400029856 - involved in the metabolism of amino and nucleotide sugars) or an RNA binding protein (PGSC0003DMG400027944). Compounds such as the dyhydrocaffeoyl spermine and L-glutathione are also found, as well as other unidentified compounds. The first two compounds confirm the implication of the glutathione pathway in the bruising response, probably as a response to stress.

### Latent variables

An assumption commonly made by causal inference method is that of causal sufficiency. It assumes that all variables involved in the causal system under investigation are observed, and that no unobserved confounders affect the observed variables. It is however rarely satisfied in practice, and consequently methods were developed that account for the presence of latent variables (i.e. unobserved confounders) impacting the observed variables. Such methods include FCI and FCI+. In the resulting inferred graph, a bidirected edge links two variables that are found affected by a latent variable. In the present case, as the observed variables depended on a previous feature selection step, it is possible that a number of features involved in the causal system have not been included. In order to assess the extent to which the feature selection step excluded important features for this causal system, one can look at the number of estimated latent variables in the system. This is done by counting the number of bidirected edges in the graphs inferred by FCI and FCI+. It has been noted however by Ogarrio et al. (2016) that FCI+, which is a variant of the FCI algorithm, tends to overestimate the presence of latent variables, when the number of observations in the dataset is small. This is certainly the case here, as there are only 74 observations. Consistently with this, FCI returns an inferred graph with only three bidirected edges, while FCI+ finds 74. Over the three bidirected edges inferred by FCI, two are between co-localised genomic variants. The third bidirected edge links two transcribed genes encoding proteins involved in unrelated pathways, which could be an indication that a factor not included in the analysis regulates both genes and acts as bridge between the two pathways. It is only inferred in 14% of the bootstrap datasets. The small number of bidirected edges inferred by FCI could be an indication that the feature selection step retained most of the important causal variables for the investigated system. Amongst the bidirected edges inferred by FCI+, 51% are between two transcribed genes, 23% between two metabolites and 15% between two genomic variants. They could represent situations in which FCI+ was not able to infer the direction of the causal relationships between the features, due to the small number of observations available.

### Adding a blacklist

As mentioned in the previous section, a number of high-confidence relationships are found between closely located genomic variants. This is due to the high correlation between the variants, arising from linkage disequilibrium due to their close proximity on the genome. Therefore, I do not expect these correlations to reflect a causal mechanism. More generally, we can expect the causal relationships to flow from, rather than towards, genomic variants, as at the scale that we are interested in, gene expression and metabolism cannot impact genomic variation. It would thus be desirable to prevent the inclusion of causal edges directed towards genomic variants during the causal reconstruction process. For some of the implementations tested here, it is possible to provide a list of directed edges that will be ignored during the reconstruction. This is the case for the four methods PC-stable, FGES, MMHC and GENIE3. Therefore, the causal inference task and bootstrapping scheme were repeated with this four methods by providing as a blacklist the list of all possible directed edges targeting a genomic variant. The average confidence score of the edges and causal queries answers over the four methods obtained without and with the blacklist were then compared. As other constraint-based methods returned results somewhat similar to PC-stable, and the other score-based method was also inferring a topology close to the one found with FGES, we can assume that adding the blacklist to these other methods would have a similar result as well.

The consensus skeleton of the inferred graphs with and without blacklist are compared in Figure 5.14. Interestingly, from the groups of linked genomic variants, only one or a few are connected to the rest of the network when adding the blacklist. The remaining variants are now disconnected from the other variables. This highlights the bias of DIABLO in selecting highly correlated variants, whereas the causal flow can be summarised with only one of them. This could be an interesting way of fine-mapping potential causal variants from a list of highly correlated GWAS high-scoring markers. Another consequence of adding the blacklist is the apparition or large increase in confidence score of some edges, e.g. between a transcribed gene encoding a big map kinase (PGSC0003DMG400002452) and one encoding a heat shock factor (PGSC0003DMG400028414), which was not detected without the blacklist, and is detected with a mean confidence score of 0.3 with the blacklist. Interestingly, this change in confidence score impacts mainly edges between metabolic compounds, while for the most part the confidence score of edges between transcribed genes remains similar. The edges with the highest mean confidence scores when adding the blacklist are presented in Supplementary Table G.1.

When comparing the queries answers of the methods with and without the blacklist, it appears that the impact of the blacklist on the orientation of the edges is more pronounced for FGES. While without the blacklist only two of the inferred edges were oriented, with the blacklist all edges are now oriented. This is a very interesting result, as it implies that forcing the causal flow to stem from the genomic variants helped the algorithm determine the direction of the other causal relationships. It showcases the importance of including biological information to help the reconstruction process. Also, with the addition of the blacklist, a number of queries answers targeting the phenotype disappear. With the blacklist, only transcribed genes as well as a few genomic variants are related to the bruising mean score, while the association of compounds to the bruising mean score disappear. With the blacklist, ancestors of the bruising mean score include the big-map-kinase gene (PGSC0003DMG400002452), F-box family protein-coding gene (PGSC0003DMG400005853), LOB domain-containing protein (PGSC0003DMG400020562) and RNA binding protein (PGSC0003DMG400027944). In addition, two genomic variants, one on chromosome 5 (4.8Mb) and the other on chromosome 7 (54.3Mb) are found as ancestors of the phenotype, which was not the case without the blacklist.

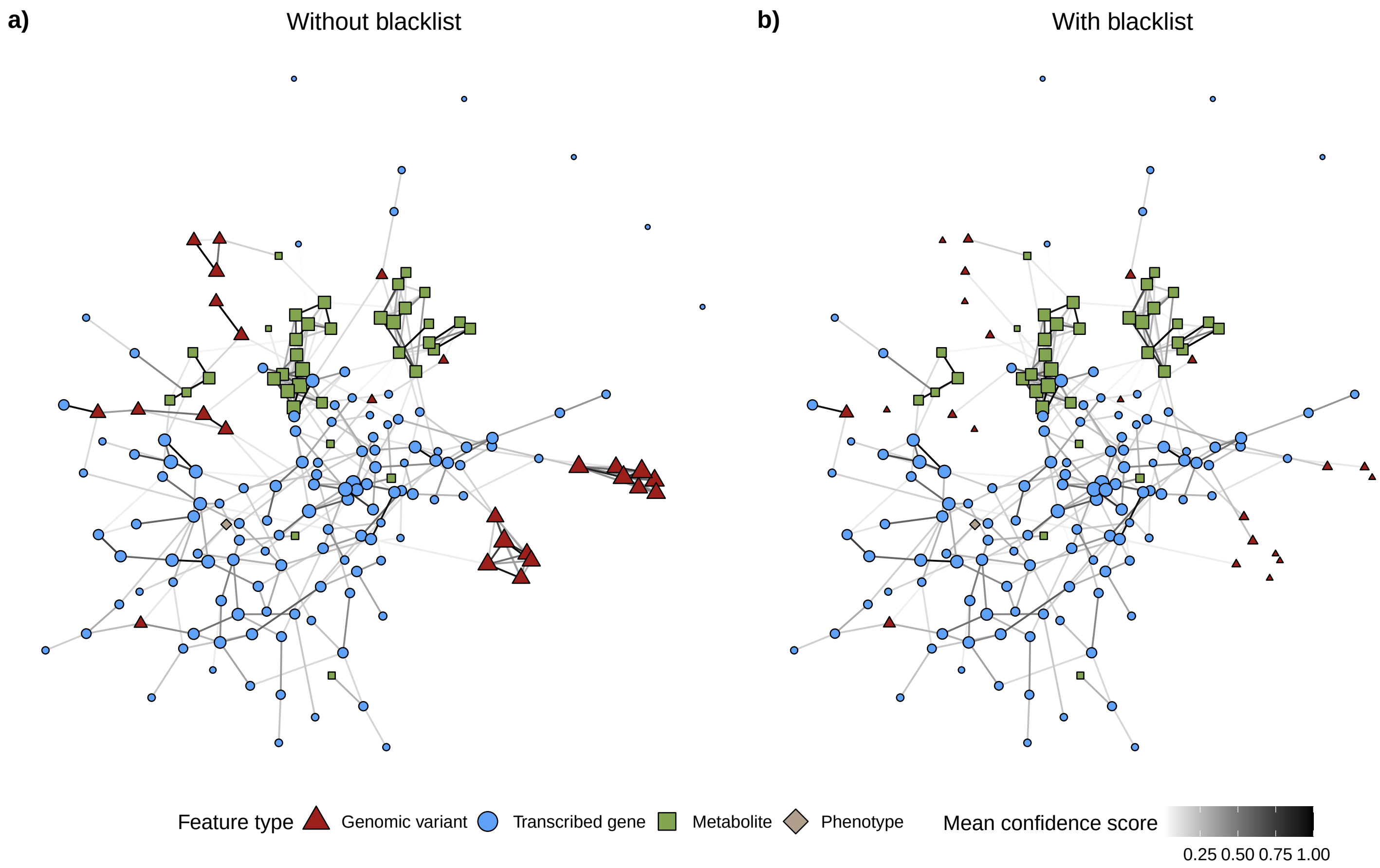


Figure 5.14: Consensus skeleton of the graph inferred with PC-stable, FGES, MMHC and GENIE3, a) without a blacklist and b) with inclusion of a blacklist during the reconstruction process. The nodes represent the features selected with DIABLO (red triangles: genomic variants, blue circles: transcribed genes, green squares: compounds, beige diamond: bruising mean score). Their size indicates the strength (i.e. weighted degree) of the nodes. The colour of the edges represents their mean confidence score over all four methods.

## 5.5 Conclusion

In this work, I aimed at bridging the gap between genotype and phenotype in the case of potato tuber bruising following mechanical impact. In an effort to uncover the biological mechanisms underlying this trait, measurements obtained at different omics scales were integrated, namely genomics, transcriptomics and metabolomics, along with phenotypic data. This was done by selecting from all three omics datasets features that were best suited to discriminate the phenotypic outcome and which co-varied. Nine state-of-the-art causal and network inference methods were then applied to the selected features in order to reconstruct causal relationships amongst them, and the results were discussed and interpreted biologically. At each step of this integrative analysis, the results were compared to typical single-omics analysis such as differential expression analysis or co-expression network reconstruction. This allowed me to assess the added benefit of combining the omics datasets rather than merely analysing them independently.

In Chapter 4, a genome-wide association study was performed to detect genomic regions associated with the bruising response. A differential expression analysis was also performed on the transcriptomics dataset to uncover potential genes driving the bruising response, and reconstructed a co-expression network amongst the measured transcribed genes. In this chapter, I extended this set of single-omics analyses to a metabolomics dataset obtained on the same set of samples. Preliminary analyses revealed the need to perform data normalisation and correct for biases arising from sample processing order. A differential analysis uncovered 107 differentially abundant compounds, among which a number that were found to be involved in the glutathione pathway, as well as the linoleic and $\alpha$-linoleic pathways. The involvement of such pathways in tuber bruising is not surprising as they play important roles in plants response to stress. The reconstruction of a network of co-abundance amongst the compounds confirmed the implication of compounds found to be differentially expressed in common pathways. This could be an interesting approach to compound identification in untargeted metabolomics: compounds clustered in a similar co-abundance module can be assumed to be involved in similar pathways. Therefore identified metabolites in a same module can provide clues about the possible identity of the non-identified compounds, via a guilt-by-association analysis.

In order to go beyond independent single-omics analysis, the DIABLO algorithm was used to select features from the three omics datasets involved in tuber bruising. This feature selection step aimed at reducing the dimension of the datasets for the subsequent causal inference analysis, by focusing on molecular features relevant to the phenotype of interest, i.e. tuber bruising. Feature selection for the transcriptomics and metabolomics datasets were in good agreement with the results from the single-omics analyses, with a number of differentially expressed features being selected. Interestingly, the construction by DIABLO of successive latent components in order to perform features selection allowed the inclusion of non-differentially expressed features, which would have been missed by the single-omics analyses. These features could be retained by DIABLO because of the synergy, i.e. co-variation, between the different omics datasets. This emphasises the need to study omics datasets in combination rather than independently. For example, the gene encoding for the glutathione S-transferase was not found to be differentially expressed, but was retained with DIABLO alongside several metabolic compounds involved in the glutathione metabolism. As this protein plays an important role in cell detoxification during response to stress, I hypothesise that it also plays a role in the response to tuber bruising. Another observation that could be made is that the selection of genomic variants by DIABLO is subject to the same biases as an association study. Mainly, the presence of highly correlated genomic variants due to close proximity in the genome, as well as population structure amongst the observations, hinders the selection of true causal variants. It is therefore necessary to control for these aspects before using DIABLO on genomics datasets. In this work, it was done by restricting the genomics dataset to only these variants found associated with the phenotype with an association test that corrected for the effect of population structure. It would be interesting to develop a way to automate this filtering when applying DIABLO to genomics datasets.

A number of causal inference methods were applied to the retained features, in order to reconstruct the flow of information through the different omics layers. Overall, methods based on similar statistical concepts returned inferred graphs with similar topology (skeleton), with the exception of ARGES, whose results were markedly different from any other method. This stands in contrast to the results of the comparisons performed on single-omics simulated data in Chapter 4, in which ARGES was found to yield very similar results to GES. Inferred causal relationships mainly involved features from the same omics layer, with some molecules acting as bridges between the layers. One possible reason could be that even after scaling each dataset, variations across samples were more similar within an omics dataset than across the different datasets. However, it is plausible that gene products interact more together than with metabolites, with a few transcripts acting as bridges between the omics layers. In particular, very few molecules were directly related to the phenotype (i.e. the bruising mean score of the observations). This could have several explanations. First, it is possible that the small sample size reduces the power of the analysis. Also, observing only a snapshot of the consequences of bruising, rather than a time series of the response, might prevent us from observing molecules with transient responses.

Interestingly, the different inferred graphs were in good agreement with the co-expression networks constructed for the transcriptomics and metabolomics datasets. This is to be expected as two highly co-expressed features are more likely to be causally related than features that are not co-expressed. Conversely, it could also indicate that the causal inference methods did not have enough data to distinguish between co-expression and true causality. This second possibility is reinforced by the fact that the different methods were not in good agreement with regards to the directionality of inferred causal relationships. More generally, using a bootstrapping scheme to assess the confidence in the reconstructed relationships highlighted that most methods were not consistent when inferring the orientation of causal edges between features. This problem most likely arises from the small sample size of the dataset considered. Indeed, the dataset contained measurements for 74 observations, a very small number compared to the sample sizes used in typical causal inference studies, i.e. hundreds to tens of thousands (see e.g. Constantinou et al., 2020; Heinze-Deml et al., 2018). Nevertheless, the biological relevance of some interactions detected highlights the usefulness of the present analysis. Also, the use of causal queries to summarise the information encoded in the causal graphs proved very useful. Answers to these queries provide a straightforward interpretation of the results of causal inference in a biological setting, particularly in the context of regulatory pathways. Rather than relying on graph visualisation, which can be uninformative in the presence of many variables, they allow scientists to quickly compare with known regulations or detect new relationships between features. Lastly, the impact of using a blacklist during the causal graph reconstruction to prevent the methods from adding edges directed towards genomic variants was assessed. This enforces the fact that at the scale that we are interested in, causality flows from the genotype, i.e. the genomic variants impact gene expression and metabolite levels, rather than the opposite. I found that using the blacklist helped in the orientation of inferred edges, especially with FGES. Therefore, it could be an interesting option to consider for reconstruction of causal graphs involving genomic variants, even if it prevents the exploration of the full graph space during the graph reconstruction process.

A number of limitations can be identified in this analysis. Firstly, biological interpretation and validation of the results were hindered by the lack of identification of a number of features, in particular metabolic compounds. Compound identification in untargeted metabolomics studies is a crucial, difficult and time-consuming step, that can provide important insights. In addition, the presence of detected compounds that arise from the same molecule but with different retention time, due to technical differences between the processing of samples, can lead to bias as they potentially increase the number of features tested and thus the p-values correction. Linking the identified metabolites to metabolic pathways is another difficulty, due to the difference in nomenclatures between databases (Jamil et al., 2020). The same difficulties were encountered with the transcriptomics dataset, as linking genes to regulatory pathways requires the translation of gene IDs between several databases. As a result, very few genes could be mapped to the KEGG database, which also contains information about metabolic pathways. One solution would be to develop or make use of plant- or even species-specific databases, which might not be available for specific organisms of interest. Multi-omics integration analyses should not overlook this step of features identification, as it can be key to interpret its results. Another difficulty lies in setting the number of features to retain as involved in the bruising response. Indeed, leaving out important molecules from the causal inference task violates the assumption of causal sufficiency made by most inference methods, as unobserved variables have a causal impact on the observed features. In the present case, I used the phenotype groups average silhouette as a measure of the ability of the selected features to separate the phenotypic outcomes. It is however not guaranteed that this measure provides a reliable indicator of whether all relevant features have been selected. Another option would be to replace missing values in the genomics dataset in order to use cross-validation as recommended by the others of the `mixOmics` package. Again, benchmarks could be used to validate the use of such an approach for feature selection. Lastly, as mentioned previously, the small sample size of the datasets limits the performance of the causal inference methods. This resulted in the lack of agreement and consistency between the oriented causal relationships inferred by the different methods, and results should thus be interpreted with care. This study, however, provides a proof of concept of the methodology used, and can be applied to larger datasets in the future.

## 5.6 Acknowledgements

Thank you to Plant and Food Research for providing the data used in this chapter. In particular, I am grateful for the help of Dr Samantha Baldwin, Dr Susan Thomson, Martin Shaw, Nigel Joyce, Dr Rebecca Bloomer and Katrina Monaghan for generating and pre-processing the genomics, transcriptomics, metabolomics and phenotypic datasets. This work was supported by The New Zealand Institute for Plant and Food Research Limited Strategic Science Investment Fund (SSIF) from the New Zealand Ministry of Business, Innovation and Employment (MBIE).

# Chapter 6

# General Discussion

## 6.1 Review of present omics data integration analyses for GRN reconstruction

Precisely deciphering the molecular mechanisms linking genotype to phenotype across different cellular layers is a crucial challenge in modern biology, with implications in numerous fields: from agriculture, where it can inform animal or crop breeding, to medicine, to help in disease diagnostics or developments of personalised treatments. Recent advances in technologies have enabled routine measurements of genotypes, as well as levels of intermediate molecules such as transcripts, proteins or metabolites. These omics datasets can then be leveraged to gain insight into biological mechanisms driving phenotypes of interest, from disease resistance in plants or humans to yield of crops or dairy cattle, for example. In response, statistical and computational tools have been developed to extract relationships between genotype and phenotype, by assessing the association between variations at the genomic level and changes in traits of interest. In addition, algorithms have been proposed to uncover the regulatory mechanisms occurring in cells from observational measurements.

However, there is still a need for methodologies that can integrate measurements of different omics layers and reconstruct regulatory relationships across these layers in the context of a specific trait of interest. In particular, many tools rely on statistical associations between molecular features. On the contrary, the detection of causal relationships between variables from observational data has rarely been applied to biological systems, particularly to multi-omics settings. Moreover, while evaluating the performance of statistical tools on synthetic datasets is a widespread practice, the models used to generate the benchmark datasets are still overlooking important biological regulatory mechanisms, resulting in optimistic performance estimates, far from what is observed with experimental datasets. Designing simulation tools that emulate the complexity and multi-scale property of biological systems is key to a realistic evaluation of existing analysis tools, and can pave the way to improvements of these methods.

In this thesis, I aimed to bring together the concepts of multi-omics analysis and causal inference, in order to reconstruct causal molecular regulatory networks linking genotype and phenotype. In particular, I focused on two different aspects of this problem: (i) benchmarking causal inference methods in the context of GRN inference, and (ii) inference of a causal multi-omics network in the tetraploid potato in the context of tuber bruising.

## 6.2 Benchmarking of statistical causal inference methods for GRN reconstruction

A number of statistical algorithms assessing causal relationships amongst a set of variables from observational data have been proposed in the last two decades, and studies have assessed and compared their performance in a general setting. However, these tools have rarely been applied to and evaluated in the context of reconstructing biological regulatory networks. This is however a critical first step in applying causal inference methods to biological datasets.

In this thesis, I developed a simulation tool, the R package `sismonr`, that generates synthetic gene expression dataset for benchmarking of network and causal inference methods (Chapter 2). Existing simulators – such as GeneNetWeaver (Schaffter et al., 2011), SysGenSIM (Pinna et al., 2011) or GeNGe (Hache, Wierling, et al., 2009) – adopt a transcript-centric view of gene expression regulation, which ignores alternative mechanisms of expression regulation mechanisms – such as post-transcriptional regulation – pervasive in biological systems. This leads to simulations with unrealistically high correlations between the transcript levels of regulators and those of target genes, thus simplifying the problem of reconstructing regulatory interactions between genes from their transcripts levels. `sismonr` improves upon these simulators in three important aspects. First, it explicitly models different types of post-transcriptional regulation between genes: namely regulation of translation, RNA and protein decay, as well as post-translational modification. This provides a more realistic account of regulations occurring between genes. Second, `sismonr` includes protein-coding as well as non-coding genes in the synthetic GRNs simulated. It is another key aspect of gene expression regulation that has been uncovered in biological systems, but is largely ignored in existing simulators. Including non-coding genes is important as (i) they entail a different dynamics of regulation, as the non-coding RNAs directly affect the expression of their targets, rather than requiring to be translated, and (ii) they can act as unobserved confounders in benchmarking studies, since they are not always measured in experimental datasets. Third, `sismonr` allows the user to define the ploidy of the simulated systems, beyond the traditional haploid or diploid situation traditionally considered. This has implications for the genetic diversity of simulated individuals, and therefore the complexity of the resulting simulations. `sismonr` makes use of a stochastic simulation algorithm, which means that it is able to simulate the absolute discrete abundance of molecules (RNAs and proteins). This is of particular interest for simulating gene regulation as some regulatory molecules can be present in very low numbers in cells. `sismonr` is available at CRAN (`https://CRAN.R-project.org/package=sismonr`) and on GitHub (`https://github.com/oliviaAB/sismonr`). A detailed tutorial has been made available online on the GitHub repository. One advantage of using R is that users have control over the different aspects of the simulation, and they can modify the properties of the simulated system. A number of plotting functionalities have been implemented, to provide an automated representation of the simulated system or of the simulation results. Overall, `sismonr` provides computational biologists with a complete toolkit for generating benchmark datasets – or even study the behaviour of existing regulatory networks – including important biological mechanisms of gene expression regulation previously left out of GRN simulators.

Next, I used `sismonr` to evaluate and compare the performance of seven popular causal inference algorithms, along with two widely-used network inference methods (Chapter 3). I generated synthetic gene expression datasets (i.e. transcripts and protein levels) for several simulation configurations. The configurations differ in the type and quantity of post-transcriptional regulation between the simulated genes. This sets the evaluation of these inference methods in the specific context of reconstructing molecular regulatory networks, which had not been done by previous benchmarking studies (Constantinou et al., 2020; Heinze-Deml et al., 2018). Moreover, by including post-transcriptional regulation, and restricting the causal inference task to only transcripts levels, I was able to assess the impact of considering only transcriptomics data on the reconstruction performance. This is an important aspect of benchmarking as many studies still rely on transcriptomics data only when reconstructing GRNs, therefore potentially missing information available in other data types such as proteomics. I found that the causal and network inference methods were not able to detect relationships between genes driven by post-transcriptional regulation. I showed by example that including additional information in the form of protein measurements could mitigate this loss of information.

Moreover, I compared the ability of the different methods to answer a number of causal queries, i.e. infer specific types of causal relationships between the genes. I found that, in line with results from previous benchmarking studies focusing on (undirected) GRN reconstruction (Marbach et al., 2012; Vignes et al., 2011), not one method outperforms the others across all situations. Rather, each method has its own strengths and weaknesses. Therefore, as has been recommended in the past, the outcome of this comparison highlights the importance of combining the results of different methods to obtain robust results. Note that as a consequence of this observation, the strategy of combining the results from different causal inference methods has been used in subsequent chapters. Interestingly, the methods that did not assume causal sufficiency, i.e. that did not make the assumption that all variables involved in the causal system are observed, did not perform substantially better than the others. This is surprising, as in the setting of gene expression regulation, limiting the observations to only transcripts levels amounts to ignoring other molecular actors (i.e. regulatory proteins). One possible explanation is that proteins do not satisfy the assumptions made by the causal inference methods about hidden confounders. Overall, the results of these performance analyses point towards a need for causal inference methods that include different omics data, in particular proteomics data, in order to reconstruct accurate regulatory networks. This comparison provides a first step towards the application of causal inference methods to the problem of reconstructing gene regulatory networks.

## 6.3 Inference of causal multi-omics networks in tetraploid potato

A second aspect of this thesis regarded the investigation of genotype-phenotype relationships in the context of tuber bruising in tetraploid potatoes. This involved for the first time assessing the genetic component of tuber bruising. However, the analysis of data arising from a polyploid organism poses an additional challenge, as specialised tools are required to analyse resulting genomics data. In a second time, I focused development of a pipeline of analysis to reconstruct a causal network of features across omics layers involved in tuber bruising. The goal was to combine multi-omics data integration with the causal inference tools previously benchmarked. Such causal analysis at a multi-omics scale is only starting to be considered (Montastier et al., 2015; Qiu et al., 2020), and more work is required to develop appropriate analyses.

Firstly, I focused on the genetic aspect of tuber bruising (Chapter 4). This involved performing a genome-wide association study to detect genomic variants associated with a number of phenotypes, in particular the response of tubers to mechanical bruising. This is a challenging step as few algorithms are available to perform GWAS for tetraploid organisms. In addition, the potato panel used presented a complex population structure, comprising full- and half-siblings as well as unrelated individuals. Moreover, the individuals of interest were selected for several traits of interest prior to the bruising experiment, therefore biasing the distribution of observed phenotypes. In consequence, particular care was taken to appropriately account for the impact of population structure during the analysis. This was done by assessing population structure using two different tools, namely STRUCTURE and DAPC. For each tool, an association model was fitted, in which the structure uncovered was included as a covariate, and their effectiveness in reducing the impact of population structure was evaluated by quantifying the inflation of resulting marker scores.

In addition, I assessed the impact of correcting for individual relatedness via the inclusion of random effects in the model, through the use of a kinship matrix. The importance of accounting for kinship in association studies for tetraploid potatoes has been previously demonstrated (e.g. Yu et al., 2006; Rosyara et al., 2016), and the results of my analysis are in accordance with these findings: the addition of the kinship matrix in the association model reduced the inflation of the resulting marker scores. However, contrary to Rosyara et al. (2016), STRUCTURE was found to be more effective than DAPC in controlling for the effect of population structure. This can be explained by the difference in the way the two populations were constructed. The individuals used in Rosyara et al. (2016) were less related, therefore the overlapping clustering as performed by DAPC was sufficient to account for the presence of sub-populations. In the present case however, STRUCTURE was more suited as it was able to uncover the dual membership of individuals to subgroups, due to the half-sibling setting. This result emphasises the need to carefully choose the tools used in accordance with the data investigated. In addition, due to the combined effects of a small number of observations and the individuals being selected for some traits of interest, very few variants reached statistical significance. I however demonstrated that even when adjusted p-values did not reach the significance threshold, variants with a high association score with the phenotype were of biological interest.

The GWAS results were then compared to a differential expression analysis performed on transcriptomics data. This allowed me to investigate potential mechanisms through which genetic variants impact the phenotype. Because the genomics data was obtained for specific regions in the genome only (through capture-bait sequencing), it was not surprising that none of the variants found associated with tuber bruising coincided precisely with genes found to be differentially expressed. However, when comparing the position of these variants with those of the genes found to be differentially expressed, it was possible to detect genomic regions in which variation at the genomic level impacted the response of tubers to mechanical bruising through their effect on key genes. Some of the highlighted genomic regions were found to be associated with tuber bruising in previous studies, but other were not, thus potentially pointing towards additional regions of interest and alternative mechanisms involved in tuber bruising. The overlap of these results with previous studies also validate the use of capture-bait sequencing in selected individuals for GWAS study. This is of particular interest to the breeding community, as it showcases the usefulness of using data directly from breeding programs, as an alternative to costly mapping populations typically used for QTL mapping studies. It also illustrates the importance of considering additional data types, such as transcriptomics data, to enrich and complement the results of an association study. Lastly, the results of these analyses show that statistical significance in GWAS must be considered with care, and even non-significant results can still provide information of interest to biologists.

Lastly, the results obtained from the association analysis were integrated with metabolomics data, in order to reconstruct multi-omics biological mechanisms involved in tuber bruising (Chapter 5). Typical single-omics analyses, e.g. differential expression and co-expression network reconstruction, were performed on the metabolomics dataset, for comparison with subsequent multi-omics analyses. Then, a multi-omics integration tool was applied to the genomics, transcriptomics, metabolomics and phenotypic data to select features across the omics datasets associated with tuber bruising and with common variations across the observations. Finding co-varying features across the omics datasets is crucial in detecting molecules involved in common biological mechanisms; this is the very added benefit of integrating different omics datasets. Indeed, when comparing selected features with results obtained from typical single-omics differential analysis, I found that the multi-omics integration tool highlighted biologically relevant features that were not detected with the single-omics analyses. I also demonstrated the importance of using the results of single-omics analyses, such as GWAS, to inform feature selection. Uninformed multi-omics feature selection resulted in retaining genomic variants because of the impact of population structure. This result provides a call to caution when integrating different omics datasets, as integration tools might not be appropriately designed to deal with biases specific to each omics dataset.

I then applied an array of causal inference methods to the selected features, in order to reconstruct a causal network of features involved in tuber bruising. The resulting causal multi-omics network consisted of many relationships between features from a same omics, with a few cross-omics relationships bridging the different layers. Some relationships could be validated using previous knowledge, as for example links found between metabolites known to be involved involved in a common pathway. However, difficulties in interpreting the resulting causal graphs arose from a lack of annotations for some of the genes (e.g. gene of unknown function) and for many metabolites. This is definitely an important aspect of network reconstruction that must be addressed. Taken together, the pieces of work of Chapters 4 and 5 provide an original approach to applying causal inference methods to a multi-omics setting, specifically in the context of uncovering regulatory mechanisms involved in a specific trait of interest. This permitted me to investigate the biological mechanisms linking variations in the genotype to changes in tubers response to bruising. It combined the added benefit of multi-omics data integration, which looks for signal across the different omics layers, and of causal inference, which goes beyond association and searches for direct (rather than indirect) relationships between molecular features. Development of such integrative analyses are urgently needed to make the most of the different datasets gathered, and this work provides a framework for data integration that makes use of statistical tools designed to detect causal inference.

## 6.4 Future work

The work presented in this thesis can be extended in several directions.

For example, in spite of the efficient Julia implementation of the Stochastic Simulation Algorithm (in the BioSimulator package), the complexity of the simulation model used by `sismonr` comes at the cost of a high computational burden. The stochastic modelling becomes computationally expensive for systems in which molecular species are highly abundant, and the number of possible biochemical reactions occurring in the simulated system is large. This can be mitigated in part by the use of approximate stochastic simulators such as the tau-leaping algorithm (Gillespie, 2001), which has been enabled in `sismonr` through the use of the Julia package `Biosimulator` (Landeros et al., 2018) implementing these approximations. I have investigated the speed of different exact and approximate versions of the Stochastic Simulation Algorithm for a few toy networks, that differ in the number of molecular species and reactions. It would be interesting to formally investigate the running time of simulations as a function of different properties of the simulated system, and to assess the gain induced by using an approximate simulator, and the resulting loss in precision of the simulations. In addition, the use of high-performance computing clusters such as NeSI, as showcased in Chapter 3, can render feasible more complex simulations, that would be unrealistic to run on personal computers. Recent work on parallelisation of stochastic simulation (e.g. Goldberg et al., 2020) could also be integrated with `sismonr` to enable the efficient simulation of large regulatory networks. Furthermore, future versions of `sismonr` could model more complex relationships between the different allelic versions of genes, such as dominance (the presence of a specific allele dominates the behaviour of the regulation) for example. Another possible avenue of improvement would be to include metabolites into the simulated networks.

The evaluation of causal inference methods presented in this work has been limited to the reconstruction of small synthetic networks (with 20 genes). It would be interesting to test if the conclusions obtained from this extensive evaluation hold for larger networks, or if the methods' performance is impacted by the size of the networks to reconstruct. For computational and time-constraints reasons, I limited the evaluation to 20 datasets per simulation configuration. The conclusions of the evaluation could be strengthened by increasing this number, for example by generating at least 100 simulated datasets per configuration. Other potential areas of investigation would be to assess how different network motifs are reconstructed by the causal inference methods, or to include different types of post-transcriptional regulations in a same synthetic network. The evaluation performed in this work provides nonetheless a pilot study that will hopefully pave the way for larger evaluation studies of causal inference methods in the context of molecular regulatory networks.

The association study performed in Chapter 4 proved that useful information can be extracted from a population issued from a breeding program. Nonetheless, the analysis could benefit from replications with more individuals, in a setting where no selection was applied. This would allow to observe a larger distribution of phenotypes, and potentially improve the detection of effects of true causal genomic variants. In addition, more replicates of each genotype could be used in the bruising experiment in order to obtain a better estimate of the tuber bruising response. Nevertheless, this study pointed towards new genomic regions potentially involved in tuber bruising, and future work could focus on investigating these regions. Also, this integration of transcriptomics data with the GWAS results provided interesting information about potential mechanisms mediating the effect of causal genomic variants on tuber bruising. Such an integrated analysis would benefit from additional biological information, for example in the form of annotation about gene functions. I used GO terms related to the genes to investigate the nature of the biological mechanisms involved. Using information from the KEGG database would also have provided important complementary information, but this was hindered by the difficulty to map the genes' Ensembl IDs to the KEGG IDs. Improvements of ID conversion tools or databases could allow similar work to benefit from previous knowledge in order to interpret their results.

Reconstructing a multi-omics causal network from genomics, transcriptomics and metabolomics data required in a first step to select features of interest across the datasets. For this task, I used the DIABLO algorithm (from the `mixOmics` package), which requires as an input the number of features to be retained from each dataset. To assess the optimal values to choose, I optimised two contradicting objectives: (i) retaining a number of features that allowed the best possible discrimination between the two phenotypic groups, and (ii) retaining a number of features small enough to enable performing causal inference on the resulting features. Future work could improve on this, by finding alternative ways to set the number of features to retain. The `mixOmics` package uses X-fold cross validation to offer an estimate of such value, however this is not available in the presence of missing data. Investigation of a workaround for this problem could be useful to researchers wanting to apply DIABLO on their own datasets that could contain missing data because of the technology used for obtaining these molecular measurements. Analysing how the reconstructed multi-omics causal graph changes with the number of features retained could also be a natural extension of this work. Finally, additional work on the identification of metabolic compounds detected will prove invaluable in improving the interpretation of the reconstructed causal network. Uncovering the identity of these compounds and mapping them to existing databases such as KEGG would shed light on the biological mechanisms at play.

## 6.5 Concluding remarks

The work undertaken in this thesis explores the integration of causal inference methods and multi-omics datasets to investigate genotype-phenotype relationships in biological systems. A simulation tool and evaluation framework have been developed in order to assess the performance of statistical methods for causal inference to reconstruct regulatory networks from observational data. This has been used to investigate the ability of causal inference methods to detect causal relationships amongst genes from transcriptomics data, in the presence of post-transcriptional regulation. The results emphasise the need to include proteomics datasets to help in the reconstruction of biological networks. In addition, different single-omics analyses, including a genome-wide association study, differential analysis and co-expression network reconstruction, were applied to genomics, transcriptomics, metabolomics and phenotypic data obtained from tetraploid potatoes, in order to shed light on the biological mechanisms of tuber bruising. These analyses were compared to a multi-omics data integration framework, which, coupled with the application of causal inference tools, led to the reconstruction of a multi-omics causal network linking molecules from the different omics datasets involved in tuber bruising.

# Appendix A

# Supplementary File for Chapter 1

## Gene regulatory networks: a primer in biological processes and statistical modelling

Olivia Angelin-Bonnet[1], Patrick J. Biggs[1,2], and Matthieu Vignes[1]

[1]School of Fundamental Sciences, Massey University, Palmerston North, 4442, New Zealand, and
[2]School of Veterinary Science, Massey University, Palmerston North, 4442, New Zealand.

*Published as*:

Olivia Angelin-Bonnet, Patrick J Biggs, and Matthieu Vignes. Gene regulatory networks: a primer in biological processes and statistical modelling, in *Gene Regulatory Networks*, pp. 347-383. Humana Press, New York, NY, 2019. `https://link.springer.com/protocol/10.1007/978-1-4939-8882-2_15`

## Abstract

Modelling gene regulatory networks not only requires a thorough understanding of the biological system depicted but also the ability to accurately represent this system from a mathematical perspective. Throughout this chapter, we aim to familiarise the reader with the biological processes and molecular factors at play in the process of gene expression regulation. We first describe the different interactions controlling each step of the expression process, from transcription to mRNA and protein decay. In the second section, we provide statistical tools to accurately represent this biological complexity in the form of mathematical models. Amongst other considerations, we discuss the topological properties of biological networks, the application of deterministic and stochastic frameworks and the quantitative modelling of regulation. We particularly focus on the use of such models for the simulation of expression data that can serve as a benchmark for the testing of network inference algorithms.

**Key words**: Gene expression regulation, Regulatory network modelling, Systems biology data simulation, Post-transcriptional regulation, Post-translational regulation, Deterministic and stochastic models, Molecular regulatory interactions

## A.1 Introduction

The different regulatory processes occurring within cells are often depicted as a network of interacting entities. These entities can be mapped onto different layers that represent the different biological molecules involved in expression regulation, for example transcripts and proteins (Figure A.1a). High-throughput studies provide us with a measurement of the variable levels of a given layer. For example microarrays or RNA sequencing technologies measure mRNA abundance, and are commonly referred to as gene expression data. We refer the interested reader to Conesa et al. (2016) for such modern data handling practices, to Auer & Doerge (2010) for associated statistical designs and to Backman & Girke (2016) for a data processing and primary analysis workflow.

From a biological perspective, entities from different layers are found to interact. Indeed, in addition to the well-known control of transcription by proteins termed transcription factors (TFs), other steps of the gene expression process are targeted by regulatory molecules beyond proteins, e.g. small molecules such as metabolites and noncoding RNAs. On top of this dynamic regulation, the information encoded in the DNA itself exerts to some extent control over the expression profile of genes. Here the term “gene” refers to a DNA sequence coding for a protein or other untranslated RNA. However it is usually impossible to measure in the same experiment data about all these molecular layers. We are therefore most of the time bound to making the most of one given data type from which we seek to extract patterns giving insight into the regulatory interactions at play. Thus gene regulatory networks (GRNs) successfully gather the detected relationships between transcripts, even if these relationships are mediated by other molecules such as proteins. GRNs represent these interactions in a graph where nodes correspond to genes (and gene products) and edges represent the regulatory relationships among them (Figure A.1b).

The modelling of such regulatory systems is an important aspect of the reverse engineering problem. Accounting for existing biological interactions can be key to a more accurate analysis of experimental data, e.g. in the analysis of differential gene expression (Dona et al., 2017). In addition, such models can be used to simulate expression data in order to assess the performances of a given network inference method, just like data can be simulated to assess gene expression differential analysis method performance (Rigaill et al., 2016). Indeed, a detailed analysis of the strengths and weaknesses of a given method can guide the choice of a practitioner to choose among the possible different reverse engineering approaches and pave the way for needed method development. A possibility is to use as a benchmark a previously studied experimental dataset, but this approach is limited by our incomplete knowledge about true – if this truth is ever an achievable objective – underlying pathways. On the contrary, the use of simulated expression data from *in silico* networks renders possible the objective comparison of the results of network inference to the true underlying interaction graph. More precisely, synthetic data allows the assessment of the impact of sample size, noise or topological properties of the underlying network on the methods performance. To make valid conclusions, one expects synthetic data to have features as close as possible to real data. Modelling such complex systems seems like a insurmountable task. However, by carefully designing each constituting element of the model it is possible to link the statistical representation of a regulatory system to the underlying biological mechanisms in a meaningful way. This is the very topic of this work.

This chapter aims at bringing together the biological and statistical representation of GRNs. In Section A.2, we provide an overview of the different regulatory mechanisms that shape the gene expression profiles. We focus on the different regulatory molecules that target each step of the expression process. In Section A.3, we introduce the reader to the basic concepts necessary to the construction of a GRN model, from the topological properties shaping biological networks to the mathematical frameworks used for the dynamic simulation of expression data and the representation of regulation from a quantitative point of view. Together, this chapter provides a first guide to GRN modelling anchored in the biological reality of gene expression regulation.

## A.2 Biological processes: from gene to protein

Proteins are the main actors in living organisms. They achieve a myriad of functions. Yet their structure, their production mechanisms and their regulation to allow the cell or organism to adequately adapt to the environment is dictated by the information contained in the genetic material of the

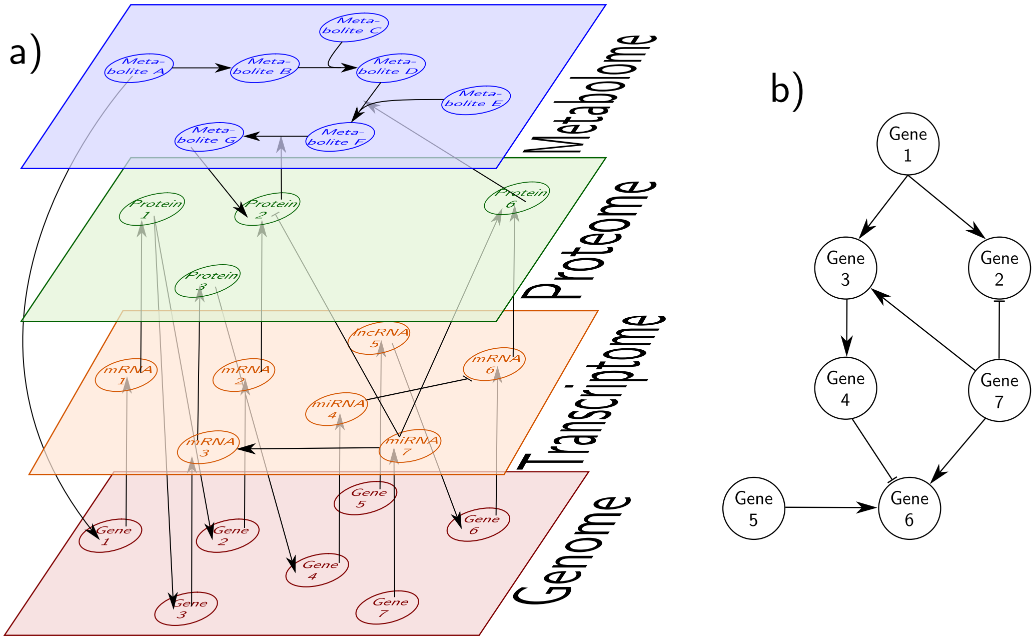


Figure A.1: Biological versus statistical representation of a GRN. a) Biological regulatory systems are complex: the different intermediary products of genes – transcripts and proteins – as well as metabolites interact in a multi-layer network. Such networks are the best representation we can give of a biological complex system. b) Statistical perspective: genes can be considered as nodes in a directed graph, where the edges represent regulatory interactions. Each parent variable node directly influences its children variables, therefore representing the regulation mechanism of a gene product on the transcription of another gene.

organism. The expression of a gene, a “coding sequence” into an active protein is a complex process involving numerous biological molecules transformed via varied reactions and interactions. The information encoded in the coding sequence of the DNA is transcribed into a messenger RNA (mRNA), which is processed and translated into a protein, according to the central dogma of biology. Once synthesized, a protein may require additional “post-translational” modifications to acquire a functional form. In this section, we aim at providing an overview of the different regulatory interactions targeting each of these steps. This knowledge certainly helps data analysts designing more *ad hoc* models to extract knowledge from modern high-throughput measurements. While it is out of the scope of this chapter to provide a detailed and comprehensive description of the specific biological mechanisms, we provide references to more biology-centred reviews of the subject. An overview of the different molecular actors of this regulation can be found in Figure 1.1.

### A.2.1 Regulation of transcription

The regulation of transcription is believed to be a key determinant of gene expression profile (Pai et al., 2015; Zlatanova & Van Holde, 2016). It mainly leverages the action of TFs which act as activators or repressors for the transcription of target genes. Regulators act by binding to proximal or distant sites on the promoter of the target genes. They impact transcription by facilitating or restraining the recruitment of the transcriptional machinery to the target gene via protein-protein interactions with its constituents. While TF binding only involves proximal promoters in bacteria, additional remote regulatory elements such as enhancers, insulators or locus control regions play an important role in the regulation of eukaryotic genes (Maston et al., 2006; Zlatanova & Van Holde, 2016). A given TF can affect the expression of one or more target genes, and its impact on gene expression (i.e. activation, repression or modulation) can change in response to a specific environmental or molecular stimulus. Typically, a TF will only regulate a few targets, but some global TFs can control transcription of large sets of genes (Balaji et al., 2006). Interestingly, while TFs play a crucial role in the control of gene expression, they are often found in low concentration, possibly only a few molecules per cell (Zlatanova & Van Holde, 2016).

Conversely, the transcription of a specific gene can be controlled by several TFs. This important feature, termed “combinatorial regulation”, provides the cell with an increased complexity in transcriptional regulation. Each gene can potentially process several inputs which dictate its resulting expression profile (Balaji et al., 2006). The different regulator molecules can act independently, if each of them affects a different aspect of the transcriptional machinery. Alternatively, TFs are often found to form complexes, either homo-dimers or hetero-dimers, thereby exerting cooperative regulation on the target (Balaji et al., 2006; Ravasi et al., 2010). Importantly, such cooperation implies that the regulation only occurs when all the components of the regulatory complex are present. Yet another mechanism of combinatorial regulation is a synergistic interaction, where the global effect of the different TFs is greater than the sum of their individual effects (Maston et al., 2006; Schilstra & Nehaniv, 2008). Finally, different TFs can compete for the same binding site on the target promoter. The respective affinity of the different molecules for the binding sequence determines which of them preferentially occupies the promoter. These affinities can be altered by environmental cues or changes in the promoter context (occupancy of neighbouring sites, etc).

In addition to protein regulators, transcription can be controlled by noncoding RNAs, whose role will be discussed later in this chapter. Furthermore, recent evidence tend to suggest that small RNAs and in particular microRNAs also play a role in transcription silencing, in addition to their impact on post-transcriptional functions detailed in the following sections (Castel & Martienssen, 2013; Catalanotto et al., 2016). Lastly, other features can affect the transcription of genes. Specifically, the methylation state of DNA, and in particular of gene promoters, has been linked to gene silencing (Gonzalez-Zulueta et al., 1995; Herman & Baylin, 2003). A widespread example is the inactivation of tumour-suppressor genes by hypermethylation as a hallmark of cancer (Jones & Baylin, 2007). DNA methylation is controlled by an array of specialized enzymes. In eukaryotic cells, the chromatin structure, that is the packaging of the DNA, affects all steps of the transcription process. This structure is dynamic and is regulated by ATP-dependent chromatin-remodelling complexes which control DNA-histones interactions, and by histone-modification factors (Li et al., 2007). Histone post-translational modifications (such as methylation, acetylation, phosphorylation and more) greatly affect the chromatin structure, notably through the recruitment of chromatin-remodelling complexes, or by directly influencing their interactions with DNA. These histone marks have been found to correlate with transcription efficiency and in some case control the access of TFs to promoters (Li et al., 2007).

### A.2.2 Regulation of translation

Following gene transcription and transcript processing, mRNAs are translated into proteins by ribosomes and associated molecules. This process is also targeted for regulation, both global and specific. The initiation of translation, that is binding of the translational apparatus to mRNAs and recognition of the translation starting site, is thought to be the step where most of the regulation occurs. As the specific mechanisms of translation initiation differ between bacteria and eukaryotes (Kozak, 2005; Zlatanova & Van Holde, 2016), regulation processes are specific to each, but similarities can be observed. Regulation of translation offers a faster modulation of the concentration of proteins compared to transcription regulation, as the former silences already existing mRNAs, while with the latter these mRNAs are still transcribed until their decay (Sonenberg & Hinnebusch, 2009).

As an example, during the response to a particular stress such as nutrient deprivation or temperature shock, cells often undergo a global decrease of their translational activity (Gebauer & Hentze, 2004; Halbeisen et al., 2008; Sonenberg & Hinnebusch, 2009). This global programming switch occurs through the control of the availability or the activity of the translational apparatus, notably through the phosphorylation state of eukaryotic initiation factors in eukaryotes. Such massive translation reduction allows a decrease of the energy demand and a reallocation of cellular resources to stress response. Specific mRNAs encoding stress-response proteins can escape this regulation via distinct mechanisms.

Alternatively to global programming, translation of mRNAs can be specifically regulated for a small set of genes via the involvement of RNA-binding proteins (RBPs) or microRNAs (miRNAs) (Gebauer & Hentze, 2004; Merchante et al., 2017). These regulatory molecules recognise and bind to specific sequences in the target transcript, mainly situated in the untranslated regions of the mRNA. RBPs mainly act through interactions with the translational apparatus, leading to the inhibition of translation (Gebauer & Hentze, 2004). miRNAs act through the RNA-induced silencing complex (RISC) complex (Hutvágner & Zamore, 2002; Valencia-Sanchez et al., 2006). The level of complementarity between a miRNA and its binding sequence on the target transcript specifies the triggered mechanism of regulation (Zlatanova & Van Holde, 2016): the extensive base-pairing between the miRNA and its target triggers the degradation of the latter, whilst partial base-pairing induces translation inhibition (Jackson & Standart, 2007). Interestingly, it has been shown that in the case of miRNA-mediated translational repression, the promoter of the target gene determines the precise mechanism of action of the miRNA (Kong et al., 2008). However the role of small RNAs are still not perfectly clear and additional processes could be discovered by further experimental studies (Wu & Belasco, 2008).

It is interesting to note that the direct impact of small RNAs on the translation of a gene can also indirectly affect other processes such as transcription of non target genes. For example, Tu et al. (2009) used miRNAs intervention experiments to detect their direct impact on TFs levels, but also reported the indirect effect of these miRNAs on the expression of theses TFs' targets. The conservation of miRNA-mRNA sequence match, particularly in the 3' untranslated regions of genes enable the identification of the miRNA-target potential pairings (Friedman et al., 2009). Conversely, evidence suggest that miRNA synthesis can also be controlled by other RNAs (Guil & Esteller, 2015). We can here again raise the concept of regulation network to start organising this knowledge. Prior interactions can be predicted to create such networks (Tu et al., 2009; Wright et al., 2014). Algorithms for RNA-RNA interaction predictions (e.g. Salari et al., 2010) are compared in Lai & Meyer (2016). Then molecular techniques can confirm the putative relationships (Engreitz et al., 2014).

The primary sequence of the transcripts also heavily influences their translation (Kozak, 2005). In particular, the formation of secondary structures within the transcript (e.g. hairpin, stem-loop, etc. which can be facilitated by the properties of the primary sequence such as GC content for example), and specifically in regions involved in translation initiation can impair the translation process. Specific structural features, such as upstream open reading frames or internal ribosome entry sites can also impact translation. The detailed features of such mechanisms are beyond the scope of this chapter, and we refer the reader to Kozak (2005). However, being aware of the existence of these mechanisms can allow the modeller to include them or at least discuss their effect on the outcome of an analysis. In this vein, Liang & Li (2007) postulate that protein-protein interactions are linked to the regulation of the corresponding genes by miRNAs.

Lastly, an interesting mechanism of translation control is the regulation via "riboswitches" (Biggs & Collins, 2011; Henkin, 2008; Serganov & Patel, 2012). A riboswitch is a regulatory sequence within mRNAs which responds to specific cues, namely temperature or the presence of particular metabolites. Thermo-sensors are a class riboswitches that respond to temperature by changing their conformation, therefore modifying the translation rate of the transcript. Alternatively, riboswitches can detect and link to specific metabolites. This provokes a modulated translational activity of the transcript via the modification of the mRNA conformation.

### A.2.3 Regulation of mRNA decay

In addition to the elimination of defective mRNAs arising for example from transcription or splicing errors, fully functional mRNAs are subject to spontaneous or targeted degradation. Regulation of mRNA decay plays an important role in the resulting transcript level. Specific degradation of transcripts can be mediated by RBPs, or by small RNAs, namely miRNAs and small interfering RNAs (siRNAs) (Halbeisen et al., 2008). Interestingly, mRNAs encoding functionally-related proteins were shown to exhibit correlated half-lives. This phenomenon suggests a common regulation of mRNAs involved in similar biological processes (Wang et al., 2002; Yang et al., 2003).

RBPs recognise and bind to specific sequences in mRNAs. It allows them to trigger the recruitment of decay factors, ultimately leading to target degradation. Alternatively, some RBPs have been found to stabilize their targets, protecting them from degradation (Kuwano et al., 2008). Just as factors regulating other aspects of gene expression, RBPs can interact to exert combinatorial control over mRNA decay rates.

Alternatively, miRNAs and siRNAs can promote the decay of target mRNAs, via interactions with RISC and possibly with other RBPs. Such a phenomenon is coined RNA interference or RNAi (Mattick & Makunin, 2006.; Valencia-Sanchez et al., 2006). As mentioned previously, such degradation is promoted by the perfect pairing of the small RNAs with the target transcript. Several mechanisms can be involved to trigger target degradation. Possibly, interactions with small RNAs and RISCs promote the endonucleolytic cleavage of the transcript. Another explanatory mechanism is that the target can be directed to P-bodies, which are small cytoplasmic granules containing RNA degradation machinery (Valencia-Sanchez et al., 2006). Targeted mRNAs are locked in these P-bodies and consequently degraded before they can be further processed, e.g. translated.

### A.2.4 Regulation of protein activity

After translation, proteins are sometimes subjected to additional modifications to acquire their fully functional state. These changes can be irreversible, i.e. proteolytic cleavage of the peptidic precursor to obtain a functional protein (Cooper, 2000; Zlatanova & Van Holde, 2016). Alternatively, the cell can modulate the activity of its protein pool via a number of reversible post-translational modifications. A common mechanism is the modification by specialised enzymes of some amino acids on the protein, such as phosphorylation, oxidation or acetylation (Walsh et al., 2005). In particular, phosphorylation is a common mechanism for the activation of enzymes, TFs or other proteins. It is used in signalling pathways to relay extracellular messages to the nucleus, via a cascade of phosphorylation which activate kinase proteins (Hunter, 1995; Lizcano & Alessi, 2002). The endpoints of such pathways are generally TFs, whose phosphorylation lead to their activation and relocalisation into the nucleus where they can modulate the expression of appropriate response genes.

Taking again the example of signal transduction in the cell, the cascade of phosphorylation is initiated by the activation of membrane receptors, that detect a particular signal in the environment – generally a vitamin, a hormone, or another metabolite. This specific ligand binds to the receptor peptide, and triggers conformational changes to lead to the activation of the receptor. Such activation prompted by the binding of a small molecule is also frequently found in metabolic pathways, as a mean to regulate the production of a specific compound (Cooper, 2000). Metabolites can bind to the enzymes responsible for their synthesis in a feedback loop that auto-regulates enzyme activity according to the abundance of this specific product. Conformational changes resulting from ligand binding can mask or reveal the catalytic site of the enzyme, thereby controlling its ability to bind with its substrates.

Lastly, peptidic chains sometimes need to assemble into multimers, to form a functionally active molecular complex (Cooper, 2000). Such protein complexes can be composed of several copies of the same protein, or of different proteins. In the latter case, the abundance of the complex, and hence its activity, is limited by the least abundant species. It is an interesting mechanism of regulation of the complex activity. Information about interactions among subunits can be found in protein-protein interaction databases (see for example Szklarczyk et al., 2017).

### A.2.5 Regulation of protein decay

Cells possess several pathways for the degradation of proteins. A first mechanism is concerned with the degradation within lysosomes, which is a non-specific process, notably solicited in response to nutrient starvation as a rapid source of amino acids (Olson & Dice, 1989). In addition, proteins can also be specifically tagged to degradation, via conjugation of a ubiquitin chain to the target peptide (Lecker, 2006; Varshavsky, 2005; Zlatanova & Van Holde, 2016). Tagged proteins are recognised by cellular machineries termed 26S proteasomes and subsequently degraded. This ubiquitin-proteasome pathway provides the cell with a way to rapidly control a regulatory process by degrading its effectors. It is notably involved in the regulation of transcription via degradation of specific TFs (Lecker, 2006).

The addition of ubiquitin on target proteins is mediated by the E1, E2 and E3 enzymes. The different isoforms of the E2 and especially E3 family confer a great specificity to this process, as each isoform can recognise different substrates. Additionally, some structural properties of proteins can impact their affinity as substrate for the ubiquitin-proteasome pathway. For example, a member of the E3 family recognises particular amino acids at the N-terminal position of proteins, in what is call the N-end rule pathway (Lecker, 2006). The nature or accessibility of specific residues can also impact the ability of ubiquitination enzymes to recognise and tag target proteins.

It is interesting to note that the ubiquitin-proteasome pathway is able to degrade only a subunit of a given protein, for example to produce a functionally active product or on the contrary inactivate the protein. This is the case for the NF-$\kappa$B TF, which is bound by its inhibitor, I$\kappa$B (Varshavsky, 2005). In response to a specific signal, the complex is ubiquitinated, and the proteasome cleaves the I$\kappa$B, thereby freeing the TF, which, in turn, is relocated into the nucleus to trigger the required cellular response.

Quantitative measurements have highlighted the coupling between synthesis and decay rates of proteins. As for transcripts, these parameters seem to be correlated among proteins intervening in common complexes or functions. It appears that proteins involved in housekeeping functions are relatively stable, with a high production rate, leading to high concentrations in cells. On the contrary, regulatory proteins tend to be less synthesised and more rapidly degraded. This is consistent with the observation that they are often found in a low concentration in the cell (Belle et al., 2006; Vogel & Marcotte, 2012).

### A.2.6 The role of genetic variation

In addition to diverse cellular molecules which perform a wide range of regulatory activities, the DNA sequence itself plays a role in regulating gene expression. Regulatory sequences present in the promoter region of genes or in the transcribed or translated sequences dictate the set of molecules and complexes that control the expression of these genes. These sequences target transcripts or corresponding proteins for particular regulatory mechanisms. Their affinity for regulators control the strength of this regulation. The impact of genetic variation on gene expression has been studied, notably via expression quantitative trait loci (eQTL) studies. eQTLs are genomic regions within which genetic variability is associated with variation in the abundance of a particular transcript (Gilad et al., 2008). More generally genetical genomics studies (also termed cellular genomics) (Gaffney, 2013; Jansen & Nap, 2001) analyse how polymorphisms lead to variation in molecular traits, such as mRNA, protein or metabolite profiles.

Using additional genomics data such as DNA methylation state or chromatin accessibility, researchers are now focusing on identifying the specific mechanisms which relate genetic variants to response molecular traits. At the transcript level, evidence tends to show that eQTLs lead to transcript abundance variability mainly via their impact on TF binding (Albert & Kruglyak, 2015; Gaffney, 2013; Veyrieras et al., 2008).

Polymorphisms at these loci also affect other aspects of transcription, but it is yet to be determined if it is a direct consequence of genetic variation or merely an indirect effect of variation in TF binding efficiency (Pai et al., 2015). Some polymorphisms have also been shown to affect mRNA degradation, notably through modification of miRNA binding sites, or other post-translational mechanisms (Gaffney, 2013; Pai et al., 2015). In a groundbreaking effort, Bessière et al. (2018) discovered instructions encoded in the sequence itself to regulate gene activity. The nucleotide composition can be directly read to accurately decipher biological mechanisms.

### A.2.7 An example: long noncoding RNAs

After this review of the possible interactions regulating the different aspects of the gene expression process, we now turn our attention to a specific class of regulators whose role in the different biological processes mentioned earlier is just starting to be appreciated. Indeed, the functional importance of long noncoding RNAs (lncRNAs) was only hinted at when experimental studies of genome-wide transcription in cells revealed that a large fraction of the genome is transcribed, even if only a small amount actually encodes proteins (Quinn & Chang, 2016; Rinn & Chang, 2012). This discovery shook the traditional central dogma of biology stating that RNAs' primary role is to serve as messengers to produce functionally active proteins. On the contrary, as highlighted above, noncoding RNAs are now known to play important regulatory roles. While we now have a fair understanding of small noncoding RNAs (e.g. miRNAs, siRNAs, etc) and the associated biological processes, lncRNAs (exceeding 200 base-pairs, as an arbitrary defining threshold) remain for most of them *terra incognita*. In particular, the extent of their functional role is yet to be determined, and there is still debate about whether the RNA molecule itself has a functional role or if only the physical changes triggered by its transcription (e.g. chromatin opening, helix unwinding, etc) impacts the transcription of neighbour genes while the produced transcript is useless (Wang & Chang, 2011; Zlatanova & Van Holde, 2016). This is notably due to the fact that their primary sequence is less conserved than those of protein-coding genes (Mercer et al., 2009; Quinn & Chang, 2016). Nonetheless, experimental studies put us on the track of lncRNA involvement in a great variety of biological processes, from regulation of gene expression and chromatin state to genomic imprinting, in particular X chromosome silencing (Ponting et al., 2009; Quinn & Chang, 2016; Rinn & Chang, 2012). In addition, characteristic features of RNA make them well-suited for regulatory functions: their fast kinetics with no need for translation and their rapid degradation is particularly convenient for a fast and transient response to external stimuli. Moreover, their ability to bind DNA and RNA allows them to interact with both genes and transcripts (Geisler & Coller, 2013; Wang & Chang, 2011). In this section, we briefly present the diverse roles played by lncRNAs in the regulation of gene expression. For a more thorough review about the biological roles of lncRNAs, we refer the reader to the review by Geisler & Coller (2013).

One of the primary focuses of early studies about lncRNAs was their involvement in chromatin modelling (Rinn & Chang, 2012), ultimately resulting in the modulation of gene expression. lncRNAs can act as scaffold which bring together different chromatin-remodelling proteins and to assemble them into a functional complex (Rinn & Chang, 2012; Wang & Chang, 2011). Alternatively, they can guide such proteins to a target location, triggering changes in chromatin structure (Mercer et al., 2009). An interesting mechanism of action follows the transcription trail of the noncodingRNA which influences the chromatin state. It consequently impacts the transcription of neighbouring genes (Geisler & Coller, 2013).

lncRNAs can also enhance or repress the initiation of transcription via interactions with the basal transcriptional machinery. For example as a response to heat shock, the interaction of a specific lncRNA with RNA polymerase II triggers the inhibition of target genes (Geisler & Coller, 2013; Mercer et al., 2009). Additionally, lncRNAs can target TFs and modulate their activity, by directly prompting conformational changes, by recruiting TFs onto the target promoter, or by withholding the TFs away from their targets (Zlatanova & Van Holde, 2016).

lncRNAs are also involved in other steps of the gene expression process. In particular, they can influence mRNA processing, in particular mRNA splicing and editing (Geisler & Coller, 2013; Mercer et al., 2009). Some lncRNAs can impact translation efficiency of their target transcripts, through mechanisms which are still not totally clear (Geisler & Coller, 2013). They also putatively control RNA stability, either by recruiting specialised degradation machinery to the transcript, or by competing with miRNAs for binding sites. In the latter case, lncRNAs play a protecting role for mRNAs in preventing or delaying their degradation. For example, they can lure miRNAs to competitively bind to the same targets (Geisler & Coller, 2013; Wang & Chang, 2011). Finally, lncRNAs can assist in protein binding to modulate their activity. Target proteins can be TFs, chromatin remodellers, or other regulatory molecules.

While a few well-studied lncRNAs provide evidence for a functional role of these transcripts, a lot remains unknown about them. In particular, it is important to keep in mind that the functional roles described above apply to a few number of characterized lncRNAs, and it is possible that a fraction of these transcribed noncoding genes are the result of transcriptional noise or experimental artefacts (Ponting et al., 2009). The modeller has the choice to include such information for a few annotated lncRNAs only, or to include the different putative roles, e.g. in a Bayesian framework.

As demonstrated throughout this section, the expression of genes is subject to a tight regulation from which arises great biological complexity. We now embrace the point of view of the statistical modelling of such biological systems. In particular, we discuss the different aspects of the construction of a model that must be carefully thought out in order to faithfully describe the biological processes under study.

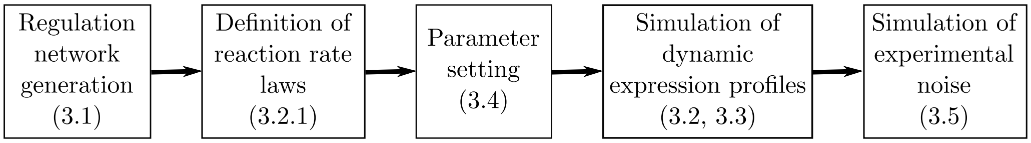

Regulation network generation (3.1)
Definition of reaction rate laws (3.2.1)
Parameter setting (3.4)
Simulation of dynamic expression profiles (3.2, 3.3)
Simulation of experimental noise (3.5)

Figure A.2: The different steps of an algorithm for expression data simulation. Each of these steps will be detailed in the referred sections.

## A.3 Modelling gene expression

Statistical models of GRNs aim at reproducing biological systems from a mathematical perspective to permit, *inter alia*, the simulation of their dynamical behaviour. A number of models for the simulation of expression data have been proposed in the last decades; an overview of the principal algorithms and their key features is presented in Table A.1. However, the design of such simulation tools is far from trivial. First and foremost, the model must be a faithful representation of the biological system, from the general topology of the underlying regulatory network to the quantitative regulation exerted on the genes and gene products. In addition, different mathematical frameworks can be used for the dynamic simulation of expression profiles, each of them carrying its own set of assumptions and limitations.

With all these considerations in mind, the next section provides a thorough reflection on the different features to examine when building a simulation network as well as a contrast of existing methods to simulate data from an *in silico* network. The general pipeline for the construction of a simulation algorithm can be found in Figure A.2.

### A.3.1 Topological properties of regulatory networks

A crucial step in simulating expression data from regulatory networks is the selection of a network topology that defines the interactions among the molecules. In a graphical representation of a GRN, nodes typically represent genes and their products, while edges correspond to regulatory interactions between molecules. Edges carry the direction of the regulation, that is, which nodes are regulators and which nodes are target molecules. The choice of the topology of a GRN is by no means an easy task. A first and simple approach for network modelling is to represent regulatory networks as random networks (Erdös & Rényi, 1959; Kauffman, 1969) (also termed Erdös-Rényi graphs) in which each pair of nodes has the same probability of being connected. This model was and is still used for expression data simulation. However topological analysis of pathways recovered from model organisms highlighted the existence of specific structural properties among biological networks, owing to the evolutionary constraints that shaped them. Algorithms for the generation of synthetic networks with similar properties where developed to construct more realistic models of biological systems. Interestingly, a number of these properties are shared with non biological systems such as the Internet or social networks (Barabási & Oltvai, 2004):

- **Small-world property**: Networks are characterized as small-world if their average path length[1] between any two nodes is small. It has been shown that most biological networks exhibit such a property (see for example Jeong et al., 2000; Albert, 2007; Wagner & Fell, 2001). This implies that components of biological networks are easily reachable from any other node, which allows

---

[1] The path length between a pair of nodes is defined as the length of the shortest path connecting the two nodes.

a rapid response to stimuli or perturbations (Albert, 2007). Synthetic random small-world networks are also referred to as Watts-Strogatz networks (Watts & Strogatz, 1998).

- **Scale-free property**: When studying the in- and out-degree distribution of (directed) biological networks, i.e. the the number of incoming and outgoing edges respectively, it has been noted that this distribution can often be modelled with a power-law distribution (Albert & Barabási, 2000; Barabási & Albert, 1999). More specifically, the probability of a node to exhibit $k$ edges is $P(k) \propto k^{-\lambda}$. An implication is that the majority of nodes interact only with a few partners, while a small number of nodes, called hubs, are highly connected. Metabolic pathways were shown to have this property (Jeong et al., 2000; Wagner & Fell, 2001), and so were GRNs (Featherstone & Broadie, 2002). For both types of networks, the scale parameter $\lambda$ usually ranges between two and three (Barabási & Oltvai, 2004; Ravasz et al., 2002). However recent findings suggest that for some organisms the in-degree distribution of transcriptional networks is not scale-free, as detailed below. An algorithm for generating random scale-free networks has been proposed by Albert & Barabási (2000). Bollobás et al. (2003) presented a directed version of scale-free networks, where both the in- and out-degree distributions are power laws, with possibly different $\lambda$ coefficients.

- **Exponential distribution of the in-degree distribution** (for transcriptional networks): alternatively to the scale-free property, studies (e.g. Guelzim et al., 2002; Balaji et al., 2006) suggested that the in-degree distribution of GRNs for some organisms is better fitted by an exponential distribution, i.e. $P(k) \propto \frac{1}{\lambda}e^{-\frac{k}{\lambda}}$. This implies that genes are regulated only by a few (generally up to three) TFs (Barabási & Oltvai, 2004), a more plausible configuration in biological networks.

- **Modularity**: real networks have a tendency to form groups of highly interconnected nodes, referred to as modules. This modular organization is characterized by a high average clustering coefficient (Wagner & Fell, 2001; Watts & Strogatz, 1998). The clustering coefficient $C$ of a node is a measure of the degree of connectivity among the direct neighbourhood of this gene. This property is important for biological systems, as it implies that biological networks are organised into relatively independent modules that each perform a distinct biological function. While inside a module the components are tightly linked, modules are only weakly connected with each other. This last property ensures to some degree robustness to the network, as disruption in one module is less likely to severely impair the rest of the network (Barabási & Oltvai, 2004). Methods to identify modules within pathways (e.g. Sanguinetti et al., 2008) could clearly inform gene network inference and this information should not be ignored when exploitable.

- **Hierarchical organization**: contrary to random or scale-free networks for which the average clustering coefficient decreases with the number of nodes in the network, biological networks are characterized by a system-independent average clustering coefficient (Ravasz et al., 2002). Moreover, the clustering coefficient of a node is a function of its degree, since: $C(k) \propto k^{-1}$ (Ravasz et al., 2002). This last property is a characteristic of a hierarchical organization of the network. This important mathematical concept allows us to reconcile the scale-free property and modular nature of biological systems. Indeed, it stresses that nodes of low connectivity tend to be found in clusters, while hub nodes constitute the junction between modules. It is to note that hub nodes will less likely be connected to each other.

- **Over-representation of network motifs**: another important feature of biological networks is the abundance of small regulatory motifs (Milo et al., 2002; Shen-Orr et al., 2002), that are recurring and non-random building blocks of the global topology (Zhu & Qin, 2005). They confer specific advantages to the system by encoding well-defined local dynamic behaviours in response to perturbations, for example buffering intrinsic stochasticity or on the contrary amplifying an external signal to trigger a cellular response (Alon, 2007). One well-known example is the negative feedback loop, in which the product of a gene regulates its own transcription (Rosenfeld et al., 2002). This auto-regulation feature allows the control of the natural fluctuation in the concentration of the gene product, as its synthesis is directly coupled to its abundance (Alon, 2007). Another famous example is the feed-forward loop, who can simultaneously process two different stimuli and whose output depends on the nature (activation or repression) of the regulatory interactions composing the motif (Mangan & Alon, 2003). A detailed quantitative analysis of such motifs and the advantages they provide to the system can be found in the book Alon (2006).

As pointed out by Pržulj et al. (2004)], such studies are based on our current and incomplete knowledge of biological networks. Despite this limitation, algorithms for the generation of graphs mimicking these structural properties have been proposed, for a more accurate representation of biological networks. In addition to the three most commonly used Erdös-Rényi, Albert-Barabási and Watts-Strogatz networks, Haynes & Brent (2009) implemented a method for simulating topologies with scale-free out-degree distribution and any desired in-degree distribution. Di Camillo et al. (2009) proposed a hierarchical modular topology model that generates networks displaying scale-free degree distribution, high clustering coefficient independent of the network size, and low average path length. However, to offer flexibility in the simulation, most simulators offer as an option for the user to choose among the different network topologies cited above. It becomes therefore possible to assess the impact of the underlying topological properties on the performances of a given network inference algorithm.

The main drawback of *in silico* networks is that none of the aforementioned network simulation methods are able to simultaneously reproduce all characteristic features of real networks (Bulcke et al., 2006). Another approach for graph generation has hence been proposed. It relies on the use of real biological networks determined experimentally. They are used as seeds from which sub-networks are sampled. Bulcke et al. (2006) proposed two sampling approaches: the cluster addition method and the neighbour addition method. Building up on this idea, Marbach et al. (2009) further refined the approach by forcing the preferential inclusion of modules in the sampled sub-networks. This module extraction method ensures a fair representation of network motifs in the generated topology, as observed in biological networks. Such an approach of sampling from real networks ensures a more faithful picture of biological pathways. Again, this is contrasted by Bulcke et al. (2006), as the real network sampling strategy relies on our current knowledge of regulatory networks, which is still incomplete, and probably biased towards well studied pathways. Lastly, Haynes & Brent (2009) pointed out that networks generated from the same source network may not be "statistically independent" as they may overlap and thus provide redundant information. It is particularly true when sampling a large number of subnetworks from a single source, as most of them will share common nodes and interactions.

### A.3.2 Mathematical frameworks and regulation functions

Once the network topology is set, the next step for data simulation is to decide on the mathematical framework to be used to compute the profiles of gene expression, which will impact the choice of regulation rules for the system. It is important to carefully consider the different options, as each formalism carries a number of underlying assumptions about the represented system. Moreover, different levels of precision about the system can be integrated. Choices depend on many factors to achieve a balance between the level of required details to make the simulations more realistic, and the computational efficiency desired. While it is not our goal to offer an extensive comparison of all the possible formalisms, we emphasize here the difference between the two mainstream formalisms in existing simulators of expression data: the continuous-and-deterministic and discrete-and-stochastic frameworks. We present the basic concepts of these approaches, and highlight the different hypotheses about the biological system underlying each model. For a more detailed and mathematically-centred review of these and other formalisms we refer the reader to de Jong (2002) and Higham (2008).

#### The continuous and deterministic approach

The continuous and deterministic approach is particularly suited to simulate data that resemble those resulting from a transcriptomics (or other 'omics) experiment. The output is a series of continuous variables, as opposed to cruder logical models that predict the activation state of each gene as a binary outcome. In such deterministic models, biological molecules are represented as time-dependent continuous variables. Typically $x_i(t)$ represents the concentration of entity (or species) $i$ at time $t$. Variation in the concentration of a species over time is assumed to occur in a continuous and deterministic way. Such changes are modelled as differential equations, in the form of:

$$\frac{dx_i}{dt} = f_i(\mathbf{X}), \tag{A.1}$$

where $\mathbf{X}$ refers to the state of the system (that is the concentration of all the species present in the system), and $f_i$ represents the change in the concentration of species $i$ as a function, often non-linear, of the global state of the system. More specifically, in the case of a chemical species and associated reactions, $f_i(\mathbf{X})$ can be written as:

$$f_i(\mathbf{X}) = v_i(\mathbf{X}) - d_i(\mathbf{X}), \tag{A.2}$$

where the vector $v_i(\mathbf{X})$ represents the synthesis rate for species $i$, while $d_i(\mathbf{X})$ models its decay rate (due to degradation, dilution, use as a reactant, etc.). Both rates are themselves expressed as a function of the system state. The ensemble of reactions occurring in the network provides a set of coupled differential equations that describe the evolution of the state of the system through time. Except for simple networks, with only a few molecules and/or linear interactions, an analytical solution is often intractable. It is however possible to integrate the model in order to compute a numerical solution. A plethora of differential equation system solvers are available in different programming languages (for example the `deSolve` package for R, the `dsolve` function of the Symbolic Math toolbox for Matlab or Berkeley Madonna).

Regulatory molecules for species $i$ appear on the right-hand side of Equations (A.1) and (A.2). They impact the abundance of their target species $i$ via regulation of the different expression steps, as we discussed them in Section A.2. The regulation of a particular reaction is hence modelled via a reaction rate law, which dictates how the rate of the regulated reaction (e.g. $v_i$ or $d_i$) evolves with the concentration of the different regulatory molecules. The vast majority of proposed simulation algorithms focus on the representation of transcription regulation, but similar consideration can be applied to any type of regulation (translation, degradation…). Two important features must be considered in order to fully characterize a reaction rate law: (i) the quantitative relation between a regulator abundance and the resulting reaction rate, and (ii) the combination scheme of the individual effects of different regulators on a common target. In the following we will discuss these two aspects, using the example of the regulation of gene transcription.

We first consider gene $i$ whose transcription is regulated by a single molecule $j$. Several choices are possible with regard to the resulting effect of the regulator abundance on the transcription rate. A simple approach is to consider that the regulation effect increases linearly with the regulator concentration (D'haeseleer et al., 1999; Yeung et al., 2002) (Figure 1.6 a), as follows:

$$f_i(x_j) = \beta \cdot x_j \tag{A.3}$$

where $f_i$ represents the transcription rate law of gene $i$, which depends on the concentration of the regulatory molecule $x_j$. In addition to the fact that this representation ignores any saturation effect arising from the limited amount of cellular resources and the maximum possible number of simultaneous transcription events, such a linear relationship can produce concentration values out of the plausible range of abundance encountered *in vivo*, possibly leading to infinitely large populations, which is biologically irrelevant. It is hence important to construct biologically credible reaction rate laws that result in realistic regulation strength and concentration values.

Alternatively, a Hill function (Figure 1.6 b) can be used to model the impact of an activator on the gene transcription:

$$f_i(x_j) = \frac{x_j^{n_{ij}}}{x_j^{n_{ij}} + K_{ij}^{n_{ij}}}, \tag{A.4}$$

or, for a repressor:

$$f_i(x_j) = \frac{K_{ij}^{n_{ij}}}{x_j^{n_{ij}} + K_{ij}^{n_{ij}}}, \tag{A.5}$$

where $K_{ij}$ represents the concentration of regulator $j$ required to obtain a half-maximum effect on the transcription rate of gene $i$, and $n_{ij}$ controls the steepness of the regulation. It must be noted that $K_{ij}$ must be non negative as it accounts for a concentration, and $n_{ij} \geq 1$. Indeed, when $n_{ij} = 0$ the resulting reaction rate law is constant. This sigmoid function accounts for the saturation of the regulatory effect: after the regulator concentration has reached a certain level, any further increase in this concentration will only result in a minimal change in the transcription rate. Additionally, tuning the parameter $n_{ij}$ enables to represent a variety of regulation behaviours, from a quasi-linear ($n_{ij}$ small) to a step-like ($n_{ij}$ high) function. Furthermore the use of a Hill function law can be justified by a thermodynamic model of the binding of TFs on the target promoter (Ackers et al., 1982; Bintu, Buchler, Garcia, Gerland, Hwa, Kondev, et al., 2005b). Considering that the mean transcription rate of a gene is proportional to the saturation of its promoter by TFs, the effect of the regulator can be further refined as:

$$f_i(x_j) = \alpha_0 \left[ 1 + (FC_{ij} - 1) \frac{x_j^{n_{ij}}}{x_j^{n_{ij}} + K_{ij}^{n_{ij}}} \right], \tag{A.6}$$

where $\alpha_0$ represents the basal transcription rate of the target gene in absence of the regulator, and $FC_{ij}$ the maximum fold-change[2] of gene expression induced by the regulator. Details of this computation can be found in the Supporting Information of Marbach et al. (2010). It is straightforward to deduce

---

[2] The fold-change of a gene is defined as the ratio of its transcription rate in the presence of a high concentration of regulatory molecules over its transcription rate in absence of regulator. From equation (A.6) it is easy to see that the transcription rate of gene $i$ tends towards $\alpha_0 \cdot FC_{ij}$ when $x_j$ becomes large, hence the fold-change tends to $\frac{\alpha_0 \cdot FC_{ij}}{\alpha_0} = FC_{ij}$.

the transcription rate law for a gene whose expression is controlled by an inhibitor, as the resulting maximum fold-change is $FC_i = 0$:

$$f_i(x_j) = \alpha_0 \left[ 1 - \frac{x_j^{n_{ij}}}{x_j^{n_{ij}} + K_{ij}^{n_{ij}}} \right] \tag{A.7}$$

Such representation is massively used in simulator algorithms, although with some variations (Bulcke et al., 2006; Hache, Lehrach, et al., 2009; Haynes & Brent, 2009; Mendes et al., 2003; Pinna et al., 2011; Roy et al., 2008; Schaffter et al., 2011). For example, the transcription rate law represented in Equation (A.6) can be adapted to account for a gene that is not expressed in the absence of its activator.

Taking the approximation that $n_{ij} \to \infty$, it is possible to simplify this Hill function, and model the transcription rate law as an on-off switch, where the maximum effect of the regulator on the transcription rate occurs as soon as the regulator molecule level exceeds a certain threshold. Below this threshold, no regulation is observed. Such representation is described by a step function (Figure 1.6 c):

$$f_i(x_j) = \begin{cases} \alpha_0 & x_j < K_{ij}, \\ \alpha_1 & x_j \geq K_{ij}. \end{cases}$$

$\alpha_0$ can be set to zero, to model the case of a gene that is not expressed in absence of its regulator. This simplification provides the basis for piecewise differential equations (de Jong, 2002). It allows us to model a non-linear interaction even if the kinetics of the regulation are not known in detail.

Once the quantitative effect of a regulator has been chosen, one must consider the overall effect of several regulators targeting a common gene. Indeed, different combinatorial regulations can be modelled. A simple example is to assume that different regulators impact the expression of the target gene independently of each other. This approach has been used by Mendes et al. (2003). For an independent combinatorial effect model, the resulting regulation effect of all regulators is equal to the product[3] of the individual effects of each regulator on the transcription rate.

Alternatively, the different TFs can assemble into a complex that will bind to the target promoter to regulate transcription. In this case, the resulting regulation effect will be limited by the least abundant regulator species. An example can be found in Roy et al. (2008), in which different TFs can

---

[3] The use of the product, rather than the sum, ensures that if the concentration of a repressor is high enough to silence the gene (resulting in an individual effect close to zero) the overall transcription rate will also tend to zero regardless of the quantity of activators present. It also implies that the overall fold-change obtained for large quantities of the different activators is the product of the fold-changes individually induced by each activator, which is justified thermodynamically in Bintu, Buchler, Garcia, Gerland, Hwa, Kondev, et al. (2005a) and Bintu, Buchler, Garcia, Gerland, Hwa, Kondev, et al. (2005b).

assemble into cliques which in turn can form TF complexes regulating the target gene. The resulting translation rate law is therefore equal to zero as soon as the concentration of one of the TFs reaches zero, as it is then not possible to form a functional complex.

An interesting approach has been proposed by Di Camillo et al. (2009). It uses fuzzy logic to represent the possible combinatorial interactions between different regulators. The advantage of such an approach is that it combines the Boolean logic functions (AND, OR, NOT, etc.) well suited to describe combinatorial behaviour with continuous regulation, as the output of fuzzy logic functions is a continuous value. Given a continuous input, that is the concentration of each regulator, the fuzzy logic applies a set of functions such as $\min$, $\max$, or $\sum$ (sum) to output the level of regulation commonly achieved by the different regulators. Di Camillo *et al.* hence represents "cooperation" (for which the regulation is only achieved in the presence of all the required regulators) as a $\min$ function applied to the set of regulator concentrations. Similarly, synergistic behaviour, direct inhibition or competition are modelled with fuzzy logic functions.

Deterministic models are traditionally used for the simulation of expression data(Bulcke et al., 2006; Di Camillo et al., 2009; Hache, Wierling, et al., 2009; Haynes & Brent, 2009; Marbach et al., 2010; Mendes et al., 2003; Roy et al., 2008) (see Table A.1). However, despite its broad use, the deterministic formalism presents several limitations, which relates to the underlying hypotheses about the biological system depicted. In particular, the assumption of continuous change in species concentration is only valid for a macroscopic description of biological systems (de Jong, 2002), i.e. when the number of molecules in the cell is large enough so that species concentrations can be considered to vary continuously when a discrete number of molecules is actually added/withdrawn from the system. When the abundance of a species reaches low values (defined as a thousand or less by Cao & Samuels, 2009), this assumption does not hold anymore, and it is more correct to represent this abundance by a discrete molecule count. Moreover, the deterministic assumption can be questioned, particularly for small systems, given the fluctuation in the timing of biochemical reaction events (de Jong, 2002). Indeed two identical genes with the same transcription rate will not produce exactly the same number of transcripts during the same time period, due to the apparent stochasticity of biological events. While this fluctuation can be averaged out for highly abundant species, it is more difficult to ignore it for species with only a few molecules per cell. As numerous studies have underlined the importance of stochasticity in biochemical systems (McAdams & Arkin, 1999; Ross et al., 1994; Wilkinson, 2009, 2012), it can be preferable to explicitly model stochasticity in the simulation.

### The discrete and stochastic approach

To overcome the limitation of continuous and deterministic models, in particular for modelling small systems, a discrete and stochastic representation of biological systems has been proposed. It must be noted that even if the continuous and deterministic approach and the discrete and stochastic framework are commonly referred to as respectively deterministic and stochastic models, there exists representations of biological systems that are either discrete and deterministic (e.g. Boolean networks) or continuous and stochastic (e.g. Chemical Langevin Equation, discussed later). One must hence keep in mind that continuous (resp. discrete) does not necessarily imply deterministic (resp. stochastic) as the first terms refer to the representation of species abundance while the latter corresponds to the variation of the system state.

In the discrete and stochastic framework, the state of the system corresponds to discrete values accounting for the number of molecules of each species present in the system. While the vast majority of deterministic approaches are species-centred, i.e. one differential equation represents the evolution of one species abundance through time, stochastic models often rely directly on the biochemical reaction formalism. These reactions can be schematically represented in the form:

$$\text{Substract 1} + \text{Substract 2} \xrightarrow{\text{Reaction rate}} \text{Product 1}$$

Or, in the context of gene expression:

$$\text{Promoter} + \text{TF} \xrightarrow{c_1} \text{Active\_promoter}$$
$$\text{Active\_promoter} \xrightarrow{c_2} \text{Active\_promoter} + \text{mRNA}$$

Each reaction is characterized by:

- A stoichiometry vector $v_j$ which represents the change in abundance of the different species resulting from one firing (i.e. one occurence) of the reaction. Negative and positives indices correspond respectively to reactants and products of the reaction.

- A propensity function $a_j(\mathbf{X})$, with $a_j(\mathbf{X})\tau$ representing the probability that the reaction will occur in the next time step $[t, t+\tau]$ given the system state at this time $t$. The rate $a_j$ depends on the current state of the system. If $c_j$ is the constant probability that one molecule of each of the $r$ reactant species $S_i$, $1 \leq i \leq r$ collide and undergo the reaction in the next time unit, then $a_j = c_j . \prod_r x_r$. Generally the number of reactants per reaction is limited to one or two, as a reaction involving more substrates can be decomposed into a set of elementary reactions (Gillespie, 2007). Taking the example reactions above, the propensity function of the binding reaction of one TF molecule on the promoter will be: $c_2 \cdot x_{Promoter} x_{TF}$. It is therefore possible to link deterministic and stochastic rate constants, as shown in Gillespie (2007).

The system state change is then computed in terms of probability by the Chemical Master Equation (CME), which computes the evolution of the probability that the system is in state $\mathbf{X}$ through time. For details about its computation, we refer the reader to the review by El Samad et al. (2005). An analytical solution of the CME provides the probability density function of the system state $\mathbf{X}(t)$. However, as for deterministic models, the computation of an analytical solution is impossible except for quite simple systems. Therefore, one way to study the behaviour of the system is to construct numerical realizations of the CME. One of the most used method is Gillespie's Stochastic Simulation Algorithm (SSA) (Gillespie, 1977). SSA increments the system state at discrete time points, by randomly selecting the next reaction to fire, according to the propensity function of every possible reaction, as well as simulating the event (reaction occurring) time (El Samad et al., 2005; Gillespie, 1977). Several exact (i.e. simulating every single reaction) alternatives to the original algorithm (the so-called direct method) have been proposed, such as the next-reaction method (Gibson & Bruck, 2000), the sorting direct method (McCollum et al., 2006), and others (see Gillespie, 2007; Pahle, 2009 for a review). However exact algorithms are limited by their computational cost which renders the simulation of large systems intractable. Several approximation methods have been proposed, and have been thoroughly discussed (El Samad et al., 2005; Karlebach & Shamir, 2008; Turner et al., 2004; Wilkinson, 2009). Approximation simulations such as the very popular tau-leaping method considerably reduce the simulation time, but at the expense of a hardly estimable loss of accuracy (Wilkinson, 2009).

Stochastic discrete simulations offer a different perspective on the modelling of regulation compared to the deterministic continuous approach, as they explicitly model the binding of regulator molecules on the target promoter. Taking the example of TFs regulating the expression of a given gene, a stochastic model can represent the binding of regulatory molecules on the promoter of the target gene as follows:

$$\begin{aligned} \text{Promoter} + \text{TF} &\rightarrow \text{Active\_promoter} \\ \text{Active\_promoter} &\rightarrow \text{Promoter} + \text{TF} \\ \text{Active\_promoter} &\rightarrow \text{Active\_promoter} + \text{mRNA} \end{aligned}$$

In this model, a TF must be bound to the promoter for a transcription event to occur. Using this representation, it is easy to represent the different combinations of TFs bound to the promoter, and the transcription rate associated with each state. The possible combinatorial regulation effects can also be explicitly stated. For example, reactions can be added to encode the formation of a regulatory complex from the different TFs and to encode the binding of the complex on the target promoter.

The advantage of a stochastic model as opposed to its deterministic counterpart is its ability to fit more precisely to the natural variation inherent to biological systems. This biological fluctuation can be crucial for understanding certain systems, as illustrated by El Samad et al. (2005) that present several examples where a deterministic model fails to correctly predict the system behaviour. As for its deterministic counterpart, the stochastic framework implies a number of hypotheses on the represented system. In particular, it relies on the essential assumption that the system is well-stirred, that is, the molecules are homogeneously spread in the volume. Moreover the simulation of temporal trajectories is computationally heavier as each reaction (in the case of exact simulation algorithms) or group of reactions (for approximate methods) is simulated. This is especially true for a system with a high number of molecules or reactions with large value propensity functions, as both factors imply high firing rates.

#### Bridging the gap

Starting from a stochastic model and in particular the CME representation of a given system, it is possible by means of simplifying assumptions to obtain the corresponding continous deterministic model. As stated in the tau-leaping approximation of the SSA, under the assumption that the time step $\tau$ in the simulation is small enough so that the propensity functions of the different reactions stay approximately constant during the interval $[t, t+\tau]$, the number of reactions occurring during that time step can be modelled as a Poisson process (El Samad et al., 2005; Higham, 2008). Consequently, it is possible to sample the number of occurrences of each reaction with propensity function $a_j$ from a Poisson law with parameter $a_j \cdot \tau$. Moreover, if the time step $\tau$ is at the same time large enough so that each reaction fires more than once during the interval $[t, t+\tau]$ (usually feasible in systems where the concentration of species is large enough (El Samad et al., 2005)), the system can be represented by a set of stochastic differential equations, termed the Chemical Langevin Equation (CLE) (El Samad et al., 2005; Gillespie, 2000). In the CLE, the population of each species evolves in a continuous but stochastic manner, with the stochastic variation due to each reaction being proportional to the propensity of the reaction. As a consequence, the number of occurrences of a reaction with a high rate will have a higher variance than that of a reaction with a small reaction rate.

By further assuming that the abundance of each species is high enough, the stochasticity can be neglected, and the system is reduced to the Reaction Rate Equation (RRE) (El Samad et al., 2005; Higham, 2008), that is a set of differential equations:

$$\frac{d\mathbf{X}(t)}{dt} = \sum_{j=1}^{M} v_j \cdot a_j(\mathbf{X}(t)) \tag{A.8}$$

In this equation, the change per time unit in the concentration of a given species amounts to the sum over all reactions of the change in the species abundance triggered by one firing of the reaction (i.e. $v_j$), weighted by the rate of the reaction (i.e. the probability that the reaction will fire in one time unit, $a_j$). As highlighted by Higham (2008), it is important to keep in mind that the solution of a deterministic model or RRE obtained by simplifying a stochastic model is not equivalent to an average of many numerical realisations of the corresponding stochastic model. It is rather a limit towards which these realisations tend when the different simplifying assumptions are fulfilled.

This connection between the stochastic and deterministic frameworks has been leveraged in the case of multi-scale systems, which are systems in which both slow and fast reactions and/or both low- and high-abundance species are present (El Samad et al., 2005). This situation can be encountered for example when simulating gene expression and metabolic reactions in a single model. On one hand, gene expression is a slow process involving genes that are present in only one or two copies per cell, and TFs whose abundance can be as low as a dozen of molecules only. On the other hand, metabolic reactions are fast enzyme-catalysed processes and involve highly abundant metabolic species. The issue with such systems is that some but not all reactions could be represented by a deterministic process, while the rest requires a stochastic modelling. In such cases, the SSA performs poorly because of fast reactions and highly abundant species which monopolise most of the computational time. On the opposite, deterministic models provide a poor approximation of slow reactions and low abundant species. Several hybrid methods have hence been proposed. Their strategy is to split reactions and/or species in two sets of slow and fast reactions/species, and to use the most appropriate representation to model each set (see Pahle, 2009 for an overview of existing methods). This principle notably underlies the slow-scale SSA algorithm (Cao et al., 2005). The interested reader is referred to El Samad et al. (2005) and Pahle (2009) for more details.

### A.3.3 Model simplifications

In addition to the mathematical framework they employ, existing simulators of expression data differ by a number of assumptions they make. Indeed, while it is crucial to accurately represent biological processes, our incomplete knowledge about the detailed mechanisms dictates the use of assumptions and simplifications in the models. These simplifications also arise from the desired level of complexity and the need for computational efficiency. An important aspect to consider when designing a model is the type of molecules one wishes to represent. Early models were restricted to the simulation of the transcript levels only (see Table A.1), and used the concentration of mRNAs as a proxy for the activity of their protein product. Such an assumption was justified by the inability to experimentally measure protein concentration (Di Camillo et al., 2009). However, as shown earlier in this chapter, a number of regulations occur post-transcriptionally. This certainly impacts protein abundance and/or activity without being reflected at the level of corresponding transcripts, except when the expression of a coding gene is linked to the activity of its corresponding protein via a feedback circuit. In particular, many studies revealed a generally weak not to say poor correlation between transcript and protein profiles (Halbeisen et al., 2008; Vogel & Marcotte, 2012). Such results suggest the need for more realistic models in which proteins are also included as the direct actors of transcription regulation. Some models already include the protein level (see Table A.1). The inclusion of other regulatory

Table A.1: Overview of existing methods of expression data simulation and their characteristics

| Simulation method | Network topology | Mathematical formalism | Simulated molecules | Simulated reactions |
|---|---|---|---|---|
| Mendes *et al.*, 2003 | • Random<br>• Small-world<br>• Scale-free<br>• Regular grid | ODEs | mRNAs | • Transcription (regulated, Hill function)<br>• mRNA decay (1$^{st}$ order process) |
| Van den Bulcke *et al.*, 2006 - SynTReN | • Sampling from source network | ODEs (steady-state) | mRNAs | • Transcription (regulated, Hill and Michaelis-Menten functions)<br>• mRNA decay (1$^{st}$ order process) |
| Ribeiro *et al.*, 2007 - SGNSim | • User-defined | Time-delay stochastic model | Gene promoters, mRNAs, proteins, RNA polymerase, ribosomes | • Transcription (different transcription rate for each promoter state)<br>• Translation (1$^{st}$ order process)<br>• mRNA and protein decay (1$^{st}$ order processes) |
| Roy *et al.*, 2008 - RENCO | • Scale-free (protein-protein interaction)<br>• Exponential degree distribution (transcription network) | ODEs | mRNAs, proteins | • Transcription (regulated, Hill function)<br>• Translation (1$^{st}$ order process)<br>• mRNA and protein decay (1$^{st}$ order processes) |
| Di Camillo *et al.*, 2009 - NETSim | • Hierarchical modular topology model | ODEs | mRNAs | • Transcription (regulated, fuzzy logic)<br>• mRNA decay (1$^{st}$ order process) |
| Haynes *et al.*, 2009 - GRENDEL | • Distinct in- and out-degree distribution | ODEs | mRNAs, proteins, environment | • Transcription (regulated, Hill function)<br>• Translation (1$^{st}$ order process)<br>• mRNA and protein decay (1$^{st}$ order processes) |
| Hache *et al.*, 2009 - GeNGe | • Random<br>• Scale-free<br>• Regulatory motifs<br>• User-defined | ODEs | mRNAs, proteins, RNA polymerase, ribosomes | • Transcription (regulated, Hill function)<br>• Translation (1$^{st}$ order process)<br>• mRNA and protein decay (1$^{st}$ order processes or Michaelis-Menten decays) |
| Schaffter *et al.*, 2011 - GeneNetWeaver | • Module extraction from source network | Chemical Langevin Equation | mRNAs, proteins | • Transcription (regulated, Hill function)<br>• Translation (1$^{st}$ order process)<br>• mRNA and protein decay (1$^{st}$ order processes) |
| Pinna *et al.*, 2011 - SysGenSIM | • Random<br>• Scale-free<br>• Random modular<br>• Modular with exponential in-degree and power law out-degree | ODEs | mRNAs, cis- and trans-eQTLs | • Transcription (regulated, Hill function)<br>• mRNA decay (1$^{st}$ order process) |
| Tripathi *et al.*, 2017 - sgnesR | • User-defined | Time-delay stochastic model | Gene promoters, mRNAs, proteins | • Transcription (different transcription rate for each promoter state)<br>• Translation (1$^{st}$ order process)<br>• mRNA and protein decay (1$^{st}$ order processes) |

molecules, and in particular the noncoding yet very likely regulatory (Holoch & Moazed, 2015; Morris & Mattick, 2014; Wery et al., 2011) fraction of the transcriptome could also be an interesting development. In addition, post-transcriptional regulations are traditionally overseen in expression simulation methods. Accounting for them would result in an increased complexity of the underlying mathematical models, but would pave the way for enhanced realism of simulated data.

Biological processes are not instantaneous. Time delays exists between for example transcription initiation and the release of a fully functional mRNA ready to be translated. Such delays have been mostly ignored, to the exception of SGNSim (Ribeiro & Lloyd-Price, 2007) (implemented in the `R` package `sgnesR` (Tripathi et al., 2017)). These stochastic models use a version of the SSA that is suited for the occurrence of delay in biochemical reactions. This can account for the time required for the transcription of a gene as well as the diffusion of molecules across cellular compartments. Additionally, spatial inhomogeneities can be considered. For example, one might want to include in the model the different cellular compartments, to account for the fact that in eukaryotic cells the synthesis of mRNA occurs in the nucleus, while their translation happens in the cytoplasm. This can be done in a deterministic framework by using partial differential equations (de Jong, 2002).

### A.3.4 Assigning values for model parameters

When simulating *in silico* expression data, it is important to carefully choose the values of the different parameters in the model to obtain more plausible data. The initial abundance of each molecular species and the different reaction rates determine the resulting level of expression for each gene. It is crucial to use reasonable values in the range of those estimated from experimental datasets. The same attention must be paid to the kinetic parameters that define the strength and amplitude of regulation. This includes, for example, the Hill coefficients for a deterministic model, or the binding and unbinding rates of the different TFs on the promoter for a stochastic model. This choice is impeded by our limited knowledge about the precise kinetics of gene expression regulation (Bulcke et al., 2006). However a number of experimental results provide insights into the dynamics of the different molecular reactions, at least for some model organisms (Belle et al., 2006; Vogel & Marcotte, 2012). The global distribution of the lifetime of transcripts and proteins, for example, is starting to be well-characterized across the different domains of life. The order of magnitude of transcription and translation rates are also available. In Milo & Phillips (2016), Milo and Phillips gather relevant quantitative pieces of information about different biological processes, from cell component typical size estimates to the rates of transcription, translation or metabolic reactions. The associated database, BioNumbers (Milo et al., 2009), allows to search the literature for quantitative properties of biological systems. This is a valuable tool for modellers who seek realistic values for model parameters. Additionally, during the model construction it can be useful to conduct a sensitivity analysis to ensure that slight variations in parameter values do not produce completely different and/or surrealistic system behaviours.

Despite the increasing availability of quantitative knowledge regarding gene expression, to the best of our knowledge, none of the existing simulation tools presently offer a rigorous justification of the values used in their model. The parameters are usually sampled from large distributions to allow a wide variety of possible dynamical behaviour (e.g. from quasi-linear to step-like regulatory functions) (Bulcke et al., 2006; Di Camillo et al., 2009). Alternatively, parameter values can be required as input from the user, as it is the case in Ribeiro & Lloyd-Price (2007), Hache, Wierling, et al. (2009) or Tripathi et al. (2017). However the choice of values in the model often seem arbitrary (Mendes et al., 2003; Pinna et al., 2011; Roy et al., 2008; Schaffter et al., 2011). An interesting approach has been proposed by Haynes & Brent (2009). In their model, the transcription, translation, transcript and protein decay rates are sampled from values experimentally measured for real genes in *S. cerevisiae*. It should be noted however that this limits the validity of the simulations to this organism. Moreover, this approach is only possible for well-characterized model organisms for which abundant and reliable quantitative information is available.

In conclusion, it is important to anchor the mathematical RNA models in the biological reality not only via the represented molecules and interactions, but also through the quantitative information which is used to simulate the different reactions involved in gene expression.

### A.3.5 Experimental noise in *in silico* data

Even though the results of transcriptomics and other omics experiments provide an estimation of transcripts or other molecules level, they do not exactly reflect their precise *in vivo* abundance. Each step of the sample preparation process introduces to some extent bias in the quantitative estimation of the molecular concentrations. Such bias in turn impedes our ability to detect correlation between molecular profiles, or introduces spurious correlations, and needs to be accounted for when developing a reverse engineering approach. Consequently, when assessing the performance of such methods, it is important to test their robustness against increasing level of noise in the data.

Accordingly, several pipelines of data simulation include a step to add experimental noise in the resulting simulated expression profiles to mimic errors and bias introduced by the used measurement technology (Bulcke et al., 2006; Hache, Wierling, et al., 2009; Haynes & Brent, 2009; Mendes et al., 2003; Pinna et al., 2011; Schaffter et al., 2011). This step is particularly relevant for deterministic models, which produce data deprived of both biological and experimental noise. On the opposite, stochastic models already introduce some kind of variability in the dynamic profiles. The generation of *in silico* experimental noise is often based on models linking the measured intensity obtained with a particular technology (e.g. microarray or RNASeq; Lowe et al., 2017) to the true underlying concentration (Irizarry et al., 2005; Rocke & Durbin, 2001; Stolovitzky et al., 2007). Alternatively, a simple Gaussian noise can be added to the simulated data to introduce variation in order to blur existing correlations among molecular profiles (Hache, Wierling, et al., 2009; Mendes et al., 2003; Pinna et al., 2011; Schaffter et al., 2011). Mendes *et al.* proposed to use a Gamma distribution for experimental noise, as microarray data are often found to display a non-Gaussian noise and as the Gamma distribution is not centred around its mean.

## A.4 Concluding remarks

Biological systems are characterized by great complexity. From a systems perspective, regulatory networks are shaped according to specific properties that can be described mathematically. From a mechanistic perspective, gene expression is regulated at each step of the lifetime of the different gene products, that are transcripts and proteins. Their synthesis, activity and decay is tightly controlled by a vast array of factors ranging from proteins to noncoding transcripts and small molecules. Additionally, these processes are affected by an inherent stochasticity which induces variability in the molecular profiles. Statistical models provide a rich framework to represent this complexity, and it is now possible to generate *in silico* graphs resembling real networks, or to simulate biologically plausible noisy dynamic expression data. A statistical model must be carefully designed to accurately reflect the underlying processes, as for example determining the list of regulators of a molecule, quantifying the effect of regulatory factors, or assigning a value to the different reaction rates or other parameters.

In this chapter, we particularly focused on the use of GRN models for the simulation of expression data that can serve as benchmark for the testing of network inference algorithms. Indeed it is important that these simulations provide realistic data to allow researchers to draw meaningful conclusions about the performances of reverse engineering methods. However, beyond the problem of simulating expression data, all the identified regulatory relationships allow the modeller to account for a fine description of the underlying biological mechanisms, when such a level of detail is required.

We know that all models presume to a greater or lesser extent a simplification of the underlying biology. As we have shown in this chapter, most simulation models currently consider transcription regulation only, and exclude noncoding RNAs and possibly even proteins. While these simplifications can be justified by our insufficient knowledge about the processes at play or by computational limitations, it results in inadequate models as they overlook the complexity of gene regulation. We would however like to moderate this desire for increasingly detailed models that could be more indicative of the underlying biological and technical influences in the data. Indeed, they allow a great flexibility in the inferred interactions, but this can become problematic and result in overfitting and in the detection of spurious regulations. The need for a trade-off calls for the use of additional data for the reverse engineering problem as well as advanced statistical tools accounting for the missing information.

## Acknowledgements

We are very grateful for enriching discussions and suggestions on this manuscript made by Samantha Baldwin and Susan Thomson (Plant and Food Research, Lincoln, New Zealand). We would like to thanks reviewers of this chapter for their useful comments. MV was partly supported by a visiting professor scholarship from Aix-Marseille University.

## Appendix B

# Supplementary Material for Chapter 2

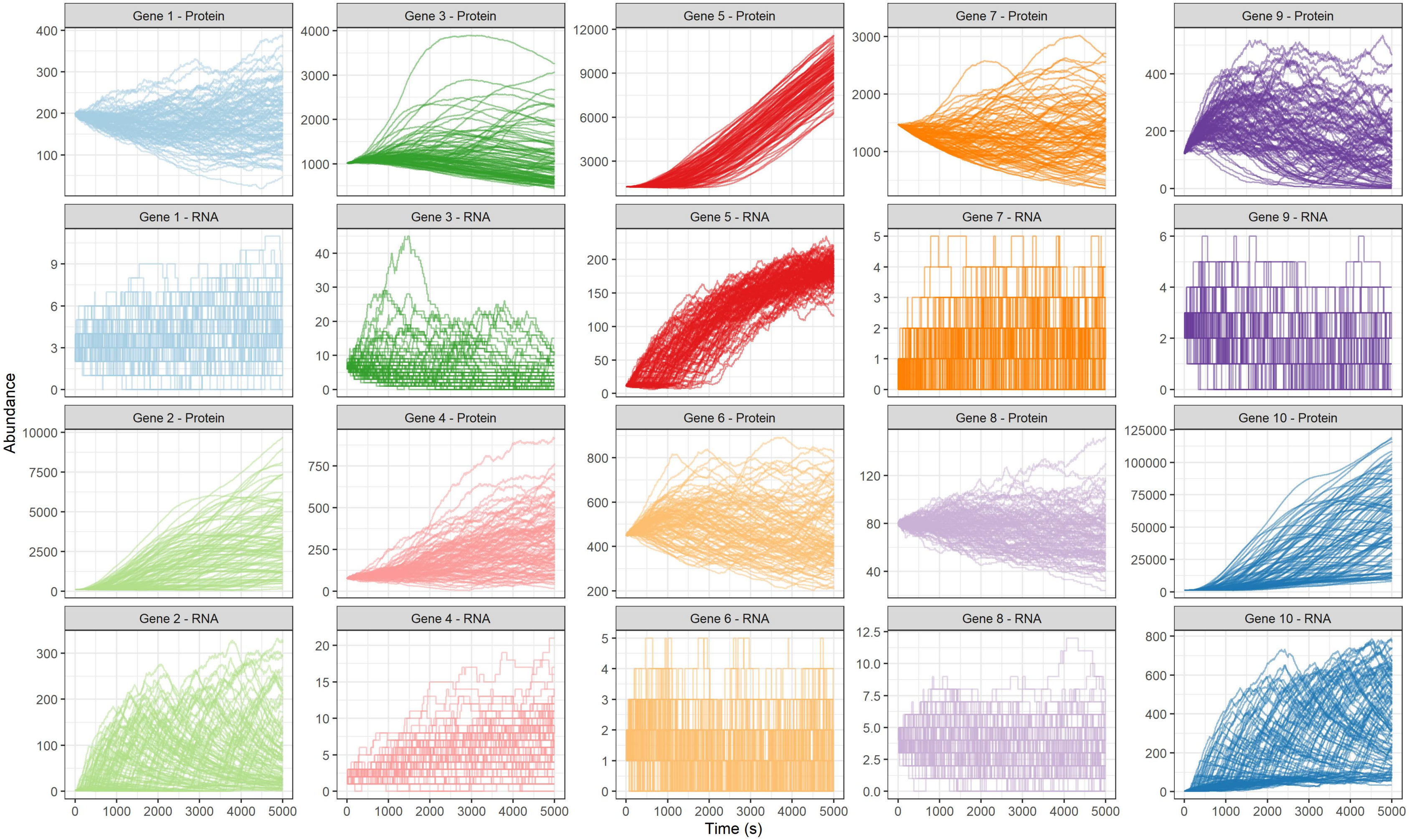


Figure B.1: RNA and protein abundance of the genes (one colour per gene) over time generated by `sismonr`, for 100 runs of the simulation. Each line corresponds to one simulation run.

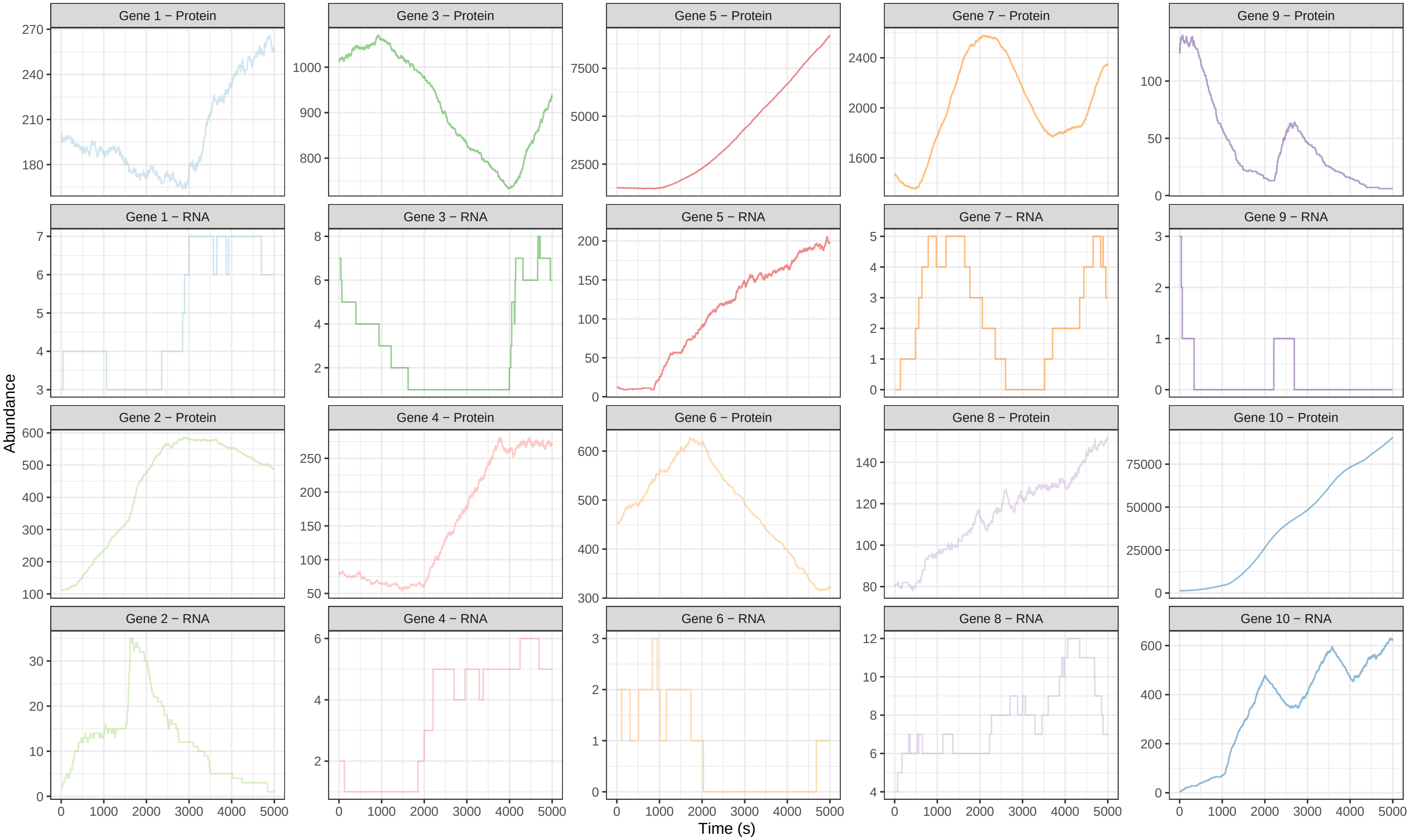


Figure B.2: RNA and protein abundance of the genes (one colour per gene) over time generated by `sismonr`, for one simulation run.

Table B.1: Values used for the transcription, translation, RNA decay and protein decay rates for the different genes in the simulation performed to compare `sismonr` and `sgnesR`.

| Gene | Transcription rate | Translation rate | RNA decay rate | Protein decay rate | RNA initial abundance | Protein initial abundance |
|---|---:|---:|---:|---:|---:|---:|
| Gene 1 | 0.0015407 | 0.0247583 | 0.0003701 | 0.0005110 | 3 | 199 |
| Gene 2 | 0.0014541 | 0.0152301 | 0.0008441 | 0.0002479 | 2 | 110 |
| Gene 3 | 0.0028237 | 0.0602271 | 0.0007811 | 0.0002235 | 7 | 1012 |
| Gene 4 | 0.0003567 | 0.0612851 | 0.0002420 | 0.0011264 | 2 | 79 |
| Gene 5 | 0.0068384 | 0.0218403 | 0.0005804 | 0.0002033 | 12 | 1265 |
| Gene 6 | 0.0014414 | 0.1178600 | 0.0014721 | 0.0002441 | 2 | 451 |
| Gene 7 | 0.0012738 | 0.3539754 | 0.0009573 | 0.0003174 | 0 | 1477 |
| Gene 8 | 0.0017283 | 0.0057417 | 0.0005600 | 0.0002848 | 4 | 80 |
| Gene 9 | 0.0007242 | 0.1481256 | 0.0006750 | 0.0012476 | 3 | 124 |
| Gene 10 | 0.0041457 | 0.1001083 | 0.0007304 | 0.0004520 | 4 | 1326 |

# Appendix C

# Supplementary File 1 for Chapter 2

## sismonr: Simulation of In Silico Multi-Omic Networks with adjustable ploidy and post-transcriptional regulation in R

---

Olivia Angelin-Bonnet[1], Patrick J. Biggs[1,2], Samantha Baldwin[3], Susan Thomson[3] and Matthieu Vignes[1]

[1]School of Fundamental Sciences, Massey University, Palmerston North, 4442, New Zealand,
[2]School of Veterinary Science, Massey University, Palmerston North, 4442, New Zealand, and
[3]The New Zealand Institute for Plant & Food Research Limited, Christchurch, 8140, New Zealand.

---

*Published as*:

> Olivia Angelin-Bonnet, Patrick J Biggs, Samantha Baldwin, Susan Thomson, and Matthieu Vignes. "sismonr: simulation of *in silico* multi-omic networks with adjustable ploidy and post-transcriptional regulation in R." *Bioinformatics* 36, no. 9 (2020): 2938-2940. https://doi.org/10.1093/bioinformatics/btaa002

## Abstract

**Summary**: We present `sismonr`, an R package for an integral generation and simulation of *in silico* biological systems. The package generates gene regulatory networks, which include protein-coding and non-coding genes along different transcriptional and post-transcriptional regulations. The effect of genetic mutations on the system behaviour is accounted for via the simulation of genetically different *in silico* individuals. The ploidy of the system is not restricted to the usual haploid or diploid situations, but can be defined by the user to higher ploidies. A choice of stochastic simulation algorithms allows us to simulate the expression profiles of the genes in the *in silico* system. We illustrate the use of `sismonr` by simulating the anthocyanin biosynthesis regulation pathway for three genetically distinct *in silico* plants.

**Availability**: The `sismonr` package is implemented in R and Julia, and is publicly available on the CRAN repository (`https://CRAN.R-project.org/package=sismonr`). A detailed tutorial is available from GitHub at `https://oliviaab.github.io/sismonr/`.

**Contact**: `m.vignes@massey.ac.nz`

## Introduction

In the past three decades, our approach to biological systems has shifted from a molecule-centric to a holistic point of view. Consequently, statistical and computational tools were developed to extract knowledge about biological networks and gene interactions from experimental gene expression data (Markowetz & Spang, 2007). Such inference tools require careful evaluation of their performance. This can be done by analysing reference experimental datasets and comparing the results to existing biological knowledge e.g. (e.g. Simoes et al., 2013), with the downside that such knowledge is often incomplete and/or biased. On the other hand, the use of simulated datasets as benchmarks ensure that the inferred networks can be compared to the indisputable network used to generate the data. Such approach has notably been used for the DREAM challenges (Marbach et al., 2010).

A number of simulation algorithms to generate *in silico* regulatory networks and simulate the expression profile of their genes have been proposed (e.g. Ribeiro & Lloyd-Price (2007), Pinna et al. (2011) - for a recent review see Angelin-Bonnet et al. (2019)). However, such simulators rely on simplified models of gene regulation that overlook most of the complexity inherent to biological systems. Here, we propose `sismonr`, an R package that generates and simulates holistic *in silico* systems. `sismonr` offers a significant improvement over existing simulators by including in one tool the ability to model non-coding genes (i.e. encoding regulatory RNAs), regulatory complexes, genetic mutations, unrestricted ploidy beyond the commonly considered haploid or diploid situation and different types of regulatory interactions ranging from transcriptional regulation to protein post-translational modification. We use a stochastic framework for the simulation of gene expression to account for intrinsic biological noise. In Supplementary Material 1, we provide an overview of existing methods for generating and simulating synthetic regulatory networks, and compare their performance to those of our package. With this more complete model of the gene expression regulation process, we come one step closer to simulating realistic *in silico* biological systems.

## C.1 Methods: biological system data simulation

The `sismonr` workflow consists of three main steps. First, the function *createInSilicoSystem()* generates an *in silico* system. A list of genes is constructed, and each of them is assigned a biological function, which dictates the type of regulation that they exert on their targets. Source-target regulatory relationships are defined via a gene regulatory network (GRN). This GRN comprises five different types of interactions or edges: regulation of transcription, translation and RNA decay, protein decay and protein post-translational modification. The out-degree of each regulator is sampled from either a power-law or an exponential distribution (see for example Guelzim et al., 2002) whose parameters can be controlled by the user. Edges are then added following a preferential attachment scheme to shape the in-degree distribution of the genes as a power-law. In addition, genes controlling a similar target can exert their regulatory role through the formation of a regulatory complex. The different kinetic parameters in the system are sampled from distributions set by the user or default distributions providing biologically realistic sets of values.

Second, to model the impact of genetic variation on gene expression profiles, the function *createInSilicoPopulation()* generates a population of genetically diverse *in silico* individuals. Rather than modelling directly the genome sequence of each individual, we represent the genetic mutations by their quantitative impact on the genes' properties. We consider *cis*-mutations, which directly affect the expression of a gene (e.g. its basal transcription or translation rate) as well as *trans*-mutations, which do not impact the expression of a gene but only the function or activity of its products (Pinna et al., 2011). To generate the *in silico* individuals, we first create a list of the variable gene alleles in the population. We then sample for each individual, according to the ploidy of the system, one or more homologs of each gene in this list of alleles.

Third, the user can choose between several versions of the Stochastic Simulation Algorithm (Gillespie, 2007; Wilkinson, 2012) to simulate the abundance of different molecules (RNAs, be they coding or non-coding, and proteins) over time for each individual, with the *simulateInSilicoSystem()* function. In order to simulate the expression of the genes in the system, we transform the GRN into a stochastic model, i.e. a list of species and a list of biochemical reactions with associated rates (see Figure C.1). We simulate transcription and translation regulation events by modelling the binding and unbinding of the regulators to and from their binding site on the target DNA or RNA, respectively.

Each regulator has the capacity to bind to a specific binding site on the target. The transcription or translation of the target then occurs with a rate depending on whether or not the binding sites are bound to their regulators. If bound, a regulator imposes a multiplicative fold change on the basal transcription or translation rate of the genes. For RNA and protein decay regulation, as well as for post-translational modification, the encounter of a regulator molecule and its target triggers the reaction at a certain rate. The natural and regulated decay of the different molecules occur even if they are bound by regulators, to targets or in complexes. As genes can be present in more than one copy in the system, the homolog of origin of the different molecules is tracked. This allows the user to compare the expression of the different homologs of a same gene. Note that these homologs can be identical, or can be different alleles of the genes, i.e. different versions of the gene with distinct properties. Details on the implementation of each step of the `sismonr` algorithm are presented in Supplementary Material 2.

## C.2 Implementation

The `sismonr` package is implemented in R and Julia. The main R functions call efficient Julia code via the R package `XRJulia` (Chambers, 2016). For stochastic simulations, `sismonr` uses the Julia module `BioSimulator` (Landeros et al., 2018), which implements different exact and approximate versions of the Stochastic Simulation Algorithm. A tutorial describing all functionalities of the package, including advanced visualisation tools, is available at `https://oliviaab.github.io/sismonr/%7D%7Bhttps://oliviaab.github.io/sismonr/`.

## C.3 Examples

In Supplementary Material 3, we detail several examples of the use of `sismonr`. This file also presents the code necessary to reproduce the examples. We start by showing how to create a simple regulatory network that exemplifies the different types of possible regulatory interactions. We then provide a small signal transduction cascade network to show the importance of modelling post-translational modifications. In order to demonstrate the ability of `sismonr` to model regulatory complexes and genetic mutations, we reproduce an *in silico* implementation of the experiments on the modelling the anthocyanin biosynthesis regulation (colour) pathway of different mutant plants (Albert et al., 2014). Lastly, we present the simulation of a system of 50 genes comprising protein-coding and non-coding genes, and transcription as well as post-transcriptional regulation. We also show that we can compare the expression of different homologs of a same gene.

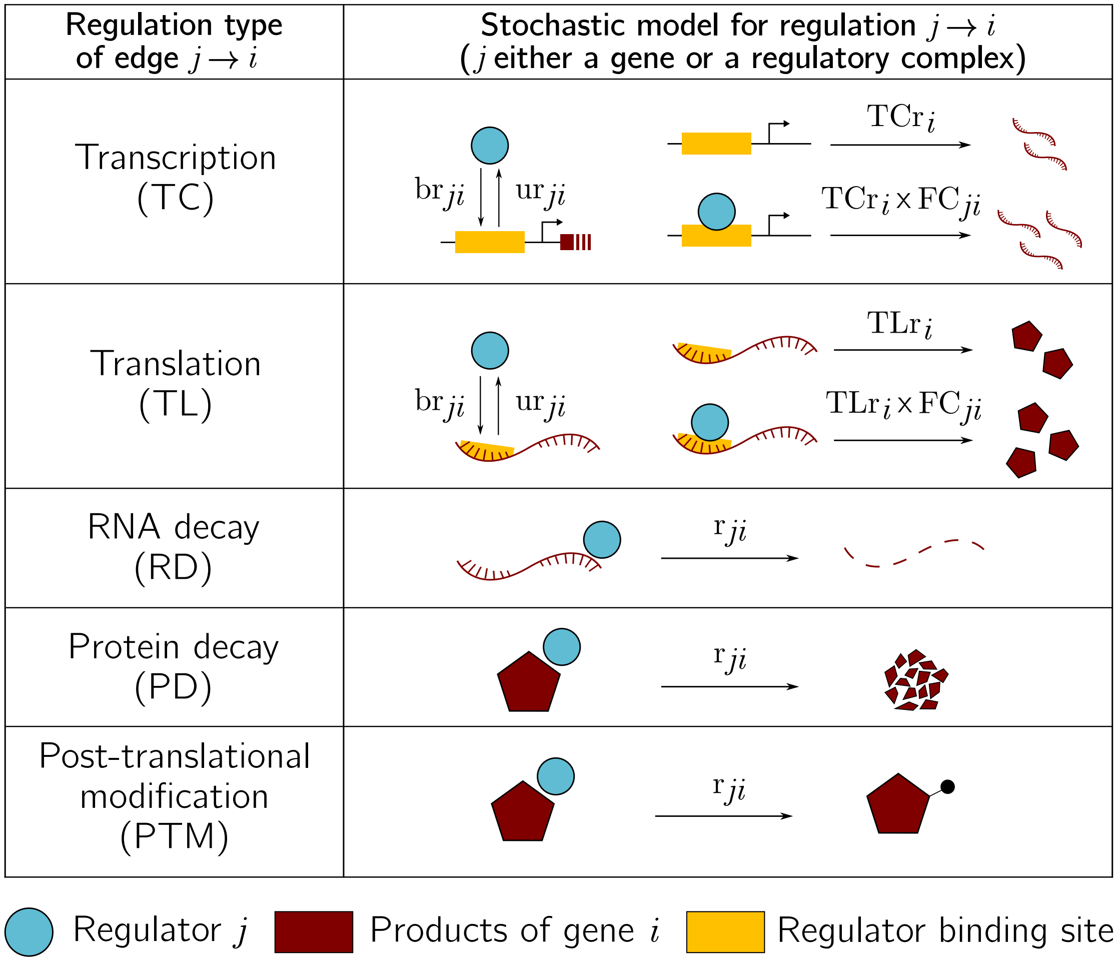


Figure C.1: Stochastic model of the different types of expression regulation. For transcription and translation regulation, the regulator $j$ binds and unbinds to and from its binding site on its target $i$ (DNA for transcription regulation and RNA for translation regulation) with rates $\mathrm{br}_{ji}$ and $\mathrm{ur}_{ji}$ respectively. Without a bound regulator, the transcription or translation of the gene $i$ occurs at its basal rate ($\mathrm{TCr}_i$ for transcription, $\mathrm{TLr}_i$ for translation). When the regulator is bound, the rate of transcription or translation of its target is multiplied by the induced expression fold-change $\mathrm{FC}_{ji}$. For RNA decay, protein decay and protein post-translational modification, the encounter of a regulator and its target triggers the decay or transformation of its target with rate $\mathrm{r}_{ji}$. Each rate is uniquely sampled for each regulation in the network.

## Conclusion

We present `sismonr`, an R package for the generation and the simulation of `in silico` biological systems. `sismonr` simulates the expression profiles of the genes linked via a regulatory network. Importantly, our model allows (i) to model five different types of expression regulation common in biological systems, expending the simulation of GRNs beyond the commonly considered transcription regulatory networks, (ii) to consider multiple ploidies, and (iii) to include regulation via non-coding RNAs. The package can be used to generate benchmark datasets for the evaluation of network inference methods. As the algorithm provides the abundance of both RNAs (coding and non-coding) and proteins, and models the impact of genetic variations, the benchmark datasets can also be used to validate multi-omic integration methods.

## Funding

This work was supported by a School of Fundamental Sciences (Massey University) Doctoral Scholarship to OA-B. SB and ST were supported by The New Zealand Institute for Plant and Food Research Limited Strategic Science Investment Fund.

## Acknowledgements

OA-B thanks John Chambers, author of the `XRJulia` package, and Alfonso Landeros, author of the `BioSimulator` module, for their help and responsiveness.

# Appendix D

# Supplementary File 2 for Chapter 2

```
library(sismonr)
library(tidyverse)
```

## Introduction

This document presents some examples of the use of `sismonr`. We start with simple examples illustrating important features of `sismonr`. We note that unless values are provided for the different kinetic parameters, these are sampled from distributions, and hence will differ when reproducing these examples. An extensive tutorial detailing the use and parameters of each `sismonr` function is available at `https://oliviaab.github.io/sismonr/`.

## D.1 Example 1: Post-transcriptional regulation in a 10-gene system

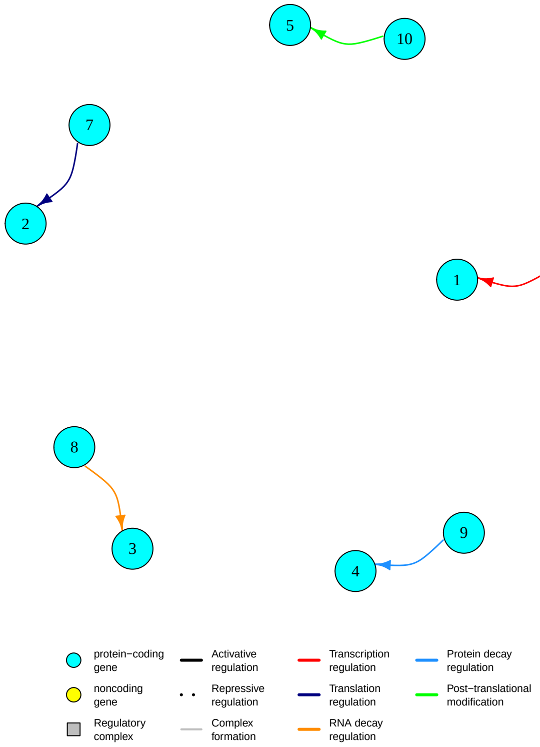
One of the major improvements of `sismonr` over existing simulators is the ability to generate and simulate regulatory networks including post-transcriptional regulation. This is important especially when using the generated datasets as benchmarks for network reconstruction methods, as post-transcriptional regulation alters the pattern of correlation between the RNA and protein profiles of the genes. Here we illustrate with a very simple example how the different types of regulation modelled by `sismonr` affect the expression profiles of target genes.

In this first example, we generate a system of 10 protein-coding genes. Genes 1 to 5 are each targeted by a different type of expression regulation. Genes 6 to 10 each regulate the expression of one of the target genes. This example illustrates how easily the user can create a personalised network, notably with the functions `addGene()` and `addEdge()` to add genes and regulatory interactions, respectively, to an existing network.

```r
## create a system of 5 protein-coding genes, no regulation
## PC.p is the probability of each gene to be protein-coding
system_regs = createInSilicoSystem(G = 5, PC.p = 1, empty = T)

## Each possible type of regulation:
## TC: transcription regulation
## TL: translation regulation
## RD: RNA decay regulation
## PD: Protein decay regulation
## PTM: Protein post-translational modification
regs = c("TC", "TL", "RD", "PD", "PTM")

for(i in 1:length(regs)){
  ## Add a new protein-coding gene with a specific biological
  ##function as defined in the vector regs
  system_regs = addGene(system_regs,
                        coding = "PC",
                        TargetReaction = regs[i])
  ## Add an activating edge from this new gene to a target gene
  system_regs = addEdge(system_regs,
                        5 + i,
                        i,
                        regsign = "1")
}
```

The generated network can easily be visualised with the `plotGRN()` function of `sismonr`:

```r
## Plot the regulatory network
plotGRN(system_regs, edge.arrow.size = 0.8)
```



We simulate the expression profiles of the genes for one *in silico* individual. The system is simulated for 1000s.

```r
## We'll run only one simulation of one individual
pop_regs = createInSilicoPopulation(1,
                                    system_regs,
                                    initialNoise = F)

## Simulate the system for this individual
sim_regs = simulateInSilicoSystem(system_regs,
                                  pop_regs,
                                  simtime = 1000)
```

By default, `sismonr` generates diploid systems, i.e. there are two copies or homologs of each gene in the system. Since the two homologs can be two different alleles (i.e. carrying different genetic mutations), `sismonr` tracks the homolog of origin of each molecule. This is represented in the output of the simulation by the `GCN` identifier:

```r
head(sim_regs$Simulation[, 1:10])
```

For example, the column `R5GCN2` corresponds to gene 5's RNAs, arising from the second homolog of the gene. However in this example we are not interested in the homolog of origin of the RNAs or

| time | trial | R5GCN2 | P5GCN2 | Pm5GCN2 | R7GCN2 | P7GCN2 | R3GCN1 | P3GCN1 | R1GCN2 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 3 | 620 | 0 | 5 | 256 | 2 | 88 | 3 |
| 1 | 1 | 3 | 620 | 0 | 5 | 256 | 2 | 88 | 3 |
| 2 | 1 | 3 | 619 | 0 | 5 | 256 | 2 | 88 | 3 |
| 3 | 1 | 3 | 618 | 1 | 5 | 256 | 2 | 88 | 3 |
| 4 | 1 | 3 | 617 | 3 | 5 | 256 | 2 | 88 | 3 |
| 5 | 1 | 3 | 617 | 3 | 5 | 256 | 2 | 88 | 3 |

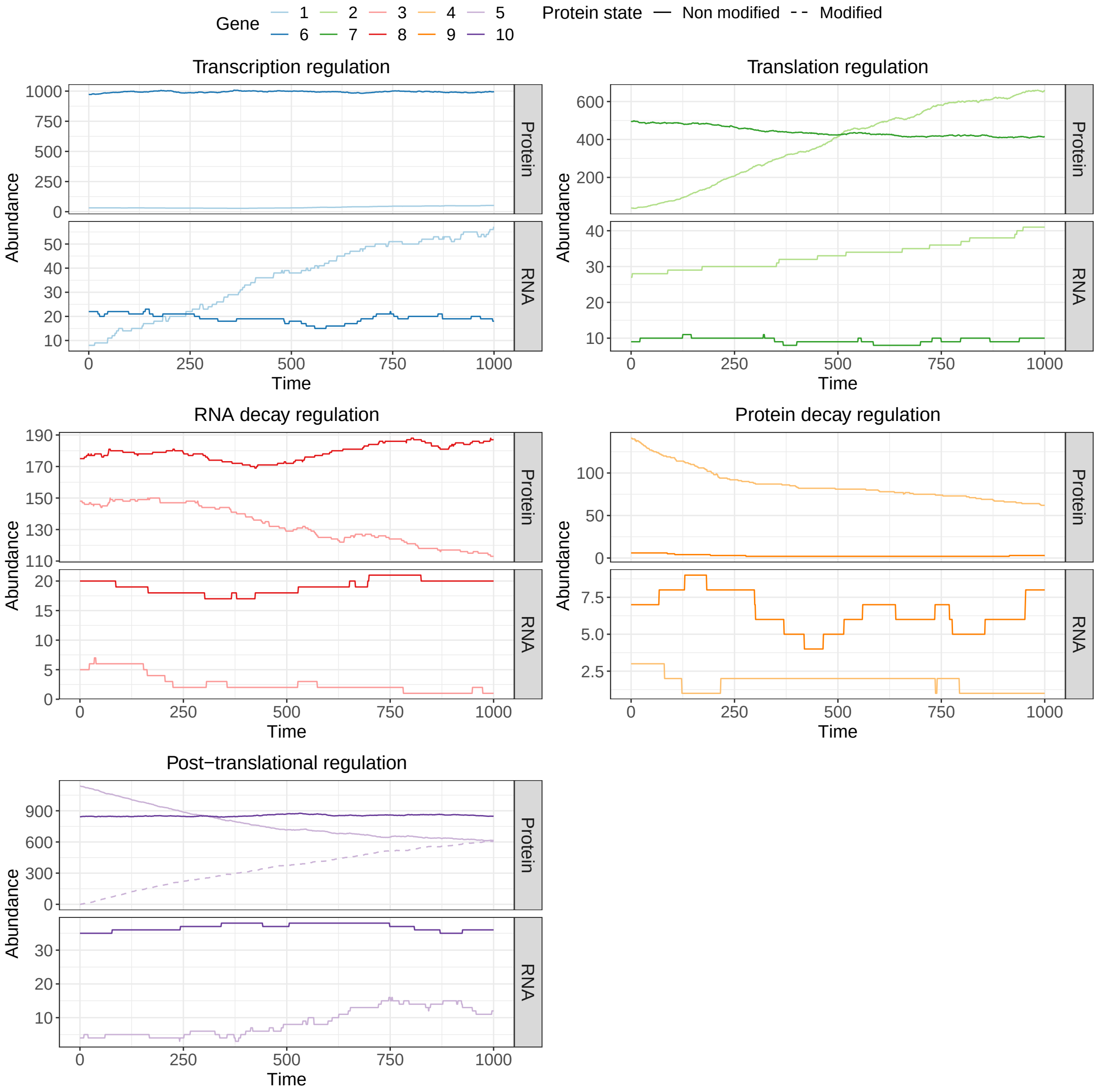
proteins, so we can merge the abundance of molecules coming from two homologs of a same gene:

```
## Merge the abundance of molecules coming from
## the two homologs of each gene
simres = mergeAlleleAbundance(sim_regs$Simulation)

head(simres[, 1:10])
```

| time | trial | Ind | R5 | P5 | Pm5 | R7 | P7 | R3 | P3 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Ind1 | 4 | 1137 | 0 | 9 | 493 | 5 | 148 |
| 1 | 1 | Ind1 | 4 | 1137 | 0 | 9 | 495 | 5 | 148 |
| 2 | 1 | Ind1 | 4 | 1136 | 0 | 9 | 495 | 5 | 148 |
| 3 | 1 | Ind1 | 4 | 1135 | 1 | 9 | 495 | 5 | 148 |
| 4 | 1 | Ind1 | 4 | 1133 | 4 | 9 | 495 | 5 | 148 |
| 5 | 1 | Ind1 | 4 | 1133 | 4 | 9 | 496 | 5 | 148 |

Now the column `R5` corresponds to gene 5's RNAs arising from either homolog of the genes.

We can see the impact of each type of regulation on the expression profile of the target genes. Note that `sismonr` provides a function for visualising the results of the simulation, but the following plot is used instead here for better emphasis of the difference between each type of regulation. The `sismonr` simulation visualisation feature will be demonstrated later.

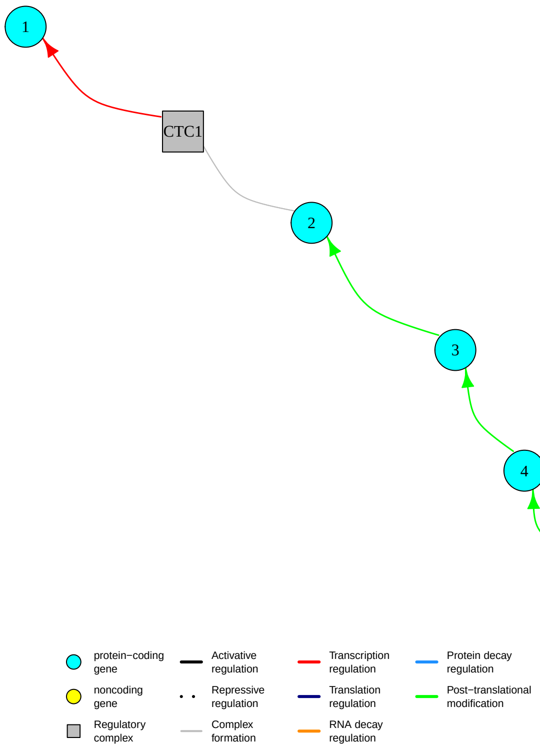


In the plot above, for each type of regulation, the RNA and protein abundance of the regulator gene are represented by the darker line, while those of the target are represented by a lighter line. As we can see clearly, each type of regulation affects a different step of the target expression. As gene 6 activates the transcription of gene 1, the RNA abundance of the latter increases over time. In the case of translation regulation, the RNA abundance of the target gene 2 remains constant while its protein abundance increases over time. Interestingly, the regulation of RNA decay of gene 3 also triggers the decrease of its protein levels, as there are no RNAs left to translate. On the contrary, regulation of protein decay does not impact the RNAs of the target gene 4. Lastly, gene 10 targets the proteins of gene 5 for post-translational modification. We can hence see the modified version of the protein (represented by a dotted line) appearing in the system. The ability of `sismonr` to model post-translational modification of proteins allows the user to simulate signal transduction within a regulatory network, as we illustrate below.

## D.2 Example 2: A small transduction network

In this example we model a small transduction network, similar to phosphorylation cascades found in eukaryotes. Upon activation (beginning of the simulation), gene 5 is expressed, and produce proteins that target gene 4's proteins for post-translational modification. Once modified, gene 4's proteins in turn modify the proteins produced by gene 3. Similarly, modified versions of gene 3's proteins target gene 2's proteins for post-translational modification. The latter can then form homodimers termed CTC1 that activate the expression of the target gene 1. Again, we use the functions `createInSilicoSystem()`, `addGene()` and `addEdge()` to create and modify an *in silico* system.

```
## Start with a system with only 1 gene (the target gene)
## empty = TRUE ensures that the returned system contains
## no regulatory interaction
system_cas = createInSilicoSystem(G = 1,
                                  PC.p = 1,
                                  empty = TRUE)

## Add a transcription factor activating the expression
## of the first gene as a homodimer:

## creating gene 2
system_cas = addGene(system_cas,
                     coding = "PC",
                     TargetReaction = "TC")
## creating homodimer of gene 2's proteins
system_cas = addComplex(system_cas, c(2, 2))
system_cas = addEdge(system_cas, "CTC1", 1,
                     regsign = "1",
                     kinetics = list("TCfoldchange" = 20))

## Add the post-translational modification cascade
for(i in 3:5){
  system_cas = addGene(system_cas,
                       coding = "PC",
                       TargetReaction = "PTM")
  system_cas = addEdge(system_cas,
                       i,
                       i-1,
                            regsign = "1")
}

plotGRN(system_cas)
```

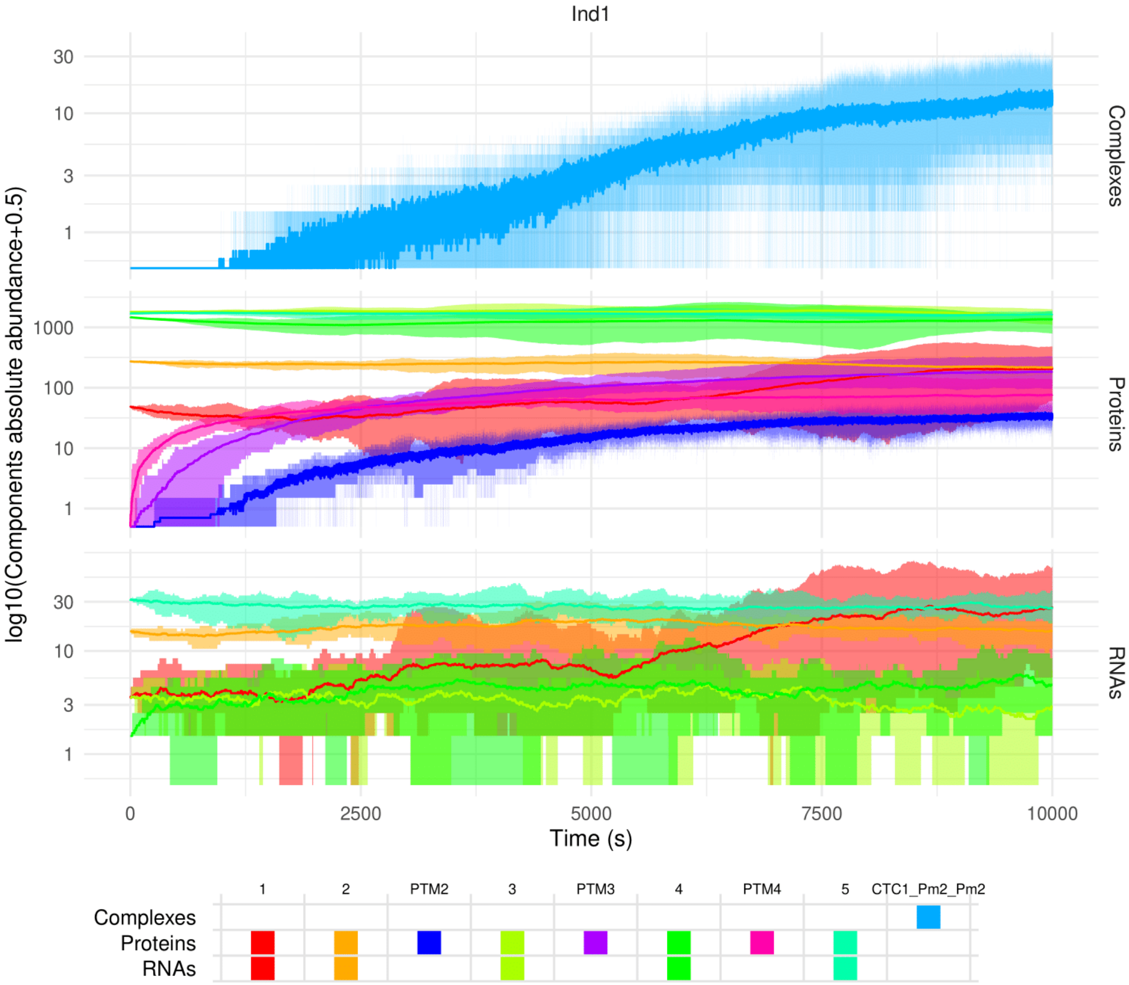


Again, we simulate the pathway for one *in silico* individual, over 10,000 seconds. We repeat the simulation 10 times (`ntrials = 10`).

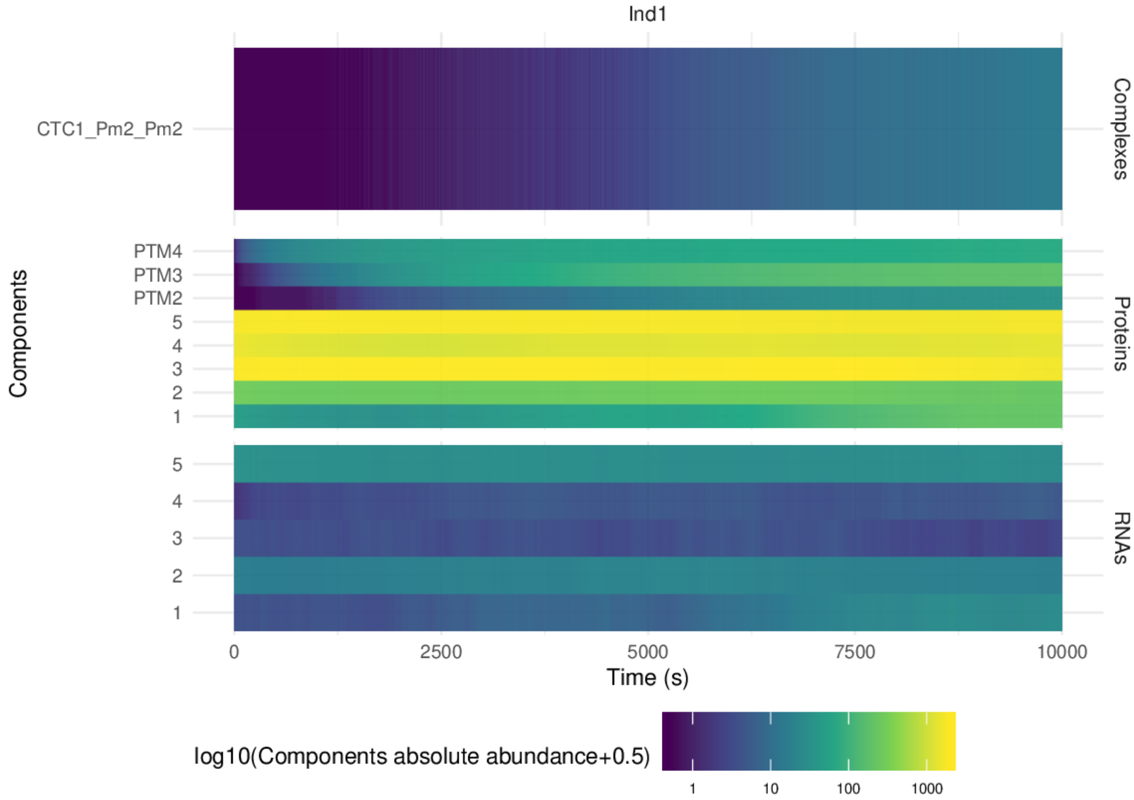
```
pop_cas = createInSilicoPopulation(1, system_cas, ngenevariants = 1)

## nepochs = number of measurements returned
sim_cas = simulateInSilicoSystem(system_cas, pop_cas,
                                    simtime = 10000,
                                    nepochs = 100000,
                                    ntrials = 10)
```

We use the simulation visualisation feature of `sismonr` to plot the expression profile of the genes over time, as curves with the function `plotSimulation()` and heatmaps with `plotHeatMap()`. In the fist case, the legend indicates which colour is associated to each gene and under which form the gene products can be found (here all genes are protein-coding so there exists a RNA and protein form for each of them). The modified versions of the different proteins are termed `PTM`. The solid lines represent the mean molecule abundance over the 10 simulations at each time-point, while the coloured areas denote the minimum and maximum values observed over the 10 simulations.

```
plotSimulation(sim_cas$Simulation, mergePTM = F)
```



```
plotHeatMap(sim_cas$Simulation, mergePTM = F, VirPalOption = "viridis")
```



From the graphs, we can see that as soon as the pathway is activated (begining of the simulation), modified proteins of gene 4 are created (PTM4 - pink line). They in turn trigger the modification of gene 3's proteins (PTM3 - purple line), that target gene 2's proteins for modification (PTM2 - dark blue line). The latter can then form homodimers CTC1 (CTC1_Pm2_Pm2 – cyan line). The complexes activate the transcription of target gene 1 (red line), whose RNA levels start increasing arount t = 5000s.

## D.3 Example 3: The anthocyanin biosynthesis regulation pathway

We next illustrate the modelling of genetic mutations and regulatory complexes by simulating the anthocyanin biosynthesis regulation pathway of Eudicots (Albert et al., 2014). Anthocyanin are pigments providing coloration to plants, flowers and fruits. (Albert et al., 2014) present the regulatory pathway controlling the expression of genes encoding enzymes responsible for the biosynthesis of anthocyanin in Eudicot plants (Figure D.1). Upon inductive conditions, the MYB gene is expressed, and the encoded proteins bind the proteins of the constitutively expressed genes WDR and bHLH1 to form a regulatory complex denoted MBW1. This complex activates the expression of the bHLH2 gene, whose protein also associates with the MYB and WDR proteins to form a second regulatory complex MBW2. This second regulatory complex reinforces the transcription of the bHLH2 gene, and activates the expression of the genes encoding enzymes involved in anthocyanin biosynthesis, here illustrated by the DFR gene. In addition, the MBW2 complex activates the expression of two repressor genes, R3-MYB and MYBrep. The former passively represses the production of MBW1 and MBW2 complexes by competitively binding the bHLH1 and bHLH2 proteins, while the later binds the MBW2 complex to form a repressive complex MBWr. This repressive complex silences the expression of the MBW2 targets, i.e. the bHLH2, DFR, MYBrep and R3-MYB genes. (Albert et al., 2014) compared the expression of the genes in the pathway for wild-type petunias, mutant petunias in which the MYBrep gene was overexpressed, and mutants in which the MYBrep gene was silenced. Here we reproduce their experiment *in silico* by simulating this regulation pathway with `sismonr`. We note that in absence of information about the values of the different kinetic parameters, we set the different parameters to reasonable values, that are justified in the section D.5.

```
## Gene ID - name correspondence
genes.name2id = data.frame("ID" = as.character(1:7),
                           "name" = c("MYB", ## 1
                                      "bHLH1", ## 2
                                      "WDR", ## 3
                                      "bHLH2", ## 4
                                      "MYBrep", ## 5
                                      "R3-MYB", ## 6
                                      "DFR"), ## 7
```

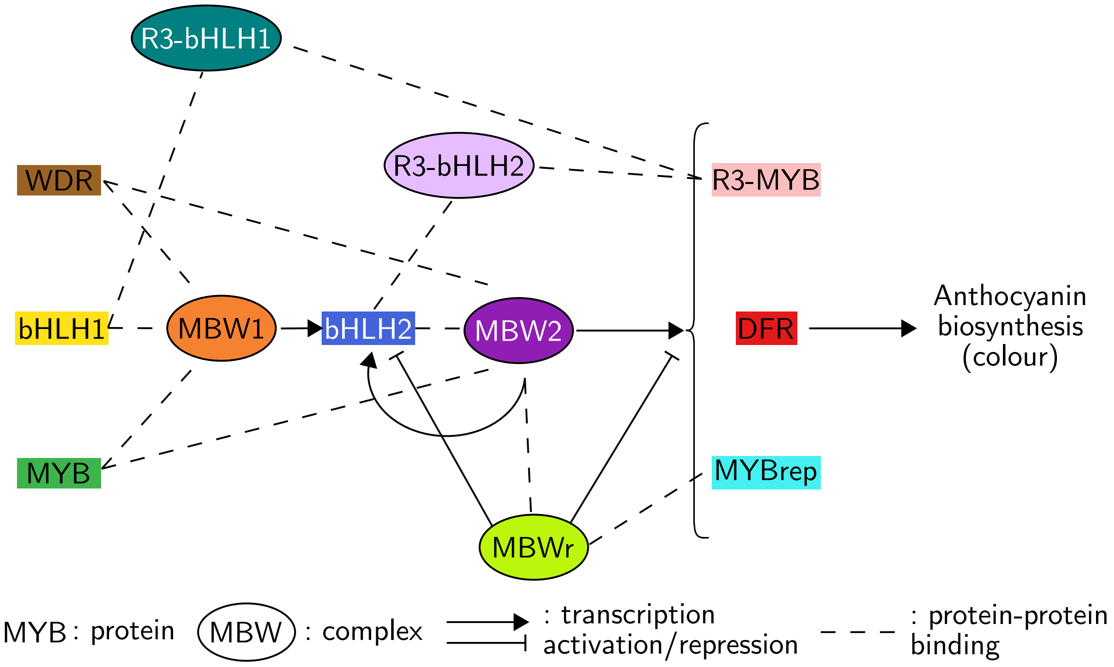

R3-bHLH1
R3-bHLH2
WDR
R3-MYB
bHLH1
MBW1
bHLH2
MBW2
DFR
Anthocyanin biosynthesis (colour)
MYB
MYBrep
MBWr
MYB : protein
MBW : complex
: transcription activation/repression
: protein-protein binding

Figure D.1: Schema of the anthocyanin biosynthesis regulation pathway as presented in Albert et al., 2014

```
                           stringsAsFactors = F)

## Complex ID - name correspondence
complexes.name2id = data.frame("ID" = paste0("CTC", 1:5),
                               "name" = c("MBW1", ## CTC1
                                          "MBW2", ## CTC2
                                          "MBWr", ## CTC3
                                          "R3-bHLH1", ## CTC4
                                          "R3-bHLH2"), ## CTC5
                               stringsAsFactors = F)

id2names = c(genes.name2id$name, complexes.name2id$name)
names(id2names) = c(genes.name2id$ID, complexes.name2id$ID)
```

```
## ---------------------------- ##
## Creating the in silico system ##
## ---------------------------- ##

## We create a system with 7 genes, and no regulatory
## interactions (they will be added manually)
colsystem = createInSilicoSystem(empty = T,
                                 G = 7,
                                 PC.p = 1,
## all genes are regulators of transcription:
                                 PC.TC.p = 1,
                                 PC.TL.p = 0,
                                 PC.RD.p = 0,
                                 PC.PD.p = 0,
                                 PC.PTM.p = 0,
                                 PC.MR.p = 0,
                                 ploidy = 2)

## Changing the kinetic parameters of the genes
colsystem$genes$TCrate = c(5, 0.1, 0.5, 0.01, 0.01, 0.1, 0.5)
colsystem$genes$TLrate = c(0.1, 0.01, 0.01, 0.01, 0.01, 0.01, 0.001),
colsystem$genes$RDrate = c(0.1, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01),
colsystem$genes$PDrate = c(0.01, 0.001, 0.001, 0.001,
                           0.001, 0.001, 0.001)
```

```
## Adding regulatory complexes in the system
compo = list(list("compo" = c(1, 2, 2, 3),
                  "formationrate" = 1, "dissociationrate" = 0.1),
             list("compo" = c(1, 3, 4,4),
                  "formationrate" = 2, "dissociationrate" = 0.1),
             list("compo" = c("CTC2", 5),
                  "formationrate" = 2.5, "dissociationrate" = 0.1),
             list("compo" = c(2, 6),
                  "formationrate" = 1.5, "dissociationrate" = 0.1),
             list("compo" = c(4, 6),
                  "formationrate" = 1.5, "dissociationrate" = 0.1))

for(comp in compo){
  colsystem = addComplex(colsystem,
                         comp$compo,
                         formationrate = comp$formationrate,
                         dissociationrate = comp$dissociationrate)
}

## Adding  regulatory interactions in the system
interactions = list(list("edge" = c("CTC1", 4),
                         "regsign" = "1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2,
                                           "TCfoldchange" = 25)),
                    list("edge" = c("CTC2", 4),
                         "regsign" = "1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2,
                                           "TCfoldchange" = 25)),
                    list("edge" = c("CTC2", 5),
                         "regsign" = "1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2,
                                           "TCfoldchange" = 50)),
                    list("edge" = c("CTC2", 6),
                         "regsign" = "1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2,
                                           "TCfoldchange" = 50)),
                    list("edge" = c("CTC2", 7),
                         "regsign" = "1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2,
                                           "TCfoldchange" = 15)),
                    list("edge" = c("CTC3", 4),
                         "regsign" = "-1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2)),
                    list("edge" = c("CTC3", 5),
                         "regsign" = "-1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2)),
                    list("edge" = c("CTC3", 6),
                         "regsign" = "-1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2)),
                    list("edge" = c("CTC3", 7),
                         "regsign" = "-1",
                         "kinetics" = list("TCbindingrate" = 0.1,
                                           "TCunbindingrate" = 2)))

for(inter in interactions){
  colsystem = addEdge(colsystem,
                      inter$edge[1],
                      inter$edge[2],
                      regsign = inter$regsign,
                      kinetics = inter$kinetics)
}


## ---------------------------------- ##
## Creating the in silico individuals ##
## ---------------------------------- ##
## We are going to simulate three different individuals/plants
## One is a wild-type plant (no mutation in any of its genes).
## The second is a mutant, in which gene 5 (the MYBrep gene)
## is overexpressed (here we increase its transcription rate by 50 +
## the gene becomes insensitive to transcription factors).
## The third is a mutant in which gene 5 (the MYBrep gene) is
## silenced (in the experiments the gene is silenced via RNA
## silencing, which increases the decay of the RNAs of the gene,
## which is what we are reproducing here).

plants = createInSilicoPopulation(3,
                                  colsystem,
                                  initialNoise = F,
                                  ngenevariants = 1)



## We add the QTL effect coefficients for the second individual
## such that the transcription rate of gene 5 is increased +
## the gene becomes insensitive to transcription factors
## (qtlTCregbind set to 0.)
## We have to change it for both homologs of the gene (GCN1 and GCN2)
## as the plants are diploid.
plants$individualsList$Ind2$QTLeffects$GCN1$qtlTCrate[5] = 50
plants$individualsList$Ind2$QTLeffects$GCN2$qtlTCrate[5] = 50
plants$individualsList$Ind2$QTLeffects$GCN1$qtlTCregbind[5] = 0
plants$individualsList$Ind2$QTLeffects$GCN2$qtlTCregbind[5] = 0

## We add the QTL effect coefficient for the second individual
## such that the RNA decay rate of gene 5 is increased
plants$individualsList$Ind3$QTLeffects$GCN1$qtlRDrate[5] = 12
plants$individualsList$Ind3$QTLeffects$GCN2$qtlRDrate[5] = 12

## Changing the initial conditions:
## As specified in Albert et al., 2014, only genes 2 and 3
## (bHLH1 and WDR) are constitutively expressed.
for(g in names(plants$individualsList$Ind1$InitAbundance)){
  for(i in names(plants$individualsList)){
    plants$individualsList[[i]]$InitAbundance[[g]]$R =
      plants$individualsList[[i]]$InitAbundance[[g]]$R *
      c(0, 1, 1, 0, 0, 0, 0)
    plants$individualsList[[i]]$InitAbundance[[g]]$P =
      plants$individualsList[[i]]$InitAbundance[[g]]$P *
      c(0, 1, 1, 0, 0, 0, 0)
  }
}

## --------------------------------------------------------------- ##
## Simulating the expression profiles of the genes for both plants ##
## --------------------------------------------------------------- ##

sim = simulateParallelInSilicoSystem(colsystem,
                                     plants, 2000,
                                     nepochs = 2000,
                                     ntrials = 100)
```

The running time for 100 repetitions of the simulation (stochastic system: 534 species, 3688 reactions) for the three individuals is around 90 minutes on a Virtual Machine Ubuntu 18.04 LTS 64-bit, 9.8 GiB RAM, 4 Intel Core i7-7700 CPU 3.60GHz cores (using parallelisation on three cores). The results of the simulation are shown in Figure D.2, and clearly illustrate the differences in expression profiles of the three types of *in silico* plants. In the wild-type plant, the rapid transcription and translation of the MYB gene (green) leads to the rapid formation of MBW1 complexes (orange), that in turn activate the transcription of the BHLH2 gene (dark blue). This allows the creation of MBW2 complexes (purple) that activate the transcription of the DFR gene (red). The MYBrep gene (in cyan) is also transcribed, but the synthesised proteins immediately bind the MBW2 complexes to form the MBWr complexes (lime).

The MYBrep-overexpressed mutant, as expected, shows increased expression of the MYBrep gene. In response, the expression of bHLH2 is more strongly repressed compared to the wild type plant. This lead to decreased levels of MBW2 complexes as all available complexes are immediatly bound by MYBrep proteins. The result is a lower expression of the DFR gene, leading to decreased levels of DFR enzymes. As these enzymes are responsible for the biosynthesis of anthocyanin, we can assume that a decrease in their abundance leads to a lower amount of anthocyanin and hence a reduced pigmentation, as observed in the experiment.

On the contrary in the MYBrep-silenced mutant, levels of MYBrep are lower than in the wild-type plant (as expected). Hence there is less feedback regulation for the expression of the bHLH2, DFR and

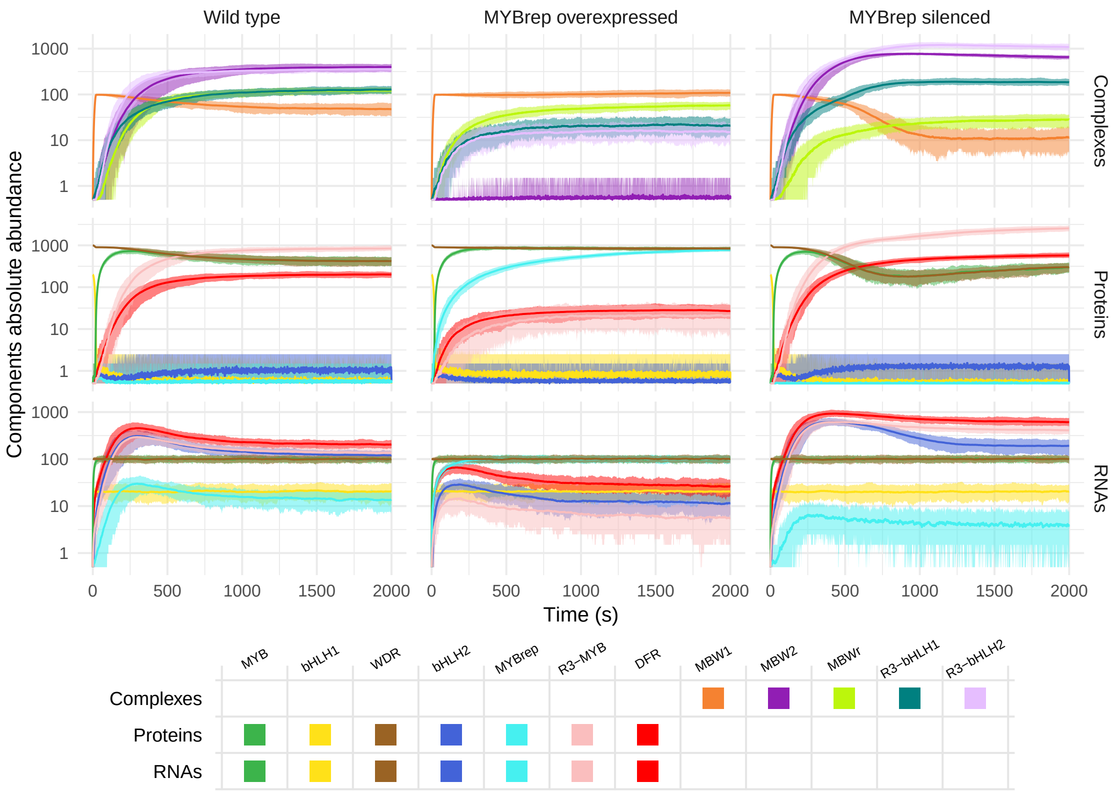


Figure D.2: Simulation of the anthocyanin biosynthesis regulation pathway. Note that this plot is similar to the plot automatically created with the `sismonr` function `plotSimulation()`. In this plot the name of each gene was used in the legend for better visualisation and the colour used for each gene and complex was matched to the pathway schema in Figure D.1.

R3-MYB genes, leading to an increased accumulation of DFR proteins and thus more pigmentation in the plants, as observed in the experiment.

In addition to observing colouration phenotypes, the authors also measured the RNA concentration of the different genes in the pathway for wild-type and mutant plants. In Table D.1 we present for each gene the experimental (i.e. as presented in (Albert et al., 2014)) and simulated mutant vs wild-type RNA concentration ratios for the different genes. Simulated ratios were computed at t = 2000s of the simulation to ensure that all RNA abundances reached a steady-state. It must be noted that the different kinetic parameters used in the simulation have not been optimised to reproduce the experimental results. Thus, we cannot expect to reproduce these ratios faithfully. However we can see that with reasonable values for the different parameters, we can obtain *in silico* values approaching the experimental observations. As expected, the RNA abundance of MYBrep gene is higher in the MYBrep-overexpressed mutant plant than in the wild-type plant (OE/WT ratio > 1), and lower in the MYBrepsilenced mutant plant (Si/WT ratio < 1). For the MYBrep-overexpressed mutant, the simulated ratio for MYBrep abundance is lower than what was measured experimentally. This arises

Table D.1: Experimental and simulated RNA ratios of the different genes in the anthocyanin biosynthesis pathway between each mutant plant and the wild-type plant.

| Gene | Gene name in petunia | Experimental OE/WT ratio | Simulated OE/WT ratio | Experimental Si/WT ratio | Simulated Si/WT ratio |
|---|---|---:|---:|---:|---:|
| WDR | AN11 | 1.56 | 1.01 | 3.12 | 1.00 |
| MYB | PHZ | 1.17 | 1.01 | 1.00 | 1.00 |
| bHLH1 | JAF13 | 0.90 | 1.01 | 0.70 | 0.99 |
| bHLH2 | AN1 | 0.11 | 0.10 | 2.17 | 1.59 |
| MYBrep | MYB27 | 40.00 | 7.23 | 0.33 | 0.28 |
| R3-MYB | MYBx | 0.05 | 0.04 | 33.33 | 3.00 |
| DFR | DFR | 0.10 | 0.13 | 3.30 | 2.98 |

from our choice to limit the increase in MYBrep transcription for the mutant plant, for visualisation purposes. However, any further increase in MYBrep transcription will lead to similar results for downstream genes as the repressor is already in excess in the system. For the MYBrep-silenced mutant individual, we chose values for the kinetic parameters related to the MYBrep mutation such that the simulated Si/WT ratio for MYBrep is close to the experimental ratio, in order to observe the effects of a similar reduction in MYBrep concentration.

In the simulation, we observe OE/WT and Si/WT ratios of one for the WDR, MYB and bHLH1 genes. This is because in the system the MYBrep repressor does not impact (directly or indirectly) the expression of these genes. These ratios are found experimentally to be different from one, suggesting that a change in MYBrep transcription does impact the expression of these genes. This finding hints at a possible direct or indirect regulation of the WDR, MYB and bHLH1 genes by MYBrep, as suggested in (Chen et al., 2019).

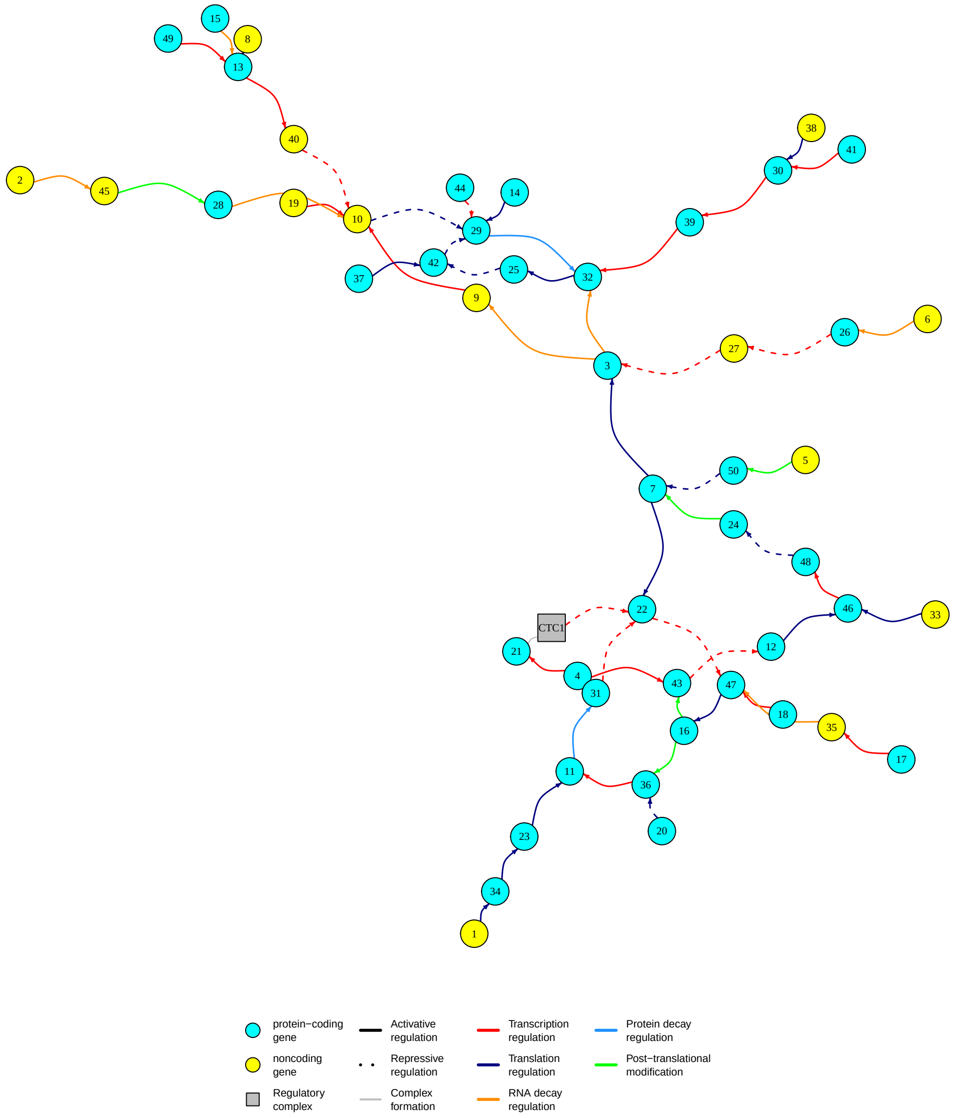
As was observed experimentally, the simulations demonstrate that an increase in MYBrep expression leads to a decrease in bHLH2 expression, and conversely a decrease in MYBrep expression results in an increase in bHLH2 expression. This is in accordance with the fact that MYBrep represses the expression of bHLH2 via MBWr complexes. An effect of similar magnitude is observed both experimentally and *in silico* for the expression of the DFR gene.

While the simulation is able to correctly reproduce a decrease in R3-MYB transcription when increasing the MYBrep expression with an amplitude similar to what is observed experimentally, it appears that a reduction in MYBrep levels lead to a drastic augmentation of R3-MYB RNA levels, beyond what the simulation was able to predict. This can be due to an incorrect parametrisation of the system or an additional regulation affecting R3-MYB that was not described in the pathway.

## D.4 Example 4: A system of 50 genes

Previous examples showed how `sismonr` can be used to model and simulate custom networks. However, `sismonr` can also generate random synthetic networks, whose properties can be controlled by the user. In particular, it is possible to control the probability of each gene to be protein-coding or non-coding, and their probability to control a specific step of the expression of their target. As an example, we generate here a network of 50 genes. We show here that it is possible to specify the probability of each gene to perform a specific regulatory function; moreover these probabilities can be defined separately for protein-coding and non-coding genes. If no value is provided by the user, default values are used, as it is the case here for the biological function probabilities for non-coding genes.

```r
system_g50 = createInSilicoSystem(G = 50, ## 50 genes
## When created, each gene as a probability of 0.7 of
##                              being protein-coding:
                                  PC.p = 0.7,
## probability of protein-coding genes to be regulators
##                                    of transcription:
                                  PC.TC.p = 0.4,
## probability of protein-coding genes to be regulators
##                                      of translation:
                                  PC.TL.p = 0.3,
## probability of protein-coding genes to be regulators
##                                        of RNA decay:
                                  PC.RD.p = 0.1,
## probability of protein-coding genes to be regulators
##          of protein post-translational modification:
                                  PC.PD.p = 0.1,
## probability of protein-coding genes to be regulators
##                                    of protein decay:
                                  PC.PTM.p = 0.05,
## probability of protein-coding genes to not be regulators:
                                  PC.MR.p = 0.05,
 ## number of copies of each gene present in the system:
                                  ploidy = 2)

plotGRN(system_g50,
        vertex.size = 6,
        vertex.label.cex = 0.7,
        edge.arrow.size = 0.3)
```



Also, when generating an *in silico* population, it is possible to define the number of alleles existing for each gene. We set this value here to five.

```
pop_g50 = createInSilicoPopulation(500,
                                   system_g50,
                                   ngenevariants = 5)
sim_g50 = simulateParallelInSilicoSystem(system_g50,
                                         pop_g50,
                                         simtime = 1000)
```

The quantitative impact of the genetic mutations carried by the different alleles of the genes on the genes' properties are stored in the `GeneVariants` element of the *in silico* population list. We show here the different alleles of genes 2 and 3.

```
pop_g50$GenesVariants[2:3]
```

```
$`2`
             1         2         3         4         5
qtlTCrate      1 0.8740067 1.1029492 1.0458018 1.0246933
qtlRDrate      1 1.0528457 0.9682368 1.0000000 0.9966078
qtlTCregbind   1 1.0000000 1.0000000 0.9139486 1.0000000
qtlRDregrate   1 0.9674559 0.9672182 1.0000000 1.0000000
qtlactivity    1 0.8787309 1.0587700 1.0000000 1.0000000
qtlTLrate      0 0.0000000 0.0000000 0.0000000 0.0000000
qtlPDrate      0 0.0000000 0.0000000 0.0000000 0.0000000
qtlTLregbind   0 0.0000000 0.0000000 0.0000000 0.0000000
qtlPDregrate   0 0.0000000 0.0000000 0.0000000 0.0000000
qtlPTMregrate  0 0.0000000 0.0000000 0.0000000 0.0000000

$`3`
             1         2         3         4        5
qtlTCrate      1 0.9945262 1.0971478 1.0055041 1.000000
qtlRDrate      1 1.0000000 1.1031950 1.0011387 1.063594
qtlTCregbind   1 1.0000000 1.0566075 1.0659187 1.000000
qtlRDregrate   1 1.0066827 0.9685712 1.0370861 1.000000
qtlactivity    1 1.0000000 0.9347300 1.2316077 1.000000
qtlTLrate      1 1.0000000 1.0037716 1.0000000 1.000000
qtlPDrate      1 1.0347019 0.8256616 1.2414675 1.000000
qtlTLregbind   1 1.0045350 0.8630083 1.0324867 1.000000
qtlPDregrate   1 1.1545935 0.9376130 0.8610631 1.159777
qtlPTMregrate  1 1.0138285 0.9195750 0.9868096 1.000000
```

The different alleles of each gene carried by an *in silico* individual can also be retrieved. Here we show which alleles of genes 1 to 5 the individual `Ind1` carries.

```
pop_g50$individualsList$Ind1$haplotype[1:5, ]
```

| GCN1 | GCN2 |
|---|---|
| 3 | 3 |
| 4 | 2 |
| 1 | 1 |
| 4 | 4 |
| 4 | 3 |

As the individuals are diploid, they carry two homologs of each gene (`GCN1` and `GCN2`). This table shows for each gene (row) which allele individual `Ind1` carries as its first (`GCN1`) and second homolog (`GCN2`) of the gene. This allows us to compare the expression of the different homologs of a gene for a given individual.

Because `sismonr` tracks the homolog of origin of each molecule, we can compare the expression of two homologs of a same gene for different individuals. For example here, we compare the expression of gene 25 for individuals `Ind1` and `Ind2`. `Ind1` carries two copies of allele 2 of the gene, while `Ind2` carries the alleles 3 and 4 of the gene:

```
pop_g50$individualsList$Ind1$haplotype[25,]
```

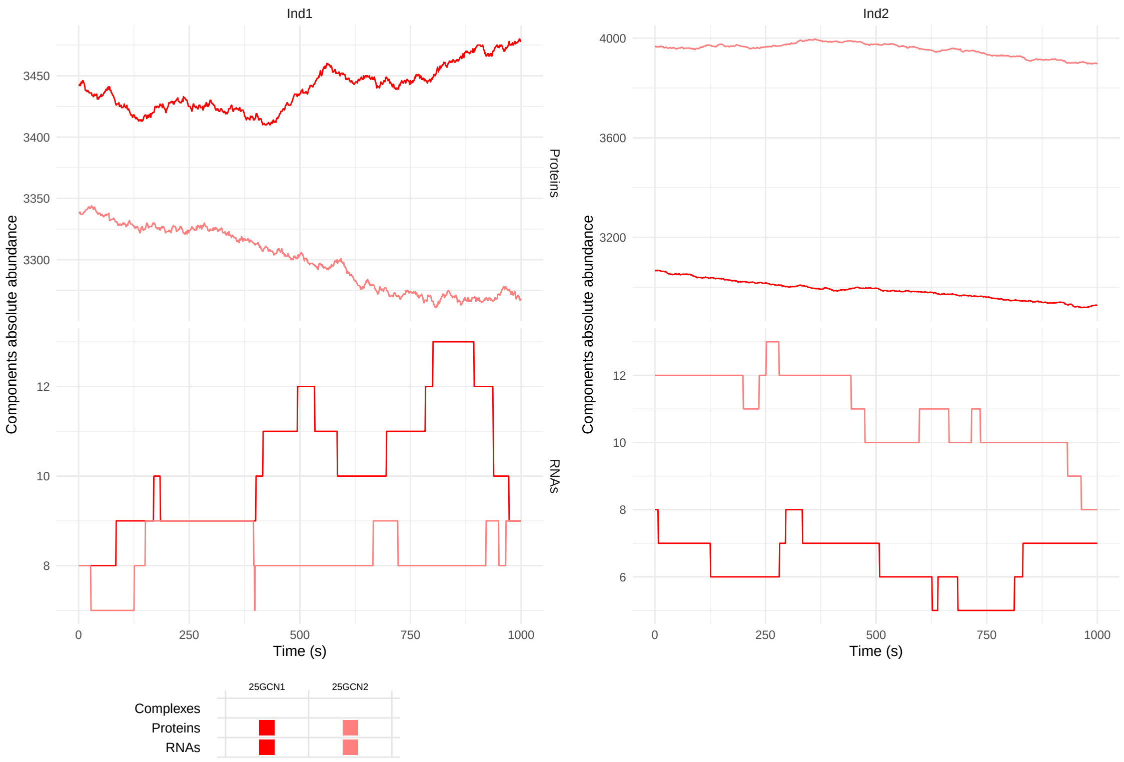
| | GCN1 | GCN2 |
|---|---|---|
| 25 | 2 | 2 |

```
pop_g50$individualsList$Ind2$haplotype[25,]
```

| | GCN1 | GCN2 |
|---|---|---|
| 25 | 4 | 3 |

```
plotSimulation(sim_g50$Simulation,
               molecules = c(25),
               inds = c("Ind1", "Ind2"),
               mergeAllele = F, ## plot separately the molecules
               ## coming from each homolog of
               ## the gene
               yLogScale = F)
```

```
Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
use `guide = "none"` instead.
```



We can see that for `Ind1` both homologs of the gene (termed `GCN1` and `GCN2`) produce a similar amount of RNAs and proteins. On the contrary for `Ind2` the two homologs produce different amounts of RNA and especially proteins. This is to be expected as the two homologs are identical for `Ind1` but for `Ind2` are two different alleles or versions of the gene, with different kinetic properties.

## D.5 Anthocyanin biosynthesis regulation pathway simulation settings

This section details the choice of parameters for the anthocyanin biosynthesis pathway simulation. Please note that these parameters have been chosen to obtain reasonable expression profiles, but have not been optimised to reproduce the RNA ratios of the genes between the different mutants as is presented in (Albert et al., 2014).

### D.5.1 Basal expression rates

The different kinetic rates associated with the expression of the genes in the pathway have been set as described in Table D.2:

Table D.2: Genes kinetic parameters for the anthocyanin biosynthesis regulation pathway

| Gene | Transcription rate (RNA/s) | Translation rate (protein/RNA/s) | RNA decay rate (/s) | Protein decay rate (/s) |
|---|---|---|---|---|
| MYB | 5 | 0.1 | 0.1 | 0.01 |
| bHLH1 | 0.1 | 0.01 | 0.01 | 0.001 |
| WDR | 0.5 | 0.01 | 0.01 | 0.001 |
| bHLH2 | 0.01 | 0.01 | 0.01 | 0.001 |
| MYBrep | 0.01 | 0.01 | 0.01 | 0.001 |
| R3-MYB | 0.1 | 0.01 | 0.01 | 0.001 |
| DFR | 0.5 | 0.001 | 0.01 | 0.001 |

In absence of information regarding the individual transcription, translation and decay rates of the genes, we assumed in the simulation that the genes in the pathway had similar kinetic properties. We initially reduced by a factor 10 the transcription rate of the bHLH2, MYBrep, R3-MYBrep and DFR genes as they are only expressed upon activation by their transcription factors. The fold-changes associated to the different activating transcription regulations were chosen to be in the order of magnitude of 10, such that once activated the transcription of the abovementioned genes occurs at a rate similar to these of the non-activated genes (WDR, bHLH1 and MYB). However the transcription rate of DFR and R3-MYB were increased by a factor 10 to be able to observe a non-null abundance of their respective RNAs in the MYBrep-overexpressed mutant individual. The MYB and WDR genes was then further increased by five as their proteins are the components of both MBW1 and MBW2 complexes. The translation rate of the DFR gene was reduced by a factor 10 compared to the other genes simply for visualisation purpose, which did not have any impact on the system behaviour as the DFR proteins are the end-product of the pathway and do not play any regulatory role. For all genes but WDR and bHLH1, we set their initial abundance to zero, as WDR and bHLH1 are the only constitutively expressed genes.

### D.5.2 Regulation rates

The parameters associated with the different transcription regulations are presented in Table D.3. Again, in absence of information about the binding and unbinding rates of the different transcription factors, we assigned the same values for all regulations.

Table D.3: Regulation kinetic parameters for the anthocyanin biosynthesis regulation pathway

| Regulatory edge | Binding rate (/molecule/s) | Unbinding rate (/s) | Transcription fold change (1) |
|---|---|---|---|
| MBW1 → bHLH2 | 0.1 | 2 | 25 |
| MBW2 → bHLH2 | 0.1 | 2 | 25 |
| MBW2 → MYBrep | 0.1 | 2 | 50 |
| MBW2 → R3-MYB | 0.1 | 2 | 50 |
| MBW2 → DFR | 0.1 | 2 | 15 |
| MBWr ⊢—⊣ bHLH2 | 0.1 | 2 | 0 |
| MBWr ⊢—⊣ MYBrep | 0.1 | 2 | 0 |
| MBWr ⊢—⊣ R3-MYB | 0.1 | 2 | 0 |
| MBWr ⊢—⊣ DFR | 0.1 | 2 | 0 |

### D.5.3 Complexes rates

The association rates of the different regulatory complexes are presented in Table D.4. were chosen according to the following rationale analysis in that we assumed:

- the formation of MBW2 complexes is slightly more efficient than the formation of the MBW1 complexes, as the latter is only starting the regulation pathway but not regulating the expression of the genes at the end of the pathway;
- the formation of MBWr complexes is slightly more efficient than the formation of the MBW2 complexes, to enforce the negative feedback mechanism;
- the formation of R3-bHLH1 and R3-bHLH2 complexes is slightly more efficient than the formation of the MBW1 complexes, to enforce the negative feedback mechanism.

We noted however that repeating the simulation with all complex association rates set to the same value lead to similar results.

Table D.4: Regulatory complexes kinetic parameters for the anthocyanin biosynthesis regulation pathway

| Regulatory complex | Association rate (/molecule/s) | Dissociation rate (/s) |
|---|---|---|
| MBW1 | 1 | 0.1 |
| MBW2 | 2 | 0.1 |
| MBWr | 2.5 | 0.1 |
| R3-bHLH1 | 1.5 | 0.1 |
| R3-bHLH2 | 1.5 | 0.1 |

### D.5.4 Genetic mutations

The value of the QTL effect coefficients used in the simulation are presented in Table D.5. The wild-type individual does not possess any mutation (all QTL effect coefficients set to one). In their experiment, (Albert et al., 2014) overexpressed the MYBrep gene in the first mutant plants by using a virus promoter for the gene. We reproduce in silico this mutation by increasing the transcription rate of the gene in the mutant individual. This is is done by setting the QTL effect coefficient `qtlTCrate` of the MYBrep gene to 50. We use this value to allow a clear visualisation. Note that this coefficient can be further increased to obtain a mRNA ratio of 40 as experimentally measured, but as the MYBrep protein is already in excess in the system any further increase in MYBrep transcription does not affect the overall results of the simulation. For this mutant individual, we also set the `qtlTCregbind` QTL effect coefficient of the MYBrep gene to zero, so that the transcription regulators (activators as well as repressors) of the gene do not show any affinity for the gene promoter and hence the transcription of the gene is independent of the regulators. This is a reasonable assumption since the MYBrep gene is expressed with a different promoter that probably does not contain binding sites for the original regulators of MYBrep. The MYBrep knock-down plants were experimentally obtained by RNA interference. We reproduce this RNA interference by increasing the decay rate of the MYBrep gene in the mutant by 12, i.e. setting the QTL effect coefficient `qtlRDrate` of the gene to 12. This value gives an mRNA abundance ratio of 0.4 when comparing wild-type plant to the MYBrep-silenced mutant plants, similar to the experimental value.

Table D.5: QTL effect coefficients for the anthocyanin biosynthesis regulation pathway

| *In silico* plant | Affected QTL effect coefficient | QTL effect coefficient value |
|---|---|---|
| Wild-type | – | – |
| MYBrep overexpressed | `qtlTCrate` | 50 |
| MYBrep overexpressed | `qtlTCregbind` | 0 |
| MYBrep silenced | `qtlRDrate` | 12 |

## Appendix E

# Supplementary Material for Chapter 3

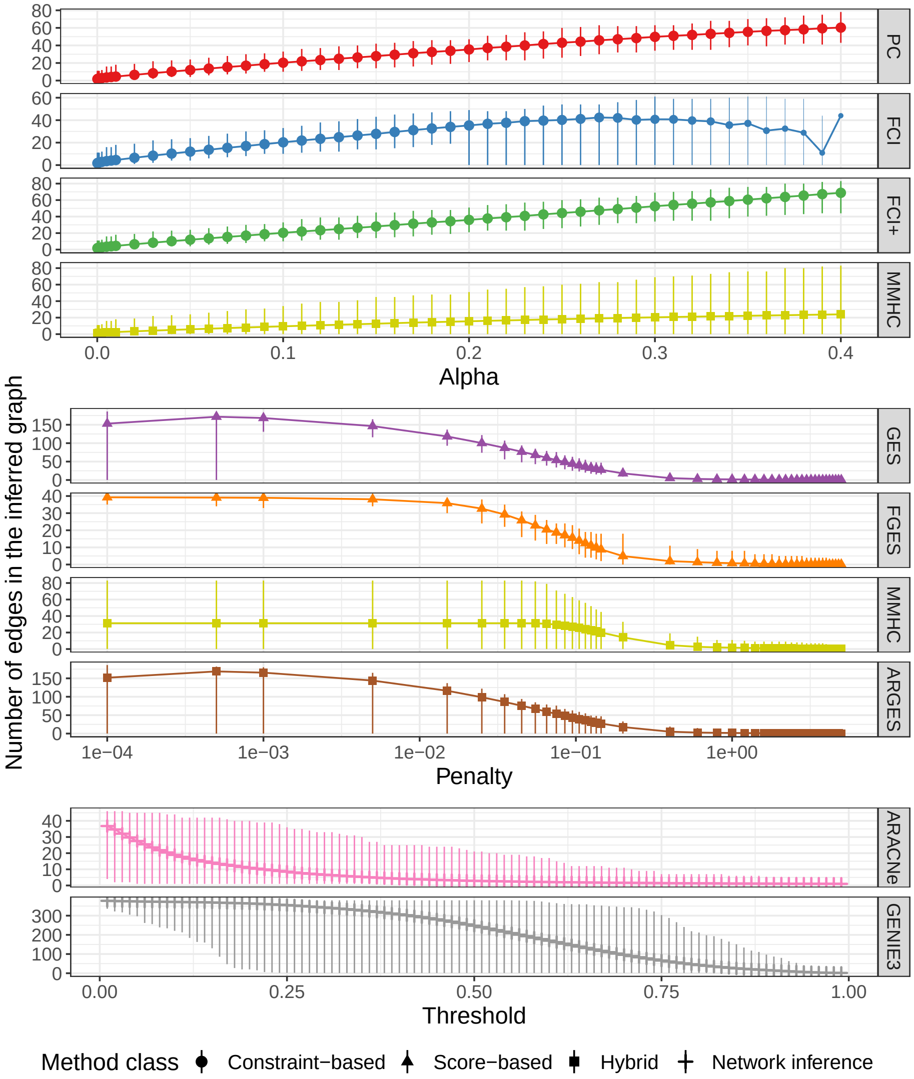
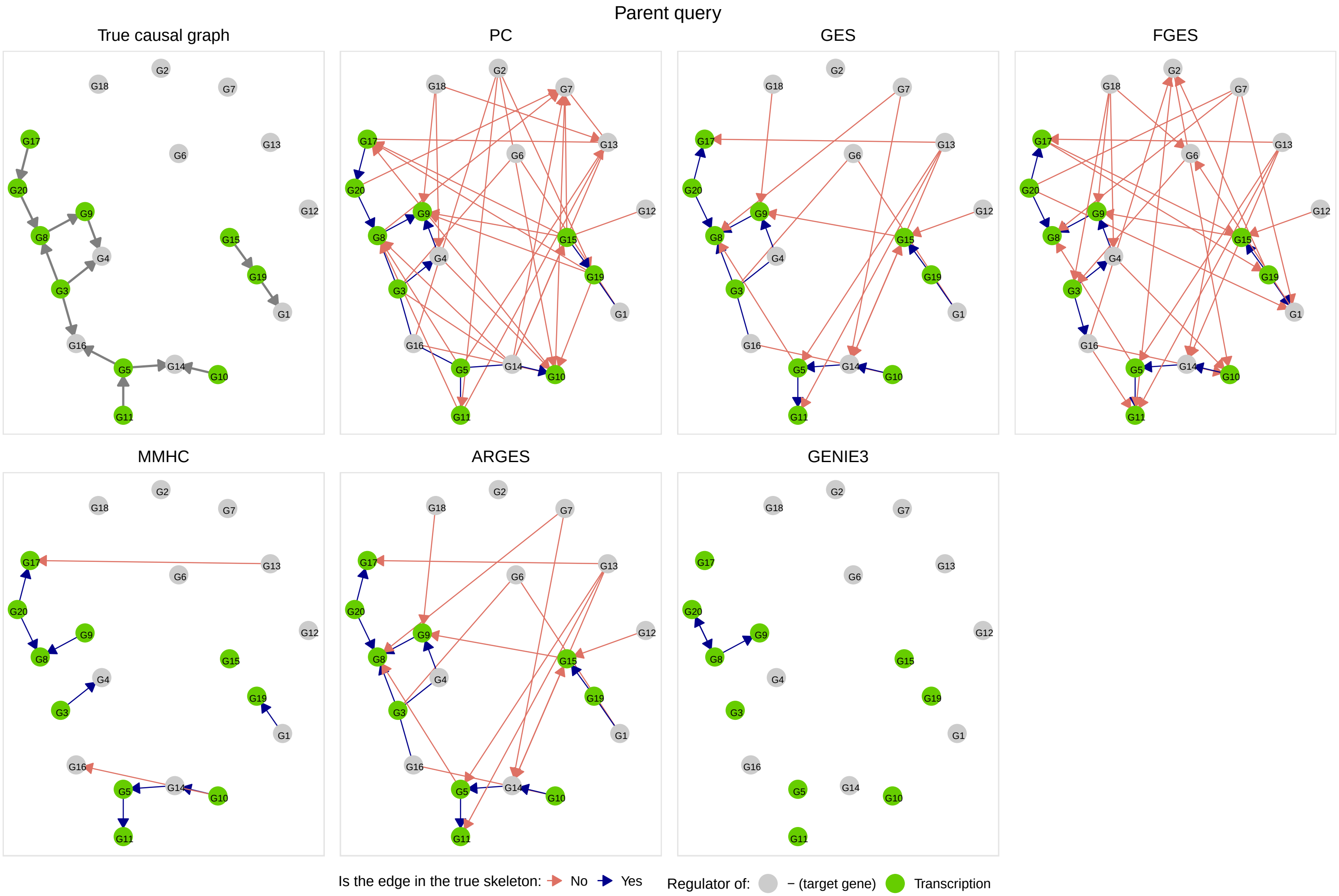


Figure E.1: Number of edges in the inferred causal graphs from different methods, as a function of the value of the tuning parameters. The points show the average number of edges across the 260 simulated datasets, with the vertical bars showing the minimum and maximum values. The size of the points represent the fraction of runs that finished within the 10-minutes limit (i.e. smaller points indicate that more runs exceeded the limit and were interrupted).



Parent query

True causal graph

PC

GES

FGES

MMHC

ARGES

GENIE3

G1
G2
G3
G4
G5
G6
G7
G8
G9
G10
G11
G12
G13
G14
G15
G16
G17
G18
G19
G20
Is the edge in the true skeleton: No Yes
Regulator of: − (target gene) Transcription

Figure E.2: True causal graph for one network from configuration 1, and graph inferred by each method for the parent query. The colour of the edges represent whether the edge is present in the true network (dark blue if yes, light red if not). The colour of the nodes in the graph show the the biological role of the genes in the network (green for transcription regulators and gray for target genes).

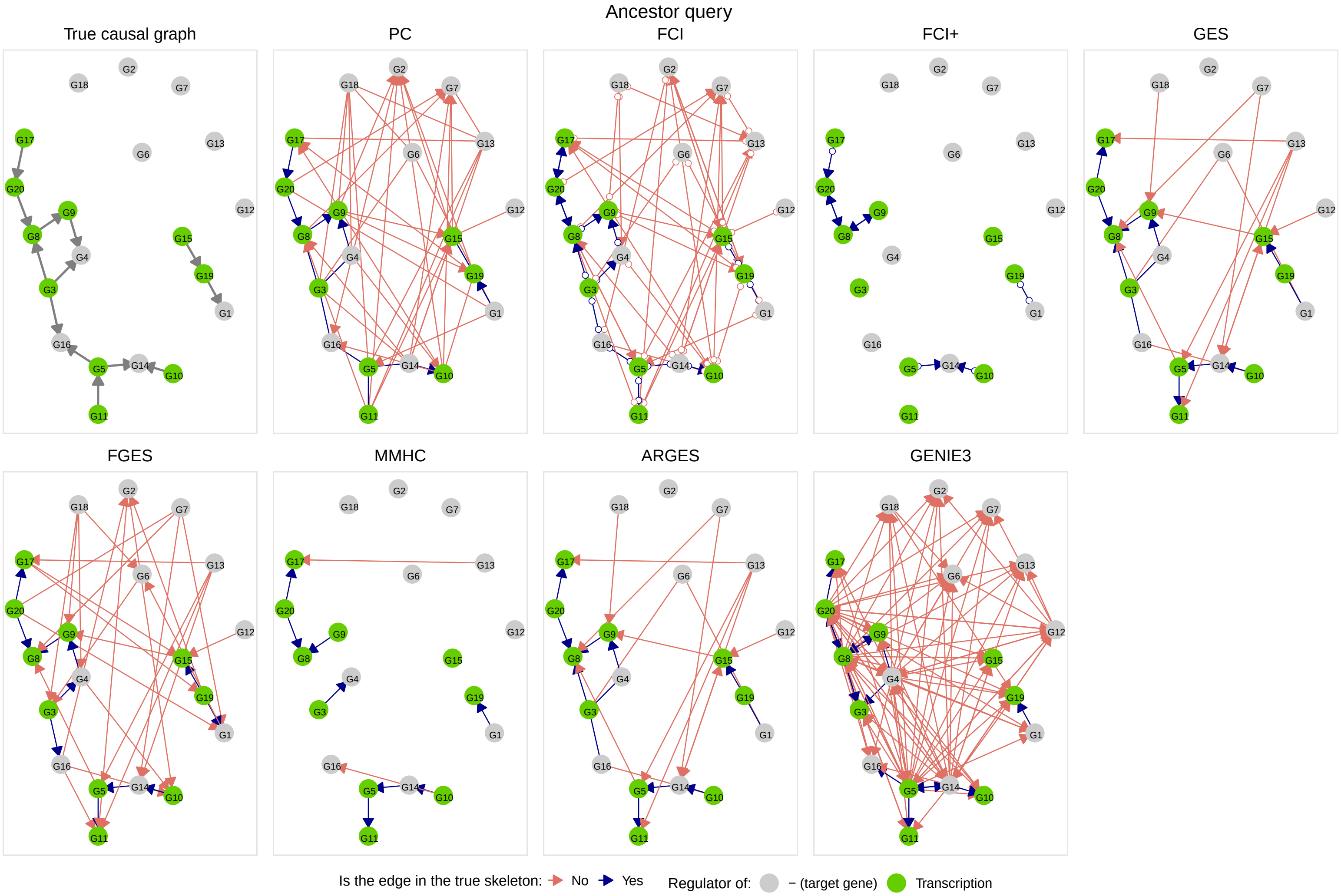


Figure E.3: True causal graph for one network from configuration 1, and graph inferred by each method for the ancestor query. The colour of the edges represent whether the edge is present in the true network (dark blue if yes, light red if not). The colour of the nodes in the graph show the the biological role of the genes in the network (green for transcription regulators and gray for target genes).

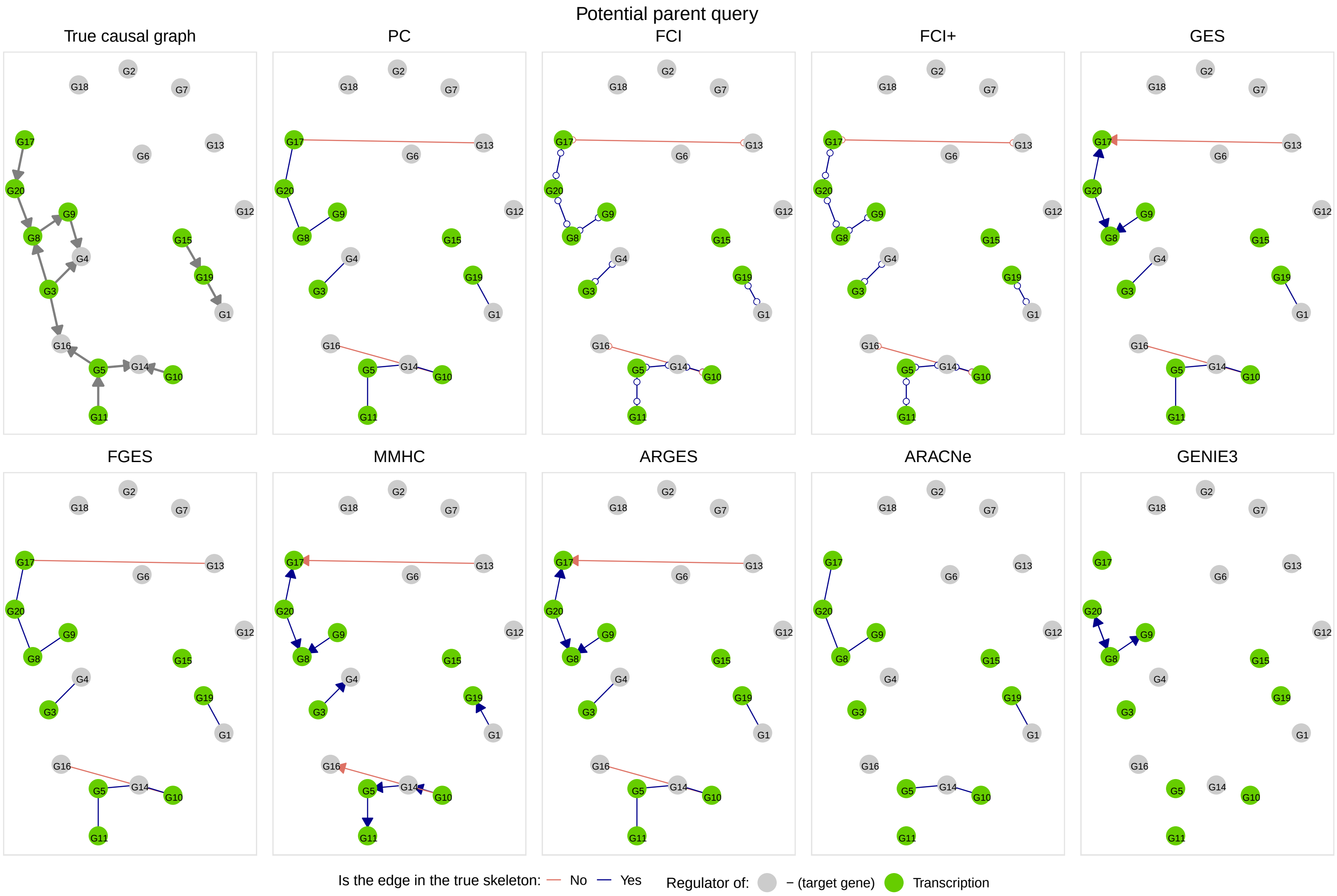

Potential parent query
True causal graph
PC
FCI
FCI+
GES
FGES
MMHC
ARGES
ARACNe
GENIE3
Is the edge in the true skeleton: No Yes
Regulator of: − (target gene) Transcription

Figure E.4: True causal graph for one network from configuration 1, and graph inferred by each method for the potential parent query. The colour of the edges represent whether the edge is present in the true network (dark blue if yes, light red if not). The colour of the nodes in the graph show the the biological role of the genes in the network (green for transcription regulators and gray for target genes).

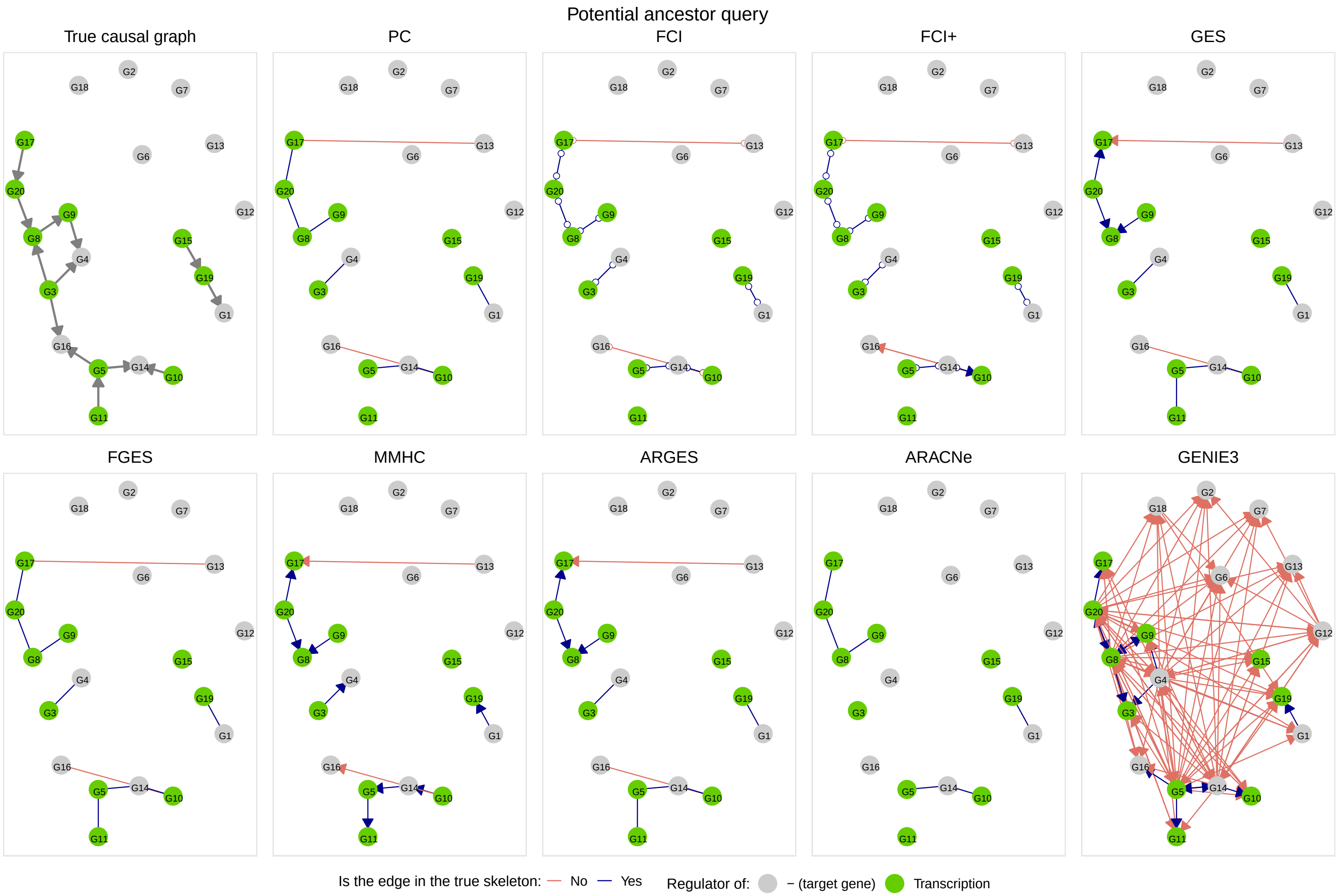


Figure E.5: True causal graph for one network from configuration 1, and graph inferred by each method for the potential ancestor query. The colour of the edges represent whether the edge is present in the true network (dark blue if yes, light red if not). The colour of the nodes in the graph show the the biological role of the genes in the network (green for transcription regulators and gray for target genes).

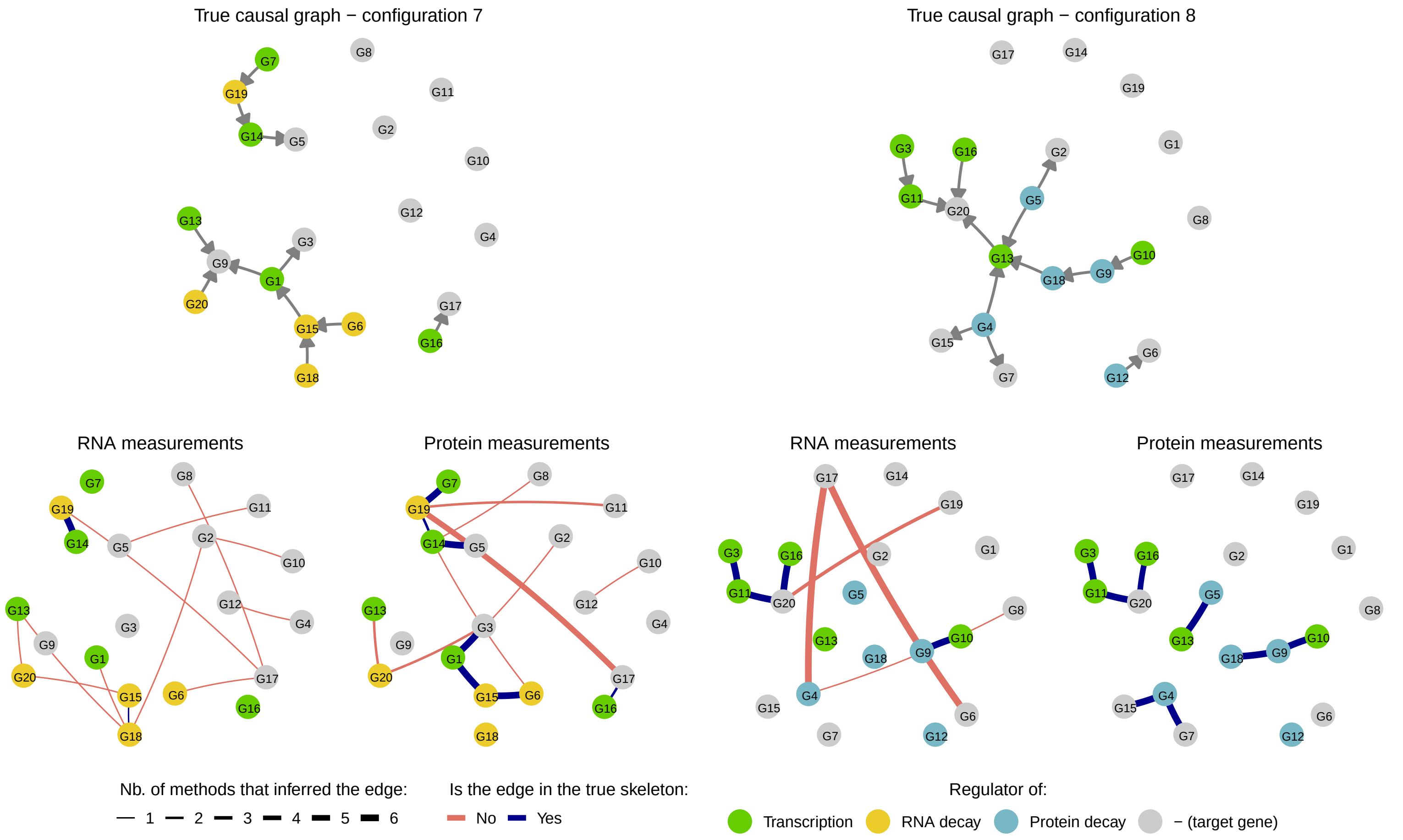
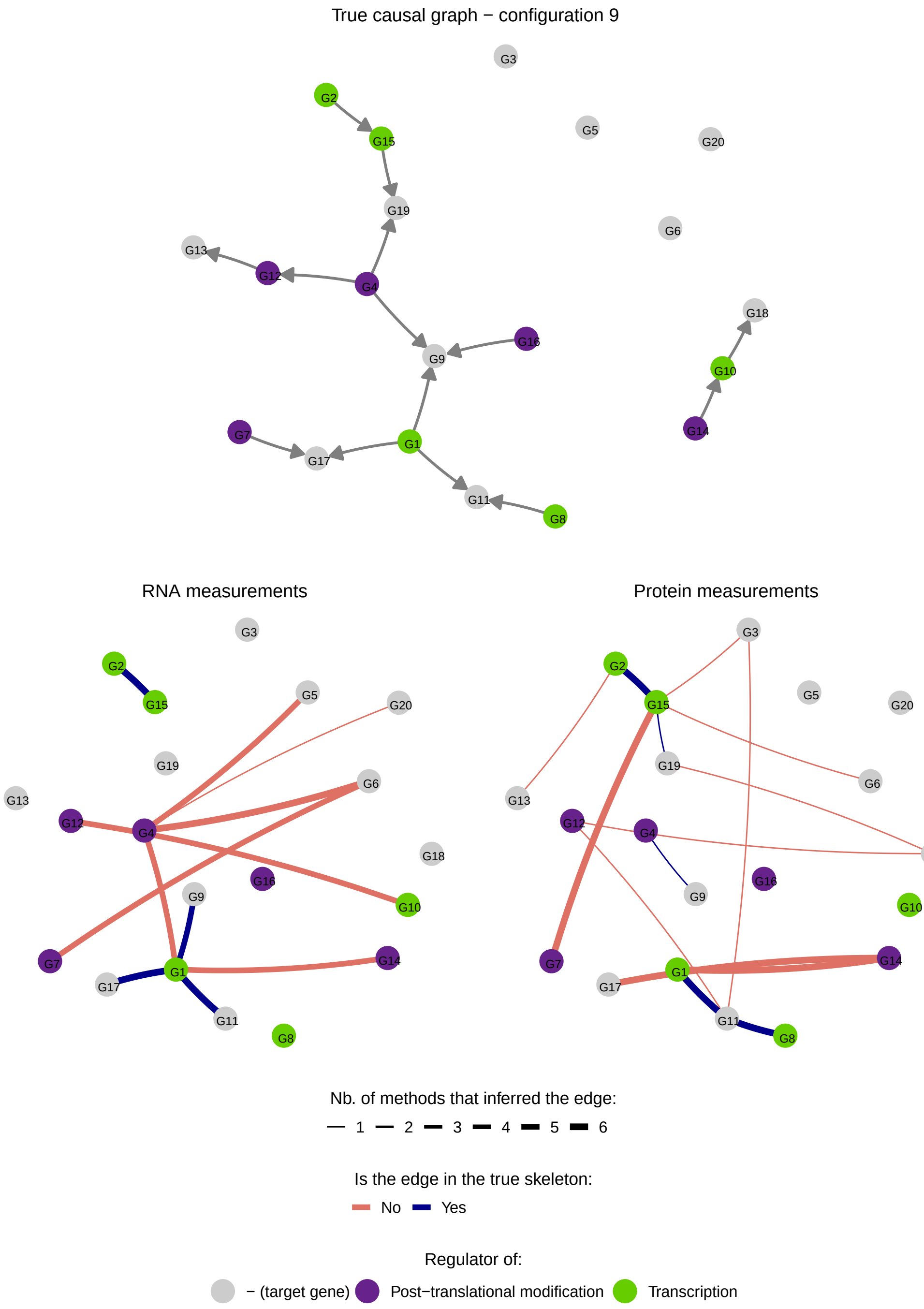


Figure E.6: Two examples of network from configuration 7 (left) and 8 (right), with the consensus skeleton of the graphs inferred by the different methods (except GES, ARGES and GENIE3 that were excluded from this plot), when using RNA and protein measurements for the causal inference. The colour of the edges indicates whether the edge is present in the true skeleton (dark blue if yes, light red if not), and the width of the edges indicate the number of methods that inferred the presence of the edge. The biological role of the genes in the network are indicated in colours, with regulators of transcription highlighted in green, regulators of RNA decay in yellow, regulators of protein decay in blue, while target genes are shown in gray.


True causal graph − configuration 9

RNA measurements

Protein measurements

Figure E.7: Two examples of network from configuration 9, with the consensus skeleton of the graphs inferred by the different methods (except GES, ARGES and GENIE3 that were excluded from this plot), when using RNA and protein measurements for the causal inference. The colour of the edges indicates whether the edge is present in the true skeleton (dark blue if yes, light red if not), and the width of the edges indicate the number of methods that inferred the presence of the edge. The biological role of the genes in the network are indicated in colours, with regulators of transcription highlighted in green, regulators of post-translational modification in purple, while target genes are shown in gray.

# Appendix F

# Supplementary Material for Chapter 4

## F.1 RNA extraction protocol

- Add <100 mg ground potato tuber to 1.7ml tube.
- Add 500 $\mu$l ice-cold Plant RNA Reagent and vortex well. Incubate horizontally with occassional mixing for 5 minutes at room temperature.
- Spin at 12,000$\times g$ for 2 minutes at room temperature. Transfer supernatant (around 500 $\mu$l) to a clean tube.
- Add 100 $\mu$l of 5M NaCl and mix by tapping the tubes.
- Add 300 $\mu$l chloroform:isoamyl alcohol (24:1) and mix thoroughly by inversion.
- Spin at 12,000 $\times g$ for 10 minutes at 4°C. Transfer upper, aqueous phase (around 500 $\mu$l) to clean tube.
- Add 500 $\mu$l of 4M LiCl; incubate at -20°C for 2-3 hours.
- Spin at 12,000 $\times g$ for 30 minutes at 4°C. Remove all supernatant with a pipette, spinning and using a finer tip if required.
- Wash twice with 1ml 80% EtOH. After second wash, remove all remaining EtOH with a fine tip and allow to air-dry around 5 minutes.
- Resuspend in 25$\mu$l of 1mM DTT and 0.75 $\mu$l RNaseOUT and flick the mix. Leave on ice (or in fridge) for 30 minutes to resuspend fully.
- Flick/vortex to resuspend. Spin 1 minute at room temperature and transfer to clean tube.

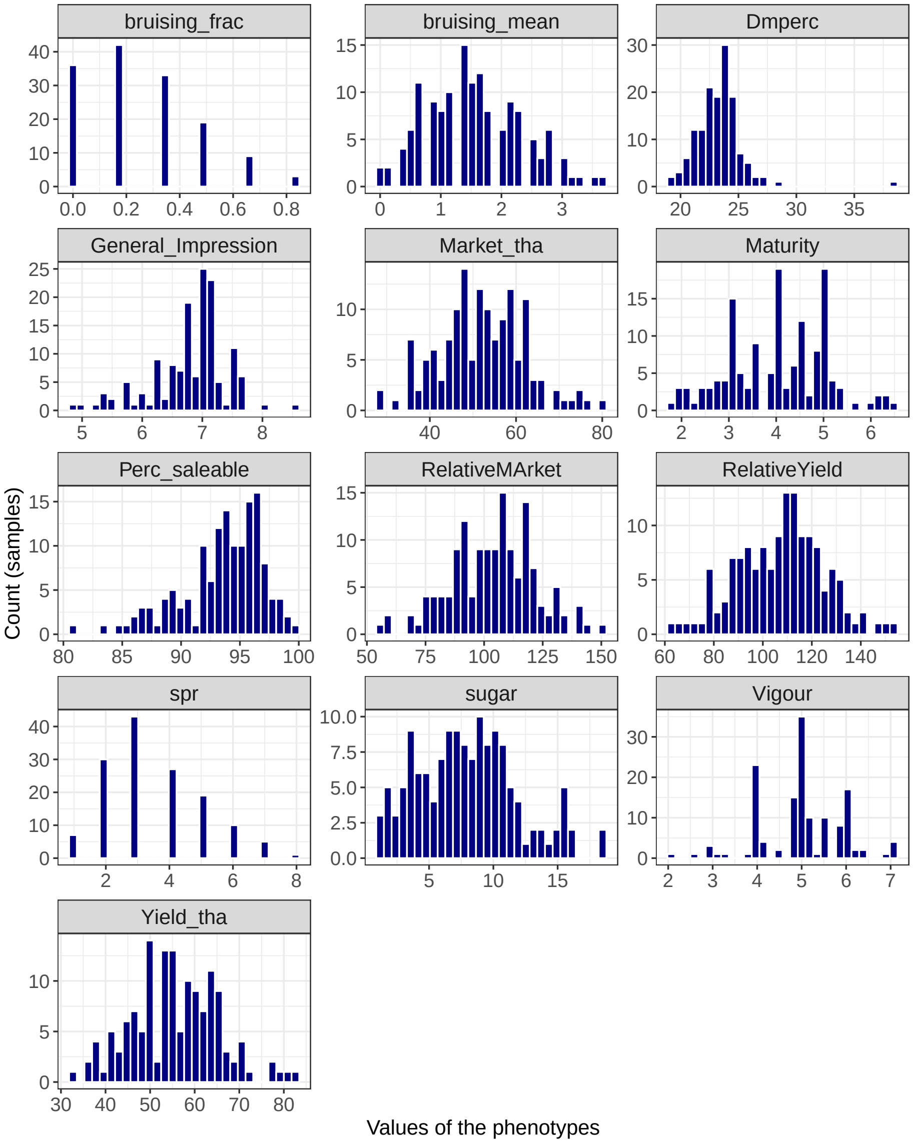


Figure F.1: Distribution of the measured phenotypes over 142 samples.

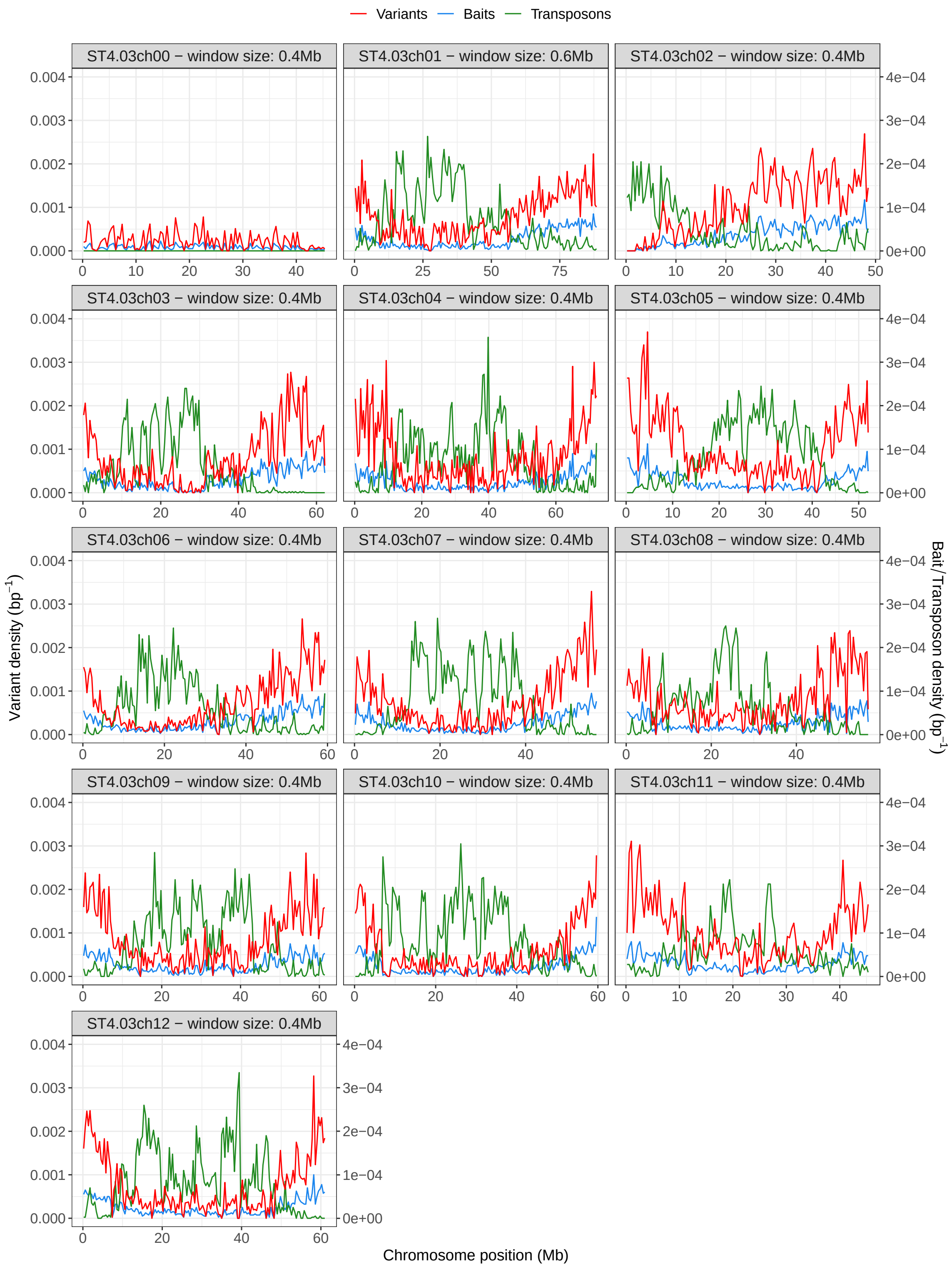


Figure F.2: Distribution of 602,955 variants (red), 20,035 baits (blue) and 53,242 transposons (green) density across the 13 chromosomes. The density is computed by regions of size indicated at the top of each plot.

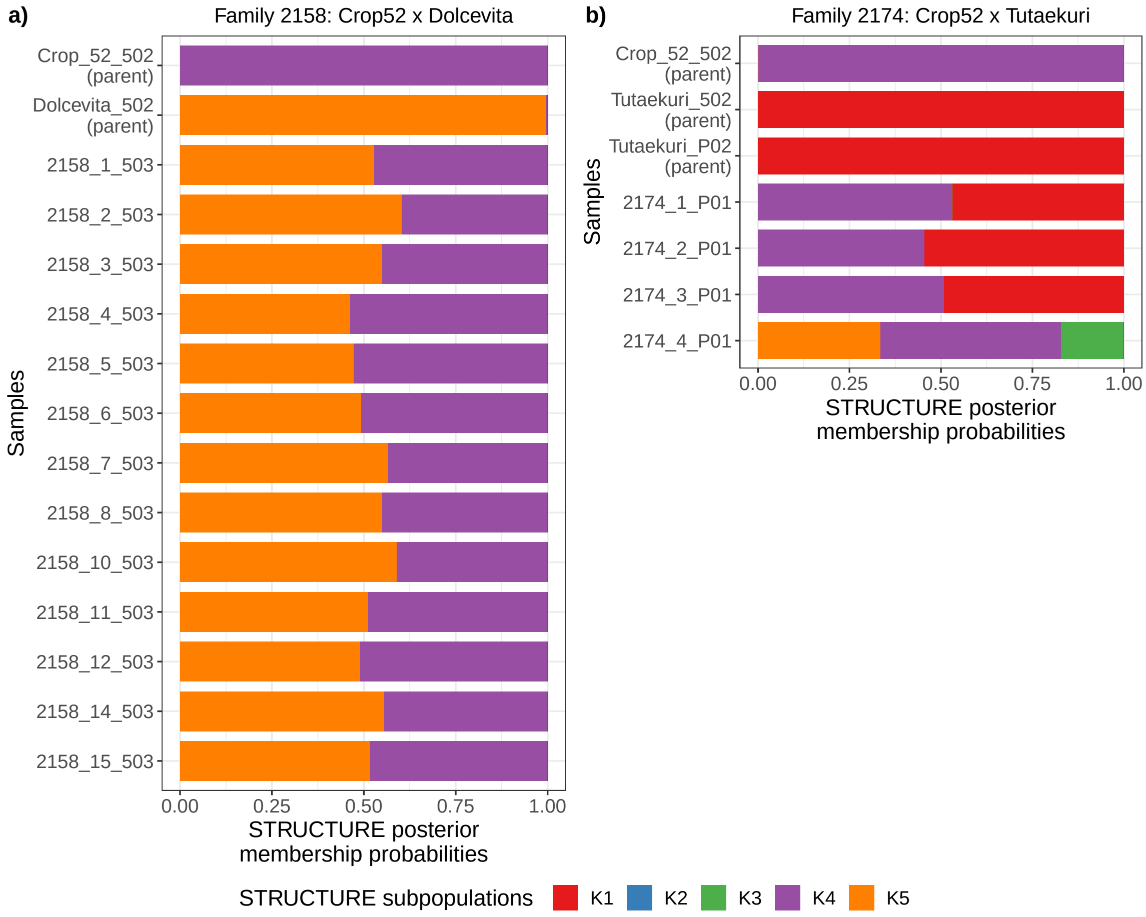


Figure F.3: Population structure uncovered with STRUCTURE for two half-sibling families. The posterior membership probability of the parents and progeny samples for the five subpopulations identified with STRUCTURE are plotted for a) family 2158 and b) family 2174. In both cases, the progeny samples membership probability profile reflects those of the parents with a 1:1 ratio, except for progeny sample 2174_4 which is probably a misslabeled sample.

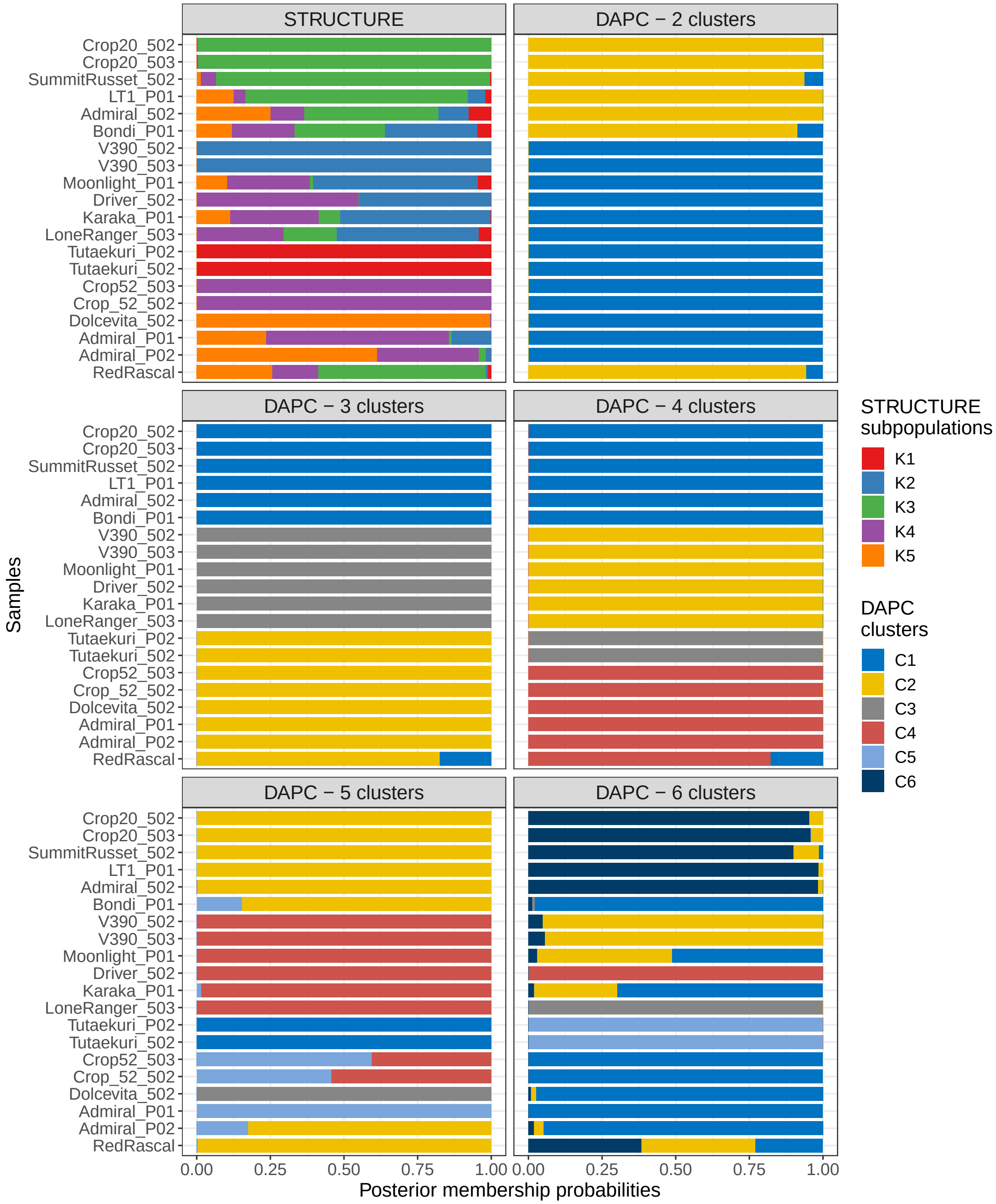


Figure F.4: Population structure uncovered with STRUCTURE and DAPC for parent samples. The posterior membership probabilities of the parent samples are displayed for the five subpopulations identified with STRUCTURE (top left panel) and for a number of clusters varying from 2 to 6 with DAPC (remaining panels).

Table F.1: Markers detected as significant per phenotype with the different GWAS settings. The estimated marker score and effect on the phenotype are indicated. Note that with the general model the marker effect is not computed. If the marker is found within a gene, the ensembl ID and description (if available) of the gene are provided.

| Chromosome | Position | GWAS setting | Score | Effect | Found in gene | Gene description |
|---|---|---|---|---|---|---|
| **Dmperc** | | | | | | |
| ST4.03ch01 | 61,773,721 | K + Q_DAPC - additive | 7.51 | 1.63 | PGSC0003DMG400008978 | Xyloglucanase inhibitor |
| | 61,773,810 | K + Q_DAPC - additive | 7.66 | 1.86 | | |
| ST4.03ch02 | 12,111,253 | K + Q_DAPC - 2-dom-ref | 6.52 | 3.82 | PGSC0003DMG400003174 | Protein AFR |
| | | K - 2-dom-ref | 6.95 | 3.92 | | |
| | 15,284,228 | K + Q_STRUCTURE - 2-dom-ref | 6.63 | 3.88 | PGSC0003DMG400017862 | Potassium transporter 11 |
| | | K + Q_DAPC - 2-dom-ref | 6.70 | 3.86 | | |
| | | K - 2-dom-ref | 7.29 | 3.97 | | |
| | 22,602,384 | K + Q_STRUCTURE - 2-dom-ref | 6.65 | 3.89 | PGSC0003DMG400003324 | Squalene epoxidase |
| ST4.03ch04 | 25,442,106 | K + Q_DAPC - additive | 7.45 | 1.76 | - | |
| | | K - additive | 7.81 | 1.79 | | |
| | | K + Q_STRUCTURE - additive | 7.91 | 1.85 | | |
| ST4.03ch11 | 39,182,523 | K + Q_STRUCTURE - 2-dom-alt | 7.02 | 3.81 | | |
| | | K + Q_DAPC - 2-dom-alt | 7.55 | 3.93 | | |
| ST4.03ch12 | 52,336,153 | K + Q_DAPC - additive | 7.13 | 1.21 | PGSC0003DMG400011612 | ABC transporter family protein |
| | | K + Q_STRUCTURE - additive | 7.26 | 1.23 | | |
| | | K - additive | 7.29 | 1.23 | | |
| **General_Impression** | | | | | | |
| | | K - 1-dom-alt | 7.00 | 0.92 | | |
| | | Q_DAPC - 1-dom-alt | 7.27 | 0.95 | | |
| | | Q_STRUCTURE - 1-dom-alt | 7.29 | 0.94 | | |

Table F.1: Markers detected as significant per phenotype with the different GWAS settings. The estimated marker score and effect on the phenotype are indicated. Note that with the general model the marker effect is not computed. If the marker is found within a gene, the ensembl ID and description (if available) of the gene are provided. *(continued)*

| Chromosome | Position | GWAS setting | Score | Effect | Found in gene | Gene description |
|---|---|---|---|---|---|---|
| ST4.03ch06 | 53,234,244 | K + Q_DAPC - 1-dom-alt | 7.34 | 0.96 | PGSC0003DMG400041029 | Bcl-2-associated athanogene |
| | | K + Q_STRUCTURE - 1-dom-alt | 7.43 | 0.97 | | |
| ST4.03ch08 | 53,949,510 | K + Q_DAPC - general | 7.28 | NA | PGSC0003DMG400022756 | Adenosine 3’-phospho 5’-phosphosulfate transporter |
| | | K - general | 7.38 | NA | | |
| | | K + Q_STRUCTURE - general | 7.94 | NA | | |
| ST4.03ch11 | 850,745 | K + Q_DAPC - 1-dom-ref | 6.27 | -0.90 | PGSC0003DMG400013331 | Glutamate decarboxylase isoform2 |
| | | Q_DAPC - 2-dom-ref | 6.55 | -0.74 | | |
| ST4.03ch12 | 58,347,097 | Q_DAPC - additive | 6.82 | -0.27 | PGSC0003DMG400029264 | Anthocyanin acyltransferase |
| **Perc_saleable** | | | | | | |
| ST4.03ch08 | 28,681,951 | K + Q_STRUCTURE - 2-dom-ref | 6.47 | -6.03 | - | |
| | | Naive - 2-dom-ref | 6.79 | -6.03 | | |
| | | Q_STRUCTURE - 2-dom-ref | 7.20 | -6.31 | | |
| **spr** | | | | | | |
| ST4.03ch09 | 58,507,500 | K + Q_DAPC - 2-dom-alt | 6.72 | 1.51 | PGSC0003DMG400036801 | F-box/LRR-repeat protein |
| | | K + Q_STRUCTURE - general | 6.94 | NA | | |
| | | K - 2-dom-alt | 6.97 | 1.53 | | |
| | | K + Q_STRUCTURE - 2-dom-alt | 7.50 | 1.62 | | |
| **Vigour** | | | | | | |
| ST4.03ch07 | 3,858,659 | K + Q_STRUCTURE - 2-dom-alt | 6.80 | -1.09 | PGSC0003DMG400027940 | ATP binding protein |
| | | Q_STRUCTURE - 2-dom-alt | 6.96 | -1.07 | | |

Table F.2: Enrichment of co-expression modules whose eigengene is significantly correlated with the bruising mean score for biological process- and molecular function-related GO terms.

| GO term ID | GO term definition | Enrichment p-value |
|---|---|---|
| **Module 74 (33 genes)** | | |
| GO:0006480 | N-terminal protein amino acid methylation | 0.009 |
| **Module 121 (19 genes)** | | |
| GO:0031348 | negative regulation of defense response | 0.015 |
| GO:0060548 | negative regulation of cell death | < 0.001 |
| **Module 182 (10 genes)** | | |
| GO:0102499 | SHG alpha-glucan phosphorylase activity | < 0.001 |
| GO:0102250 | linear malto-oligosaccharide phosphorylase activity | < 0.001 |
| GO:0008184 | glycogen phosphorylase activity | < 0.001 |
| GO:0030170 | pyridoxal phosphate binding | 0.019 |
| GO:0004645 | 1,4-alpha-oligoglucan phosphorylase activity | < 0.001 |
| GO:0042802 | identical protein binding | < 0.001 |

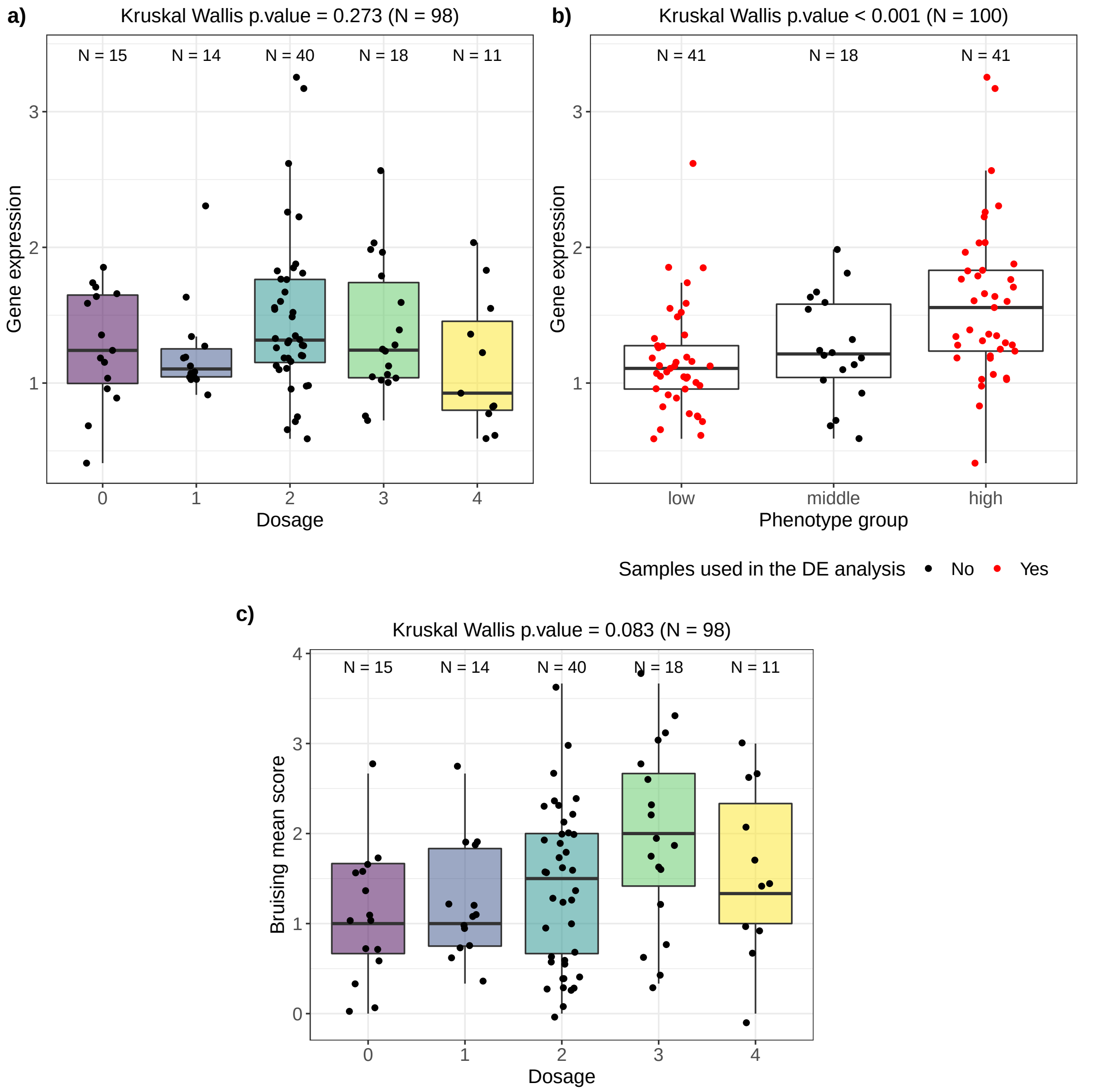


Figure F.5: Relationship between the dosage of the high-scoring marker ST4.03ch02_41355270 (for the bruising fraction phenotype), the expression of the gene PGSC0003DMG400026406 in which it is found (RPKM counts), and the bruising mean score. a) The gene expression for the samples is plotted against their dosage for the considered marker. b) The gene expression for the samples is plotted against the phenotype group in which they belong (low/middle/high bruising mean score). Samples used for the differential expression analysis are highlighted in red. c) The bruising mean score of the samples is plotted against their dosage for the considered marker.

# Appendix G

# Supplementary Material for Chapter 5

## G.1 Metabolomics data generation protocol

### G.1.1 Samples preparation

Freeze dried samples were weighed into Eppendorf tubes and extracted with 70% ethanol, 30% water at 10mg/ml equivalent and extensively vortex mixed. They were centrifuged at 14000 $\times g$ for 5 minutes. A 400$\mu$l aliquot of the supernatant was filtered with a Single Step® vial containing a 0.22$\mu$m PVDF (Thompson™ Part No. 65531-200) filter. Vials were held at 8O°C in the LCMS sample chamber before injection.

Quality Control (QC) samples were prepared by taking an equal volume of every sample to make a multi-mix. To this mix were added stable $^{13}$C isotopes at a 1:10 dilution from stock of Cambridge Isotopes Lab Metabolomics QC Standard 1, Cat. Number MSK-QC-1 and Standard 2, Cat. Number MSK-QC2-1. This resulted in each $^{13}$C compound being at 0.4$\mu$g/ml in the QC mix except $^{13}C_{11}$-L-Tryptophan which was at 4$\mu$g/ml.

### G.1.2 Liquid Chromatograph Mass Spectrometry (LCMS) conditions

The LCMS system consisted of a Thermo Scientific™ (San Jose, CA, USA) Q Exactive™ Plus Orbitrap coupled with a Vanquish™ UHPLC system (Binary Pump H, Split Sampler HT, Dual Oven); calibrated immediately prior to sample analysis batch with Thermo™ premixed solutions (Pierce™ LTQ ESI Positive and negative ion calibration solutions, catalogue numbers: 88322 and 88324 respectively).

#### Aqueous normal phase conditions (H)

A 2 $\mu$l aliquot of each prepared extract was separated with a mobile phase consisting of 0.1 % formic acid in acetonitrile (A) and 5mM ammonium acetate in water (B) by normal phase chromatography (Hypersil Gold HILIC 1.9$\mu$m, 100mm x2.1mm, P/N:26502-102130) maintained at 55 °C with a flow rate of 400 $\mu$l/min. A gradient was applied: 0-1 min/5% B, linear increase to 12 min/98% B, isocratic 16min/98% B, equilibration 16-17 min/5% B, isocratic to end 20min/5% B.

#### Reverse phase conditions (C18)

A 2 $\mu$l aliquot of each prepared extract was separated with a mobile phase consisting of 0.1 % formic acid in type 1 water (A) and 0.1 % formic acid in acetonitrile (B) by reverse phase chromatography (Accucore Vanquish C18 1.5$\mu$m, 100mm x2.1mm, P/N: 27101-102130, Thermo Scientific) maintained at 40°C with a flow rate of 400$\mu$l/min. A gradient was applied: 0-1 min/0% B, linear increase to 7 min/50% B, linear increase to 8min/98% B, isocratic to 11 min/98% B, equilibration 11-12min/0% B, isocratic to end 17 min/0% B.

The eluent from (H) and (C18) chromatography was scanned from 0.5-16 and 0.4-11.5 minutes respectively by API-MS (Orbitrap) with heated electrospray ionisation (HESI) at 350°C in the negative and positive mode with capillary temperature of 320°C. Data were acquired for precursor masses from m/z 80–1200 amu (H) and m/z 100-1500(C18) at 70K resolution (AGC target 3e6, maximum IT 100ms, profile mode) with data dependent ms/ms for product ions generated by normalised collision energy (NCE:35, 45, 65) for C18 and (NCE:25, 45, 75) for (H) at 17.5K resolution (TopN 10, AGC target 2e5, Maximum IT 50ms, Isolation 1.4 m/z).

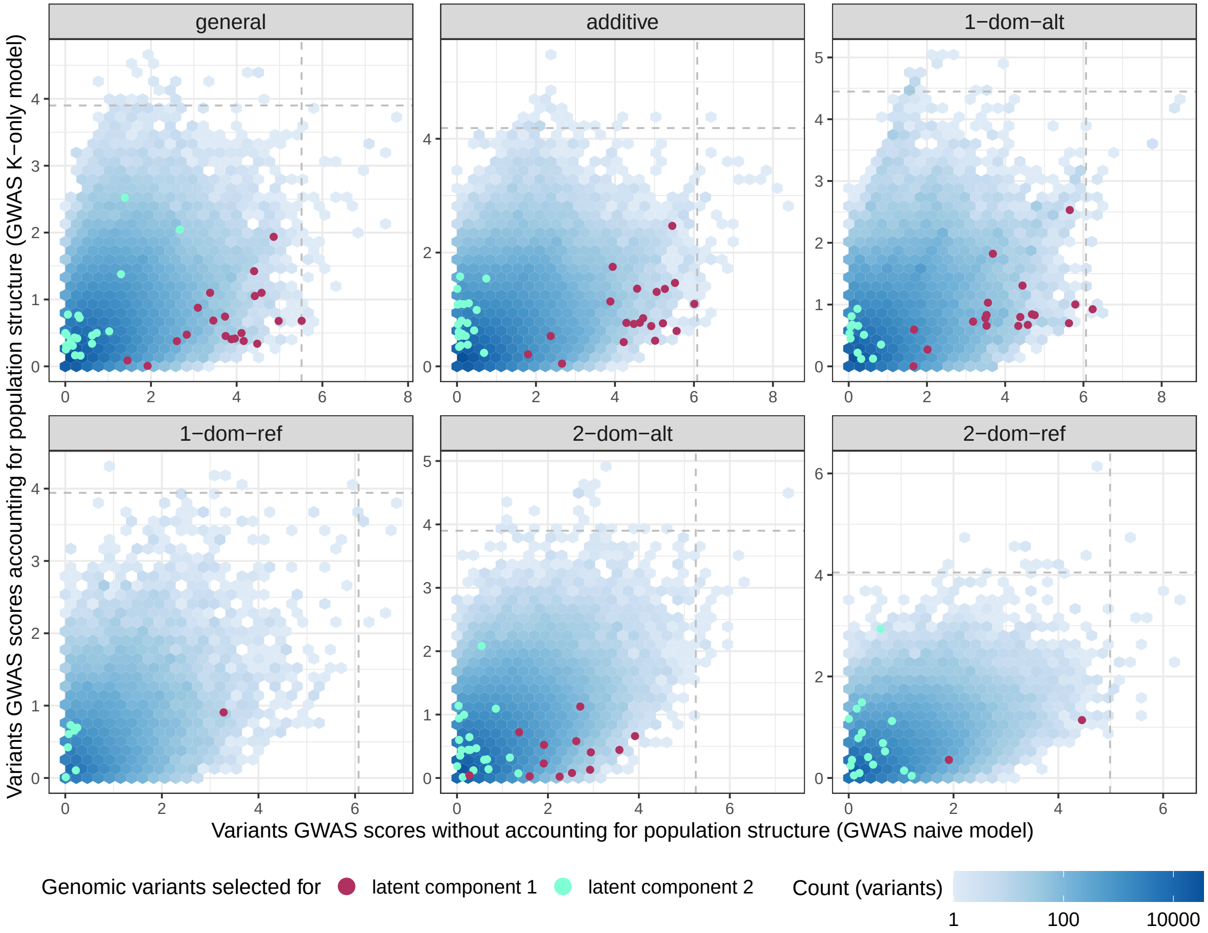


Figure G.1: Distribution of the variants GWAS scores without (x-axis) and with (y-axis)correcting for population structure, across the different genetic models tested. The variants selected with the full DIABLO analysis are shown as points on top of the hex-plot. Their colours indicate whether they are retained for the first (maroon) or second (light green) latent component. The dotted lines show the scores above which variants are considered as high-scoring markers.

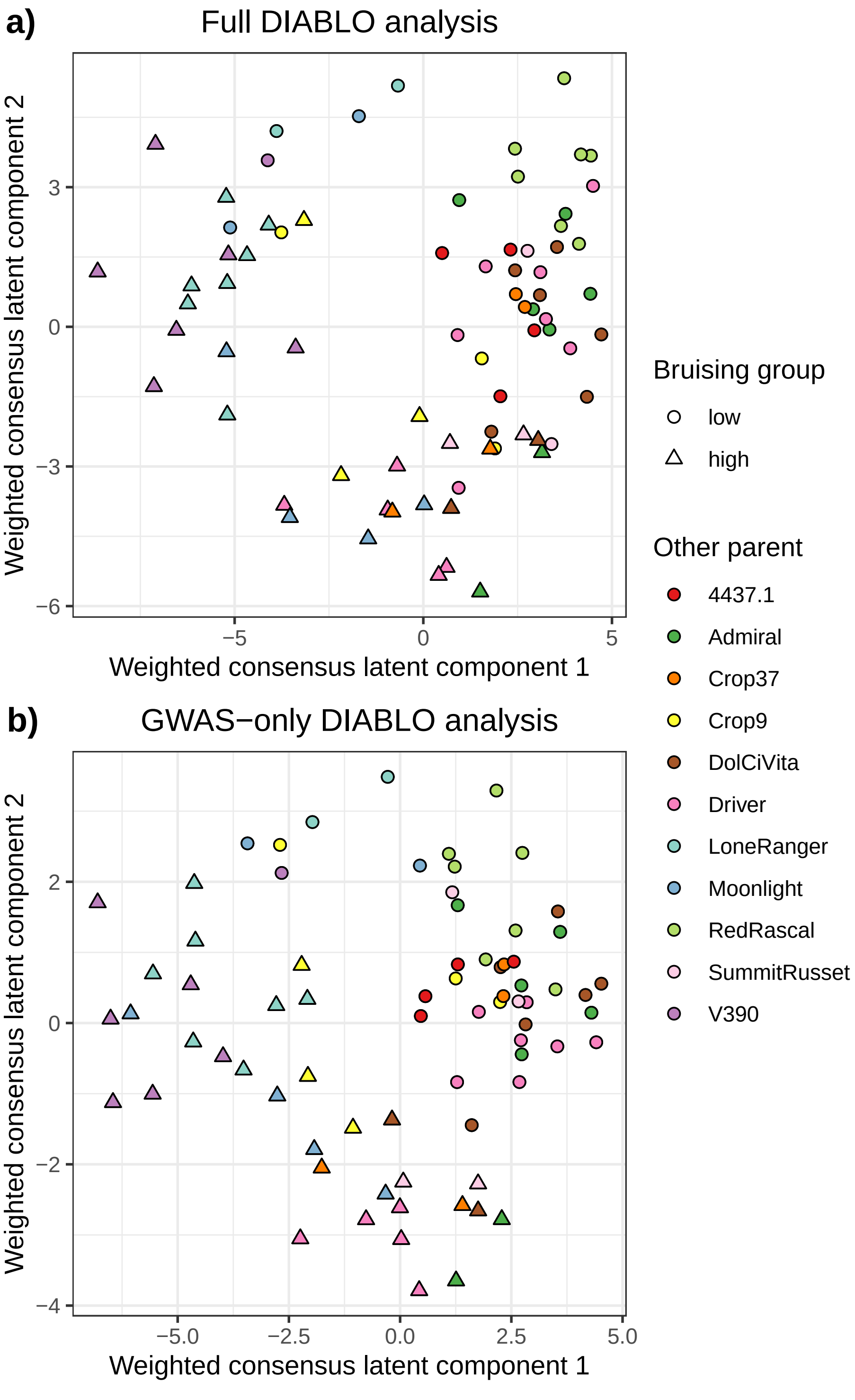


Figure G.2: Weighted consensus reduced space of a) the full DIABLO analysis and b) the GWAS-only DIABLO analysis. Each point represents one observation (i.e. a biological replicate of one sample), with their shape indicating whether they belong to the low bruising group (circles) or the high bruising group (triangles). The colour of the points represents the parent (other than Crop52) of the samples.

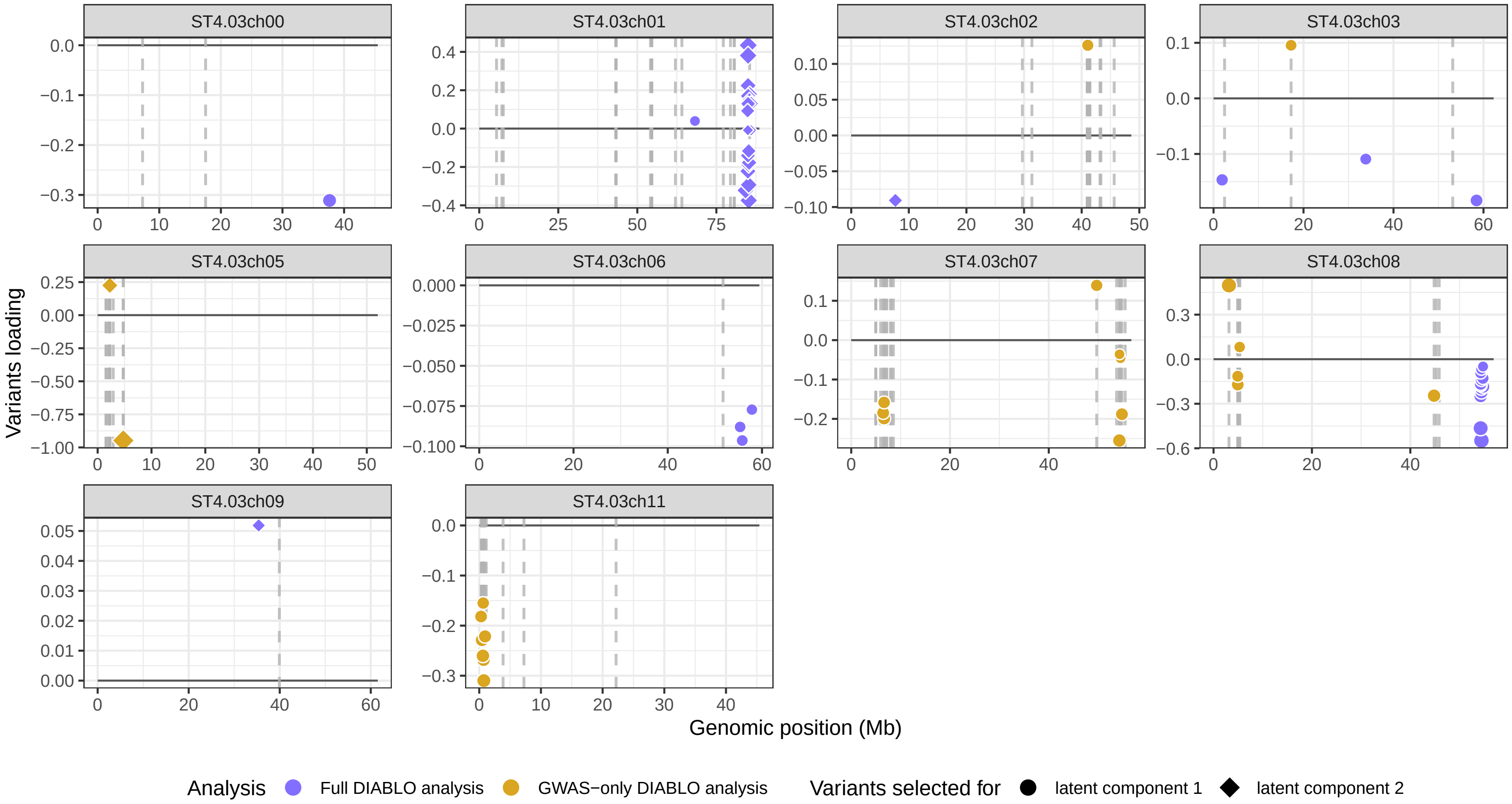


Figure G.3: Genomic position across the chromosomes of the genomic variants selected with the full (blue) and GWAS-only (yellow) DIABLO analyses. Variants selected for the first latent components are shown as circles, and those selected for the second latent component as diamonds.

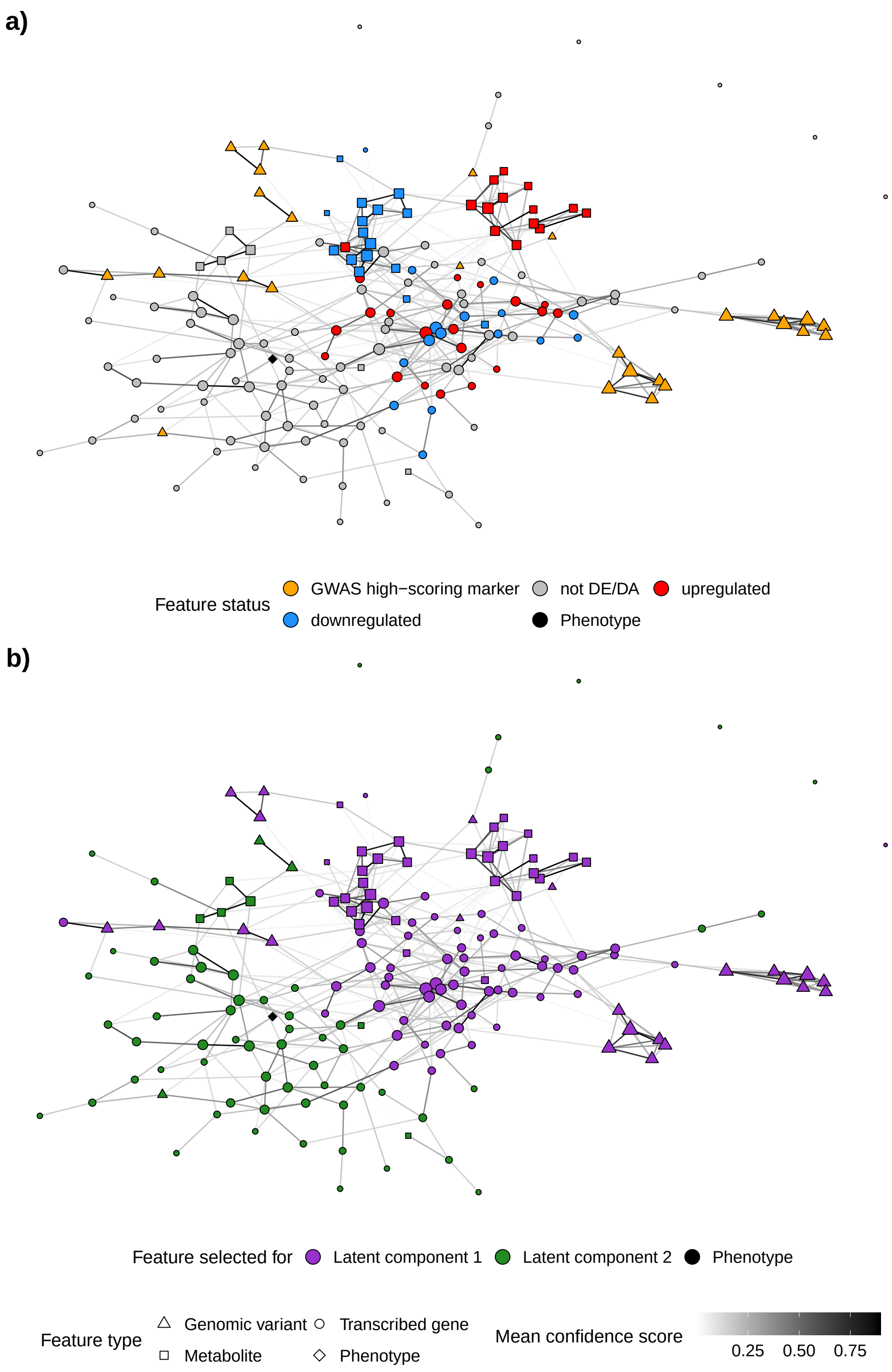


Figure G.4: Consensus skeleton of the graphs inferred with the nine causal and network inference methods. The nodes represent the features selected with DIABLO (triangles: genomic variants, circles: transcribed genes, squares: compounds, diamond: bruising mean score). Their size indicate the strenght (i.e. weighted degree) of the nodes. The colour of the edges represent the mean confidence score over all methods. a) Features coloured by status in the single-omics analysis. b) Features coloured by the latent component for which they were selected with DIABLO.

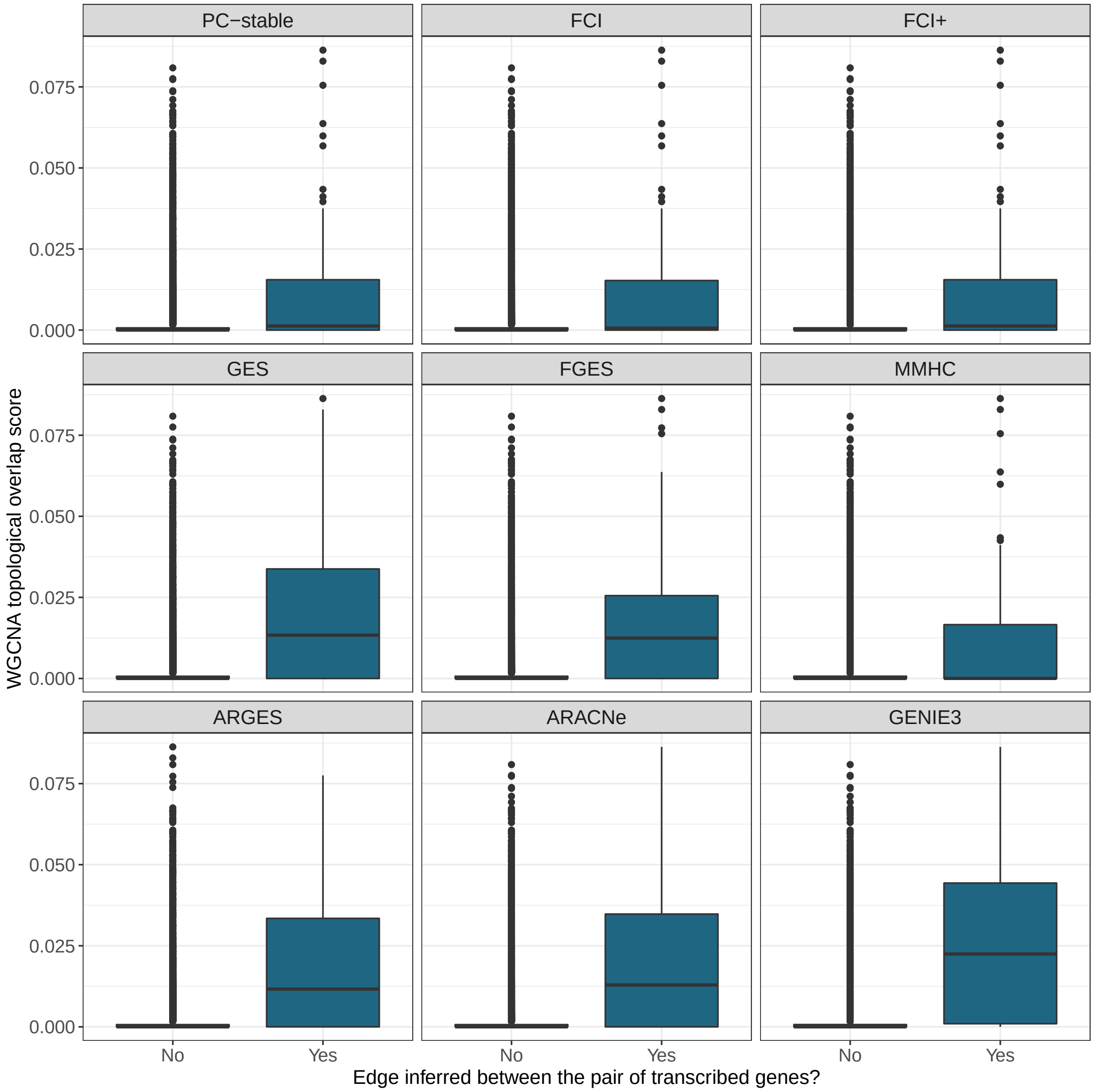


Figure G.5: Comparison for the transcriptomics dataset of the WGCNA topological overlap score distributions between pairs of transcribed genes that are found causally linked in the graph inferred by each causal and network inference method and those that are not linked.

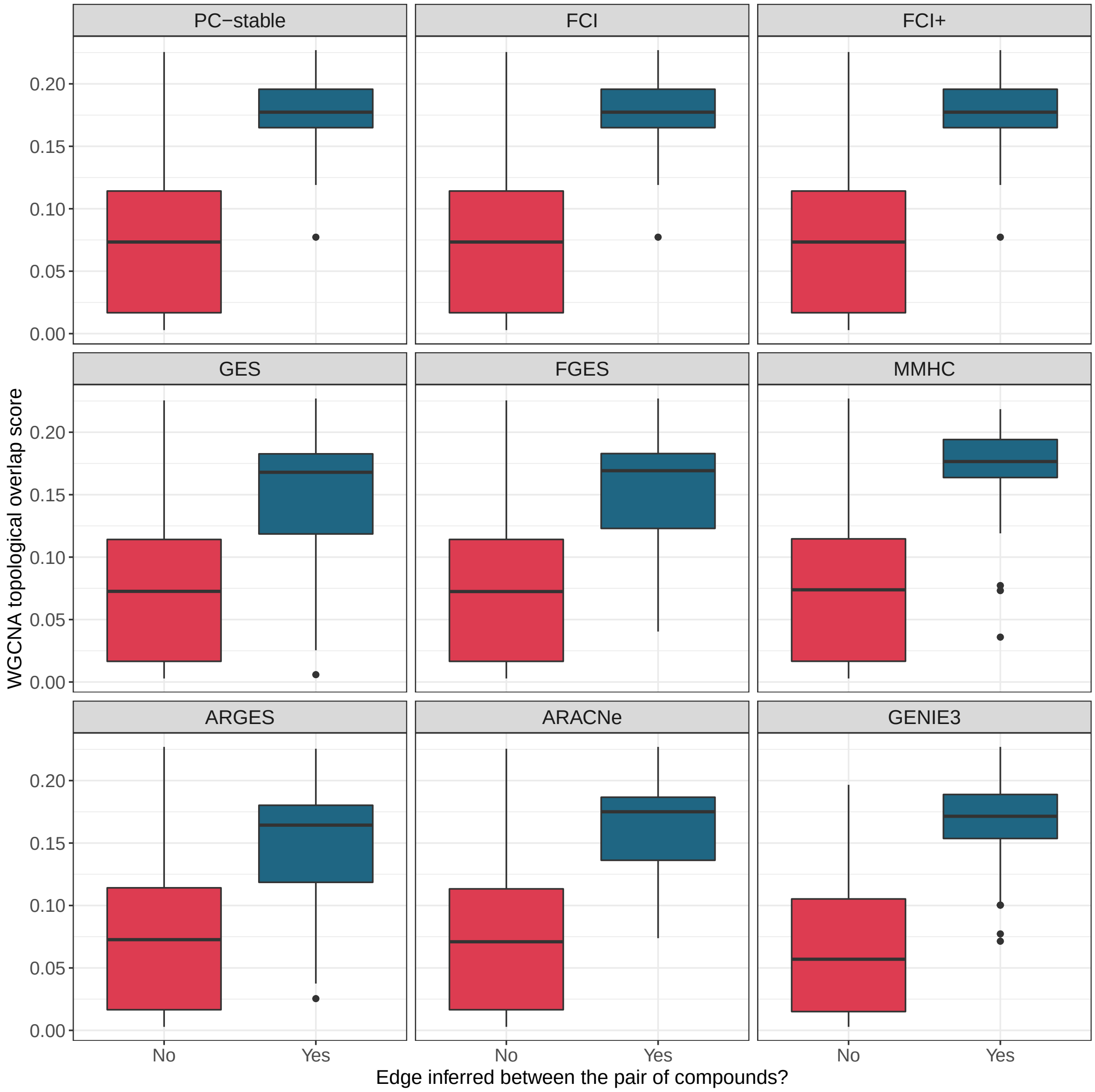


Figure G.6: Comparison for the metabolomics dataset of the WGCNA topological overlap score distributions between pairs of genes that are found causally linked in the graph inferred by each causal and network inference method and those that are not linked.

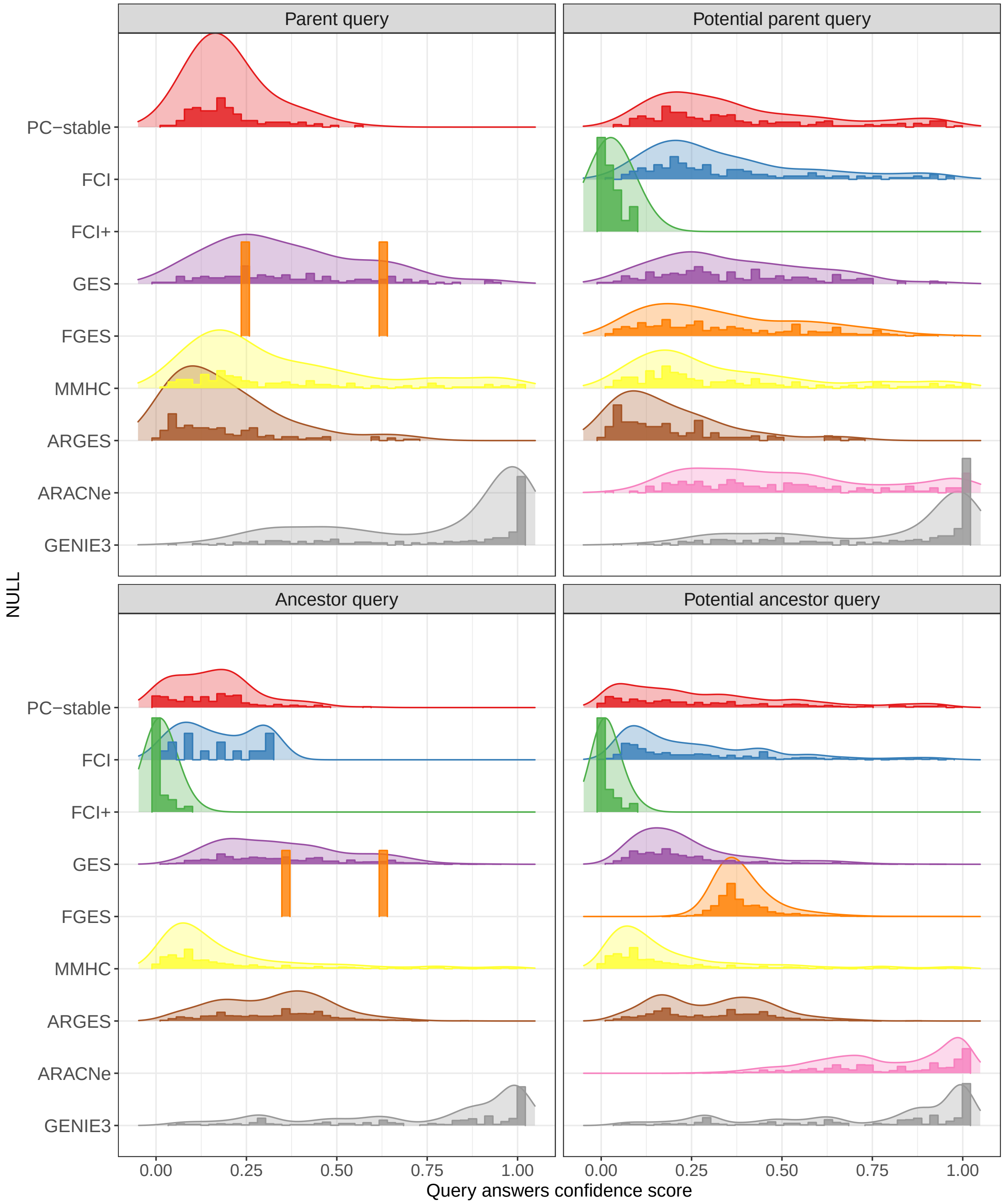


Figure G.7: Distribution of the confidence score of positive answers to each causal query for the different causal and network inference methods.

Table G.1: Edges in the consensus skeleton of inferred graphs (with blacklist) with a high average confidence score across the causal and network inference methods. For comparison, the mean confidence score of the edges across the four considered methods (PC-stable, FGES, MMHC and GENIE3) without blacklist is presented. When possible, the possible pathway in which the molecules are involved is indicated. Genomic variants are represented as follows: *chromosome, genomic position*; transcribed genes as: *description - chromosome, genomic position (Ensembl ID) - whether it is differentially expressed or not*; metabolic compounds as: *description if identified - formula if identified, molecular weight - whether it is differentially abundant or not*. Note that most chemical formulas have been automatically generated by the metabolomics analysis software.

| Edge between | And | Mean confidence score - blacklist | Mean confidence score - no blacklist | Possible pathway |
|---|---|---|---|---|
| 9-HOTrE (9-hydroxy-10E,12Z,15Z-octadecatrienoic acid) - $C_{18}H_{30}O_3$, 294.22 g/mol - upregulated | (10E,12Z)-9-Hydroxy-10,12-octadecadienoic acid - $C_{18}H_{32}O_3$, 296.23 g/mol - upregulated | 1.00 | 1.00 | $\alpha$-linoleic acid metabolism |
| $C_{20}H_{28}O_3P_2$, 378.15 g/mol - upregulated | $C_9H_{23}N_{11}O_2P_2$, 379.15 g/mol - upregulated | 1.00 | 1.00 | - |
| $C_{18}H_{26}N_4O_3S$, 378.17 g/mol - upregulated | $C_{20}H_{30}O_3P_2$, 380.17 g/mol - upregulated | 1.00 | 1.00 | - |
| $C_{17}H_{20}N_{10}O_2S$, 428.15 g/mol - upregulated | 431.15 g/mol - upregulated | 1.00 | 1.00 | - |
| pigment from marker pen - 265.16 g/mol - downregulated | 265.66 g/mol - downregulated | 1.00 | 1.00 | - |
| L-Glutathione (reduced) - $C_{10}H_{17}N_3O_6S$, 307.08 g/mol - downregulated | L-Glutathione (reduced) - $C_{10}H_{17}N_3O_6S$, 307.08 g/mol - downregulated | 1.00 | 1.00 | Glutathione metabolism |
| Leucine-rich repeat - ST4.03ch00, 23.1Mb (PGSC0003DMG400010887) - upregulated | Cc-nbs-lrr resistance protein - ST4.03ch10, 59Mb (PGSC0003DMG402011427) - not DE | 1.00 | 1.00 | Disease/Pathogen resistance |
| Breakdown product of Glutathione - $C_{13}H_6O$, 178.04 g/mol - downregulated | $C_{13}H_{17}N_3O_2P_2$, 309.08 g/mol - downregulated | 1.00 | 1.00 | Glutathione metabolism |
| Peptide transporter - ST4.03ch06, 56.1Mb (PGSC0003DMG400006606) - upregulated | Peptide transporter - ST4.03ch00, 38Mb (PGSC0003DMG400022107) - upregulated | 1.00 | 1.00 | - |
| Non-symbiotic hemoglobin - ST4.03ch01, 85.5Mb (PGSC0003DMG400025176) - not DE | Enoyl-CoA-hydratase - ST4.03ch01, 85.1Mb (PGSC0003DMG403025826) - not DE | 0.99 | 0.99 | Fatty acid-related |

Table G.1: Edges in the consensus skeleton of inferred graphs (with blacklist) with a high average confidence score across the causal and network inference methods. For comparison, the mean confidence score of the edges across the four considered methods (PC-stable, FGES, MMHC and GENIE3) without blacklist is presented. When possible, the possible pathway in which the molecules are involved is indicated. Genomic variants are represented as follows: *chromosome, genomic position*; transcribed genes as: *description - chromosome, genomic position (Ensembl ID) - whether it is differentially expressed or not*; metabolic compounds as: *description if identified - formula if identified, molecular weight - whether it is differentially abundant or not*. Note that most chemical formulas have been automatically generated by the metabolomics analysis software. *(continued)*

| Edge between | And | Mean confidence score - blacklist | Mean confidence score - no blacklist | Possible pathway |
|---|---|---|---|---|
| $C_9H_8O_2$, 148.05 g/mol - not DA | Possible breakdown product - $C_{10}H_8O_3$, 176.05 g/mol - not DA | 0.98 | 0.98 | - |
| Heat shock protein binding protein - ST4.03ch09, 3.8Mb (PGSC0003DMG400002680) - not DE | TVLP1 - ST4.03ch07, 51.9Mb (PGSC0003DMG400027646) - not DE | 0.96 | 0.96 | Stress response |
| Possible breakdown product - $C_{10}H_8O_3$, 176.05 g/mol - not DA | $C_8H_{15}N_{11}$, 265.15 g/mol - not DA | 0.96 | 0.96 | - |
| Multidrug resistance pump - ST4.03ch08, 2.9Mb (PGSC0003DMG400004474) - not DE | ST4.03ch08, 3,137,398bp | 0.96 | 0.97 | Disease/Pathogen resistance |
| BHLH domain class transcription factor - ST4.03ch03, 58Mb (PGSC0003DMG400014246) - not DE | DNA-directed RNA polymerase II 19 kD polypeptide rpb7 - ST4.03ch03, 58Mb (PGSC0003DMG400014251) - not DE | 0.96 | 0.94 | Stress response |
| 366.65 g/mol - downregulated | N1, N5, N14-(dihydrocaffeoyl)spermine - $C_{37}H_{50}N_4O_9$, 694.36 g/mol - downregulated | 0.92 | 0.93 | Glutathione metabolism |
| N1,N10-Bis(dihydrocaffeoyl)spermidine - $C_{25}H_{35}N_3O_6$, 473.25 g/mol - downregulated | N1, N5, N14-(dihydrocaffeoyl)spermine - $C_{37}H_{50}N_4O_9$, 694.36 g/mol - downregulated | 0.90 | 0.90 | Glutathione metabolism |
| $C_8H_{21}N_{18}PS$, 432.17 g/mol - upregulated | $C_5H_{13}N_5S$, 175.09 g/mol - upregulated | 0.82 | 0.82 | - |

Table G.1: Edges in the consensus skeleton of inferred graphs (with blacklist) with a high average confidence score across the causal and network inference methods. For comparison, the mean confidence score of the edges across the four considered methods (PC-stable, FGES, MMHC and GENIE3) without blacklist is presented. When possible, the possible pathway in which the molecules are involved is indicated. Genomic variants are represented as follows: *chromosome, genomic position*; transcribed genes as: *description - chromosome, genomic position (Ensembl ID) - whether it is differentially expressed or not*; metabolic compounds as: *description if identified - formula if identified, molecular weight - whether it is differentially abundant or not*. Note that most chemical formulas have been automatically generated by the metabolomics analysis software. *(continued)*

| Edge between | And | Mean confidence score - blacklist | Mean confidence score - no blacklist | Possible pathway |
|---|---|---|---|---|
| Conserved gene of unknown function - ST4.03ch08, 49.3Mb (PGSC0003DMG400017523) - upregulated | Conserved gene of unknown function - ST4.03ch07, 3.3Mb (PGSC0003DMG400030726) - upregulated | 0.79 | 0.76 | - |
| Breakdown product of Glutathione - $C_{13}H_6O$, 178.04 g/mol - downregulated | L-Glutathione (reduced) - $C_{10}H_{17}N_3O_6S$, 307.08 g/mol - downregulated | 0.74 | 0.74 | Glutathione metabolism |
| Calcium ion binding protein - ST4.03ch04, 71.8Mb (PGSC0003DMG400009911) - not DE | Prephenate dehydrogenase - ST4.03ch02, 7.7Mb (PGSC0003DMG400042196) - not DE | 0.74 | 0.72 | Tyrosine biosynthesis/signalling |
| Conserved gene of unknown function - ST4.03ch08, 49.3Mb (PGSC0003DMG400017523) - upregulated | Conserved gene of unknown function - ST4.03ch03, 55.1Mb (PGSC0003DMG400024534) - downregulated | 0.71 | 0.67 | - |
| Conserved gene of unknown function - ST4.03ch01, 85.3Mb (PGSC0003DMG400025951) - not DE | Gene of unknown function - ST4.03ch01, 85.4Mb (PGSC0003DMG400025955) - not DE | 0.70 | 0.69 | - |
| pigment from marker pen - 265.16 g/mol - downregulated | N1,N10-Bis(dihydrocaffeoyl)spermidine - $C_{25}H_{35}N_3O_6$, 473.25 g/mol - downregulated | 0.70 | 0.70 | - |

## Appendix H

# Contribution to publications and copyright permission

DRC 16

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

# STATEMENT OF CONTRIBUTION
# DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

| | |
|---|---|
| Name of candidate: | Olivia Angelin-Bonnet |
| Name/title of Primary Supervisor: | Dr Matthieu Vignes |
| Name of Research Output and full reference: | |
| Angelin-Bonnet, O., Biggs, P. J., & Vignes, M. (2019). Gene regulatory networks: a primer in biological processes and statistical modelling. In Gene Regulatory Networks (pp. 347-383). Humana Press, New York, NY. | |
| In which Chapter is the Manuscript /Published work: | 1 |
| Please indicate: | |
| • The percentage of the manuscript/Published Work that was contributed by the candidate: | 80 |
| and | |
| • Describe the contribution that the candidate has made to the Manuscript/Published Work: | |
| The candidate made the necessary research, prepared the manuscript and the figures | |
| For manuscripts intended for publication please indicate target journal: | |
| | |
| Candidate's Signature: | Olivia Angelin-Bonnet Digitally signed by Olivia Angelin-Bonnet<br>Date: 2021.01.29 13:36:21 +13'00' |
| Date: | 29/01/2021 |
| Primary Supervisor's Signature: | Matthieu Vignes Digitally signed by Matthieu Vignes<br>Date: 2021.02.01 13:09:14 +13'00' |
| Date: | 01/02/2021 |

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Oct 13, 2020

---

This Agreement between Massey University -- Olivia Angelin-Bonnet ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

| | |
|---|---|
| License Number | 4926860401672 |
| License date | Oct 13, 2020 |
| Licensed Content Publisher | Springer Nature |
| Licensed Content Publication | Springer eBook |
| Licensed Content Title | Gene Regulatory Networks: A Primer in Biological Processes and Statistical Modelling |
| Licensed Content Author | Olivia Angelin-Bonnet, Patrick J. Biggs, Matthieu Vignes |
| Licensed Content Date | Jan 1, 2019 |
| Type of Use | Thesis/Dissertation |
| Requestor type | academic/university or research institute |
| Format | print and electronic |
| Portion | full article/chapter |
| Will you be translating? | no |

| | |
|---|---|
| Circulation/distribution | 1 - 29 |
| Author of this Springer Nature content | yes |
| Title | Investigation of genotype and phenotype interactions using computational statistics |
| Institution name | Massey University |
| Expected presentation date | Feb 2021 |
| Requestor Location | Massey University<br>Private Bag 11 222<br>Palmerston North, 4442<br>New Zealand<br>Attn: Massey University |
| Total | 0.00 USD |

Terms and Conditions

## Springer Nature Customer Service Centre GmbH
## Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

### 1. Grant of License

**1. 1.** The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

**1. 2.** The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

**1. 3.** If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

## 2. Scope of Licence

**2. 1.** You may only use the Licensed Content in the manner and to the extent permitted by these Ts&Cs and any applicable laws.

**2. 2.** A separate licence may be required for any additional use of the Licensed Material, e.g. where a licence has been purchased for print only use, separate permission must be obtained for electronic re-use. Similarly, a licence is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence. Any content owned by third parties are expressly excluded from the licence.

**2. 3.** Similarly, rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

**2. 4.** Where permission has been granted **free of charge** for material in print, permission may also be granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

**2. 5.** An alternative scope of licence may apply to signatories of the STM Permissions Guidelines, as amended from time to time.

## 3. Duration of Licence

**3. 1.** A licence for is valid from the date of purchase ('Licence Date') at the end of the relevant period in the below table:

| Scope of Licence | Duration of Licence |
|---|---|
| Post on a website | 12 months |
| Presentations | 12 months |
| Books and journals | Lifetime of the edition in the language purchased |

## 4. Acknowledgement

**4. 1.** The Licensor's permission must be acknowledged next to the Licenced Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

## 5. Restrictions on use

**5. 1.** Use of the Licensed Material may be permitted for incidental promotional use and minor editing privileges e.g. minor adaptations of single figures, changes of format, colour and/or style where the adaptation is credited as set out in Appendix 1 below. Any other changes including but not limited to, cropping, adapting, omitting material that affect the meaning, intention or moral rights of the author are strictly prohibited.

**5. 2.** You must not use any Licensed Material as part of any design or trademark.

**5. 3.** Licensed Material may be used in Open Access Publications (OAP) before publication by Springer Nature, but any Licensed Material must be removed from OAP sites prior to final publication.

## 6. Ownership of Rights

**6. 1.** Licensed Material remains the property of either Licensor or the relevant third party and any rights not explicitly granted herein are expressly reserved.

## 7. Warranty

IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND
WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

## 8. Limitations

**8. 1.** ***BOOKS ONLY:***Where **'reuse in a dissertation/thesis'** has been selected the following terms apply: Print rights of the final author's accepted manuscript (for clarity, NOT the published version) for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

## 9. Termination and Cancellation

**9. 1.** Licences will expire after the period shown in Clause 3 (above).

**9. 2.** Licensee reserves the right to terminate the Licence in the event that payment is not received in full or if there has been a breach of this agreement by you.

## Appendix 1 — Acknowledgements:

**For Journal Content:**
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

For **Advance Online Publication papers:**
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

**For Adaptations/Translations:**
Adapted/Translated by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

**Note: For any republication from the British Journal of Cancer, the following credit line style applies:**

Reprinted/adapted/translated by permission from [**the Licensor**]: on behalf of Cancer Research UK: : [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT]** (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from The [**the Licensor**]: on behalf of Cancer Research UK: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

**For Book content:**
Reprinted/adapted by permission from [**the Licensor**]: [**Book Publisher** (e.g. Palgrave Macmillan, Springer etc) [**Book Title**] by [**Book author**(s)] [**COPYRIGHT**] (year of publication)

**Other Conditions**:

Version 1.2

**Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.**

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

# STATEMENT OF CONTRIBUTION
# DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

| | |
|---|---|
| Name of candidate: | Olivia Angelin-Bonnet |
| Name/title of Primary Supervisor: | Dr Matthieu Vignes |
| Name of Research Output and full reference: | |
| Angelin-Bonnet, O., Biggs, P. J., Baldwin, S., Thomson, S., & Vignes, M. (2020). sismonr: simulation of in silico multi-omic networks with adjustable ploidy and post-transcriptional regulation in R. Bioinformatics, 36(9), 2938-2940. | |
| In which Chapter is the Manuscript /Published work: | 2 |
| Please indicate: | |
| • The percentage of the manuscript/Published Work that was contributed by the candidate: | 90 |
| and | |
| • Describe the contribution that the candidate has made to the Manuscript/Published Work: | |
| The candidate did the necessary research, developed and programmed the algorithm, and prepared the manuscript and figures | |
| For manuscripts intended for publication please indicate target journal: | |
| | |
| Candidate's Signature: | Olivia Angelin-Bonnet Digitally signed by Olivia Angelin-Bonnet Date: 2021.01.29 13:35:21 +13'00' |
| Date: | 29/01/2021 |
| Primary Supervisor's Signature: | Matthieu Vignes Digitally signed by Matthieu Vignes Date: 2021.02.01 13:08:07 +13'00' |
| Date: | 01/02/2021 |

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

OXFORD UNIVERSITY PRESS LICENSE
TERMS AND CONDITIONS

Oct 11, 2020

---

This Agreement between Massey University -- Olivia Angelin-Bonnet ("You") and Oxford University Press ("Oxford University Press") consists of your license details and the terms and conditions provided by Oxford University Press and Copyright Clearance Center.

| | |
|---|---|
| License Number | 4926121476445 |
| License date | Oct 11, 2020 |
| Licensed content publisher | Oxford University Press |
| Licensed content publication | Bioinformatics |
| Licensed content title | sismonr: simulation of *in silico* multi-omic networks with adjustable ploidy and post-transcriptional regulation in R |
| Licensed content author | Angelin-Bonnet, Olivia; Biggs, Patrick J |
| Licensed content date | Jan 21, 2020 |
| Type of Use | Thesis/Dissertation |
| Institution name | |
| Title of your work | Investigation of genotype and phenotype interactions using computational statistics |
| Publisher of your work | Massey University |

| | |
|---|---|
| Expected publication date | Feb 2021 |
| Permissions cost | 0.00 USD |
| Value added tax | 0.00 USD |
| Total | 0.00 USD |
| Title | Investigation of genotype and phenotype interactions using computational statistics |
| Institution name | Massey University |
| Expected presentation date | Feb 2021 |
| Portions | All article (abstract, main text and Figure 1) |
| Requestor Location | Massey University<br>Private Bag 11 222<br><br>Palmerston North, 4442<br>New Zealand<br>Attn: Massey University |
| Publisher Tax ID | GB125506730 |
| Total | 0.00 USD |

Terms and Conditions

## STANDARD TERMS AND CONDITIONS FOR REPRODUCTION OF MATERIAL FROM AN OXFORD UNIVERSITY PRESS JOURNAL

1. Use of the material is restricted to the type of use specified in your order details.

2. This permission covers the use of the material in the English language in the following territory: world. If you have requested additional permission to translate this material, the terms and conditions of this reuse will be set out in clause 12.

3. This permission is limited to the particular use authorized in (1) above and does not allow you to sanction its use elsewhere in any other format other than specified above, nor does it apply to quotations, images, artistic works etc that have been reproduced from other sources which may be part of the material to be used.

4. No alteration, omission or addition is made to the material without our written consent. Permission must be re-cleared with Oxford University Press if/when you decide to reprint.

5. The following credit line appears wherever the material is used: author, title, journal, year, volume, issue number, pagination, by permission of Oxford University Press or the sponsoring society if the journal is a society journal. Where a journal is being published on behalf of a learned society, the details of that society must be included in the credit line.

6. For the reproduction of a full article from an Oxford University Press journal for whatever purpose, the corresponding author of the material concerned should be informed of the proposed use. Contact details for the corresponding authors of all Oxford University Press journal contact can be found alongside either the abstract or full text of the article concerned, accessible from www.oxfordjournals.org Should there be a problem clearing these rights, please contact journals.permissions@oup.com

7. If the credit line or acknowledgement in our publication indicates that any of the figures, images or photos was reproduced, drawn or modified from an earlier source it will be necessary for you to clear this permission with the original publisher as well. If this permission has not been obtained, please note that this material cannot be included in your publication/photocopies.

8. While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by Oxford University Press or by Copyright Clearance Center (CCC)) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and Oxford University Press reserves the right to take any and all action to protect its copyright in the materials.

9. This license is personal to you and may not be sublicensed, assigned or transferred by you to any other person without Oxford University Press’s written permission.

10. Oxford University Press reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC’s Billing and Payment terms and conditions.

11. You hereby indemnify and agree to hold harmless Oxford University Press and CCC, and their respective officers, directors, employs and agents, from and against any and all claims

arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

12. Other Terms and Conditions:

v1.4

**Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.**

# References

Acharjee, A., Kloosterman, B., Visser, R. G. F., & Maliepaard, C. (2016). Integration of multi-omics data for prediction of phenotypic traits using random forest. *BMC Bioinformatics*, *17*(5), 180. `http://doi.org/10.1186/s12859-016-1043-4`

Ackers, G. K., Johnson, A. D., & Shea, M. A. (1982). Quantitative model for gene regulation by lambda phage repressor. *Proceedings of the National Academy of Sciences*, *79*(4), 1129–1133. `http://doi.org/10.1073/pnas.79.4.1129`

Ahmed, S. S., Roy, S., & Kalita, J. K. (2018). Assessing the Effectiveness of Causality Inference Methods for Gene Regulatory Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1. `http://doi.org/10.1109/TCBB.2018.2853728`

Ainsworth, H. F., Shin, S. Y., & Cordell, H. J. (2017). A comparison of methods for inferring causal relationships between genotype and phenotype using additional biological measurements. *Genetic Epidemiology*, *41*(7), 577–586. `http://doi.org/10.1002/gepi.22061`

Albert, F. W., & Kruglyak, L. (2015). The role of regulatory variation in complex traits and disease. *Nature Reviews Genetics*, *16*(4), 197–212. `http://doi.org/10.1038/nrg3891`

Albert, N. W., Davies, K. M., Lewis, D. H., Zhang, H., Montefiori, M., Brendolise, C., … Schwinn, K. E. (2014). A Conserved Network of Transcriptional Activators and Repressors Regulates Anthocyanin Pigmentation in Eudicots. *The Plant Cell*, *26*(3), 962–980. `http://doi.org/10.1105/tpc.113.122069`

Albert, R. (2007). Network inference, analysis, and modeling in systems biology. *The Plant Cell*, *19*(11), 3327–3338. `http://doi.org/10.1105/tpc.107.054700`

Albert, R., & Barabási, A. L. (2000). Topology of evolving networks: Local events and universality. *Physical Review Letters*, *85*(24), 5234–5237. `http://doi.org/10.1103/PhysRevLett.85.5234`

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*(1), 47–97. `http://doi.org/10.1103/RevModPhys.74.47`

Alon, U. (2006). *An introduction to systems biology: Design principles of biological circuits*. CRC press.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nat Rev Genet*, *8*(6), 450–461. `http://doi.org/10.1038/nrg2102`

Altay, G., & Mendi, O. (2017). Inferring genome-wide interaction networks. In J. M. Keith (Ed.), *Bioinformat-*

*ics: Volume ii: Structure, function, and applications* (pp. 99–117). New York, NY: Springer New York. http://doi.org/10.1007/978-1-4939-6613-4_6

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*(10), R106. http://doi.org/10.1186/gb-2010-11-10-r106

Andrews, S., Krueger, F., Segonds-Pichon, A., Biggins, L., Krueger, C., & Wingett, S. (2018). FastQC: a quality control tool for high throughput sequence data.

Angelin-Bonnet, O., Biggs, P. J., Biggs, P. J., Baldwin, S., Thomson, S., & Vignes, M. (2020). Sismonr: Simulation of in silico multi-omic networks with adjustable ploidy and post-transcriptional regulation in R. *Bioinformatics*. http://doi.org/10.1093/bioinformatics/btaa002

Angelin-Bonnet, O., Biggs, P. J., & Vignes, M. (2019). Gene Regulatory Networks: A Primer in Biological Processes and Statistical Modelling. In G. Sanguinetti & V. A. Huynh-Thu (Eds.), *Gene regulatory networks: Methods and protocols* (pp. 347–383). Humana Press, New York, NY. http://doi.org/10.1007/978-1-4939-8882-2_15

Aranzana, M. J., Kim, S., Zhao, K., Bakker, E., Horton, M., Jakob, K., … Nordborg, M. (2005). Genome-Wide Association Mapping in Arabidopsis Identifies Previously Known Flowering Time and Pathogen Resistance Genes. *PLoS Genetics*, *1*(5), e60. http://doi.org/10.1371/journal.pgen.0010060

Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J. C., … Stegle, O. (2018). Multi-Omics Factor Analysis-a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, *14*(6), e8124. http://doi.org/10.15252/msb.20178124

Aten, J. E., Fuller, T. F., Lusis, A. J., & Horvath, S. (2008). Using genetic markers to orient the edges in quantitative trait networks: The NEO software. *BMC Systems Biology*, *2*(1), 34. http://doi.org/10.1186/1752-0509-2-34

Auer, P. L., & Doerge, R. W. (2010). Statistical design and analysis of RNA sequencing data. *Genetics*, *185*(2), 405–416. http://doi.org/10.1534/genetics.110.114983

Auerbach, J., Howey, R., Jiang, L., Justice, A., Li, L., Oualkacha, K., … Aslibekyan, S. W. (2018). Causal modeling in a multi-omic setting: insights from GAW20. *BMC Genetics*, *19*(S1), 74. http://doi.org/10.1186/s12863-018-0645-4

Backman, T. W. H., & Girke, T. (2016). systemPipeR: NGS workflow and report generation environment. *BMC Bioinformatics*, *17*(1), 388. http://doi.org/10.1186/s12859-016-1241-0

Balaji, S., Babu, M. M., Iyer, L. M., Luscombe, N. M., & Aravind, L. (2006). Comprehensive Analysis of Combinatorial Regulation using the Transcriptional Regulatory Network of Yeast. *Journal of Molecular Biology*, *360*(1), 213–227. http://doi.org/10.1016/j.jmb.2006.04.029

Baldwin, S. J., Dodds, K. G., Auvray, B., Genet, R. A., Macknight, R. C., & Jacobs, J. M. E. (2011). Association mapping of cold-induced sweetening in potato using historical phenotypic data. *Annals of Applied Biology*, *158*(3), 248–256. http://doi.org/10.1111/j.1744-7348.2011.00459.x

Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512.

http://doi.org/10.1126/science.286.5439.509

Barabási, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, *5*(2), 101–113. http://doi.org/10.1038/nrg1272

Barber, R. F., & Ramdas, A. (2015). The p-filter: multi-layer FDR control for grouped hypotheses. Retrieved from http://arxiv.org/abs/1512.03397

Bartel, J., Krumsiek, J., Schramm, K., Adamski, J., Gieger, C., Herder, C., … Theis, F. J. (2015). The Human Blood Metabolome-Transcriptome Interface. *PLoS Genetics*, *11*(6), 1005274. http://doi.org/10.1371/journal.pgen.1005274

Bazakos, C., Hanemian, M., Trontin, C., Jiménez-Gómez, J. M., & Loudet, O. (2017). New Strategies and Tools in Quantitative Genetics: How to Go from the Phenotype to the Genotype. *Annual Review of Plant Biology*, *68*(1), 435–455. http://doi.org/10.1146/annurev-arplant-042916-040820

Belle, A., Tanay, A., Bitincka, L., Shamir, R., & O'Shea, E. K. (2006). Quantification of protein half-lives in the budding yeast proteome. *Proceedings of the National Academy of Sciences*, *103*(35), 13004–13009. http://doi.org/10.1073/pnas.0605420103

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289–300. http://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Besnier, F., & Glover, K. A. (2013). ParallelStructure: A R Package to Distribute Parallel Runs of the Population Genetics Program STRUCTURE on Multi-Core Computers. *PLoS ONE*, *8*(7), e70651. http://doi.org/10.1371/journal.pone.0070651

Bessière, C., Taha, M., Petitprez, F., Vandel, J., Marin, J. M., Bréhélin, L., … Lecellier, C. H. (2018). Probing instructions for expression regulation in gene nucleotide compositions. *PLoS Computational Biology*, *14*(1). http://doi.org/10.1371/journal.pcbi.1005921

Biggin, M. (2011). Animal Transcription Networks as Highly Connected, Quantitative Continua. *Developmental Cell*, *21*(4), 611–626. http://doi.org/10.1016/J.DEVCEL.2011.09.008

Biggs, P. J., & Collins, L. J. (2011). RNA networks in prokaryotes I: CRISPRs and riboswitches. *Advances in Experimental Medicine and Biology*, *722*, 209–220. http://doi.org/10.1007/978-1-4614-0332-6_13

Bintu, L., Buchler, N. E., Garcia, H. G., Gerland, U., Hwa, T., Kondev, J., … Phillips, R. (2005a). Transcriptional regulation by the numbers: Applications. *Current Opinion in Genetics and Development*, *15*(2), 125–135. http://doi.org/10.1016/j.gde.2005.02.006

Bintu, L., Buchler, N. E., Garcia, H. G., Gerland, U., Hwa, T., Kondev, J., & Phillips, R. (2005b). Transcriptional regulation by the numbers: Models. *Current Opinion in Genetics and Development*, *15*(2), 116–124. http://doi.org/10.1016/j.gde.2005.02.007

Bisognin, D. A., Manrique-Carpintero, N. C., & Douches, D. S. (2018). QTL Analysis of Tuber Dormancy and Sprouting in Potato. *American Journal of Potato Research*, *95*(4), 374–382. http://doi.org/10.

1007/s12230-018-9638-0

Blais, A., & Dynlacht, B. D. (2005). Constructing transcriptional regulatory networks. *Genes and Development*, *19*(13), 1499–1511. http://doi.org/10.1101/gad.1325605

Bland, J. M., & Altman, D. G. (1995). Multiple significance tests: the Bonferroni method. *BMJ*, *310*(6973), 170. http://doi.org/10.1136/BMJ.310.6973.170

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, *30*(15), 2114–2120. http://doi.org/10.1093/bioinformatics/btu170

Bollobás, B., Borgs, C., Chayes, J., & Riordan, O. (2003). Directed scale-free graphs. *Proceedings of the Fourteenth Annual ACMSIAM Symposium on Discrete Algorithms*, 132–139.

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., & Thorsson, V. (2006). The inferelator: An algorithn for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*(5), R36. http://doi.org/10.1186/gb-2006-7-5-r36

Bonnet, E., Calzone, L., & Michoel, T. (2015). Integrative Multi-omics Module Network Inference with Lemon-Tree. *PLOS Computational Biology*, *11*(2), e1003983. http://doi.org/10.1371/journal.pcbi.1003983

Boopathi, N. (2013). QTL Identification. In *Genetic mapping and marker assisted selection* (pp. 117–163). Springer India. http://doi.org/10.1007/978-81-322-0958-4_6

Bornholdt, S. (2008). Boolean network models of cellular regulation: Prospects and limitations. *Journal of the Royal Society Interface*, *5*(SUPPL. 1). http://doi.org/10.1098/rsif.2008.0132.focus

Bourke, P. M., Voorrips, R. E., Visser, R. G. F., & Maliepaard, C. (2015). The double-reduction landscape in tetraploid potato as revealed by a high-density linkage map. *Genetics*, *201*(3), 853–863. http://doi.org/10.1534/genetics.115.181008

Bourke, P. M., Voorrips, R. E., Visser, R. G. F., & Maliepaard, C. (2018). Tools for Genetic Studies in Experimental Populations of Polyploids. *Frontiers in Plant Science*, *9*. http://doi.org/10.3389/fpls.2018.00513

Bradshaw, J. E., Hackett, C. A., Pande, B., Waugh, R., & Bryan, G. J. (2008). QTL mapping of yield, agronomic and quality traits in tetraploid potato (Solanum tuberosum subsp. tuberosum). *Theoretical and Applied Genetics*, *116*(2), 193–211. http://doi.org/10.1007/s00122-007-0659-1

Brzyski, D., Peterson, C. B., Sobczyk, P., Candès, E. J., Bogdan, M., & Sabatti, C. (2017). Controlling the Rate of GWAS False Discoveries. *Genetics*, *205*(1), 61–75. http://doi.org/10.1534/GENETICS.116.193987

Buccitelli, C., & Selbach, M. (2020). mRNAs, proteins and the emerging principles of gene expression control. *Nature Reviews Genetics*, 1–15. http://doi.org/10.1038/s41576-020-0258-4

Bulcke, T. den, Van Leemput, K., Naudts, B., Remortel, P. van, Ma, H., Verschoren, A., … Marchal, K. (2006). SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, *7*(1), 43. http://doi.org/10.1186/1471-2105-7-43

Burgess, S., Small, D. S., & Thompson, S. G. (2017). A review of instrumental variable estimators for Mendelian randomization. *Statistical Methods in Medical Research*, *26*(5), 2333–2355. `http://doi.org/10.1177/0962280215597579`

Bush, W. S., & Moore, J. H. (2012). Chapter 11: Genome-Wide Association Studies. *PLoS Computational Biology*, *8*(12), e1002822. `http://doi.org/10.1371/journal.pcbi.1002822`

Bushnell, B. (2016). BBMap short read aligner, and other bioinformatic tools.

Butte, A. J., & Kohane, I. S. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*, 418–429. `http://doi.org/10.1142/9789814447331_0040`

Bylesjö, M., Rantalainen, M., Cloarec, O., Nicholson, J. K., Holmes, E., & Trygg, J. (2006). OPLS discriminant analysis: Combining the strengths of PLS-DA and SIMCA classification. *Journal of Chemometrics*, *20*(8-10), 341–351. `http://doi.org/10.1002/cem.1006`

Cai, J., & Huo, J. (2020). Sparse generalized canonical correlation analysis via linearized Bregman method. *Communications on Pure and Applied Analysis*, *19*(8), 3933–3945. `http://doi.org/10.3934/cpaa.2020173`

Cantini, L., Zakeri, P., Hernandez, C., Naldi, A., Thieffry, D., Remy, E., & Baudot, A. (2021). Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nature Communications*, *12*(1), 1–12. `http://doi.org/10.1038/s41467-020-20430-7`

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2005). The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics*, *122*(1), 14116. `http://doi.org/10.1063/1.1824902`

Cao, Y., Li, H., & Petzold, L. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The Journal of Chemical Physics*, *121*(9), 4059–4067. `http://doi.org/10.1063/1.1778376`

Cao, Y., & Samuels, D. C. (2009). Discrete stochastic simulation methods for chemically reacting systems. *Methods in Enzymology*, *454*(08), 115–140. `http://doi.org/10.1016/S0076-6879(08)03805-6`

Carpenter, M. A., Joyce, N. I., Genet, R. A., Cooper, R. D., Murray, S. R., Noble, A. D., … Timmerman-Vaughan, G. M. (2015). Starch phosphorylation in potato tubers is influenced by allelic variation in the genes encoding glucan water dikinase, starch branching enzymes I and II, and starch synthase III. *Frontiers in Plant Science*, *6*(MAR), 143. `http://doi.org/10.3389/fpls.2015.00143`

Castel, S. E., & Martienssen, R. A. (2013). RNA interference in the nucleus: Roles for small RNAs in transcription, epigenetics and beyond. *Nature Reviews Genetics*, *14*(2), 100–112. `http://doi.org/10.1038/nrg3355`

Catalanotto, C., Cogoni, C., & Zardo, G. (2016). MicroRNA in control of gene expression: An overview of nuclear functions. *International Journal of Molecular Sciences*, *17*(10). `http://doi.org/10.3390/ijms17101712`

Cavill, R., Jennen, D., Kleinjans, J., & Briedé, J. J. (2016). Transcriptomic and metabolomic data integration.

*Briefings in Bioinformatics*, *17*(5), 891–901. http://doi.org/10.1093/bib/bbv090

Chambers, J. M. (2016). An interface to Julia. In *Extending R*. Boca Raton, FL : CRC Press, Taylor & Francis Group, [2016].

Chandrasekaran, V., Parrilo, P. A., & Willsky, A. S. (2012). Latent variable graphical model selection via convex optimization. *Annals of Statistics*, *40*(4), 1935–1967. http://doi.org/10.1214/11-AOS949

Chen, L., Emmert-Streib, F., & Storey, J. (2007). Harnessing naturally randomized transcription to infer regulatory relationships among genes. *Genome Biology*, *8*(10), R219. http://doi.org/10.1186/gb-2007-8-10-r219

Chen, L., Hu, B., Qin, Y., Hu, G., & Zhao, J. (2019). Advance of the negative regulation of anthocyanin biosynthesis by MYB transcription factors. *Plant Physiology and Biochemistry*, *136*, 178–187. http://doi.org/10.1016/J.PLAPHY.2019.01.024

Chen, Z. J. (2007). Genetic and Epigenetic Mechanisms for Gene Expression and Phenotypic Variation in Plant Polyploids. *Annual Review of Plant Biology*, *58*(1), 377–406. http://doi.org/10.1146/annurev.arplant.58.032806.103835

Chickering, D. M. (2003). Optimal structure identification with greedy search. *Journal of Machine Learning Research*.

Claassen, T., & Heskes, T. (2012). A Logical Characterization of Constraint-Based Causal Discovery. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 135–144. Retrieved from http://arxiv.org/abs/1202.3711

Claassen, T., Mooij, J. M., & Heskes, T. (2013). Learning sparse causal models is not NP-hard. In *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*. Retrieved from http://arxiv.org/abs/1309.6824

Collard, B. C. Y., Jahufer, M. Z. Z., Brouwer, J. B., & Pang, E. C. K. (2005). An introduction to markers, quantitative trait loci (QTL) mapping and marker-assisted selection for crop improvement: The basic concepts. *Euphytica*, *142*(1-2), 169–196. http://doi.org/10.1007/s10681-005-1681-5

Colombo, D., & Maathuis, M. H. (2014). *Order-Independent Constraint-Based Causal Structure Learning* (No. 116) (Vol. 15, pp. 3921–3962).

Colombo, D., Maathuis, M. H., Kalisch, M., & Richardson, T. S. (2012). Learning high-dimensional DAGs with latent and selection variables. *The Annals of Statistics*, *Vol. 40*(No. 1), 294–321. http://doi.org/10.1214/11-AOS940

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., … Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology*, *17*(1). http://doi.org/10.1186/s13059-016-0881-8

Constantinou, A. C. (2019). Evaluating structure learning algorithms with a balanced scoring function. Retrieved from http://arxiv.org/abs/1905.12666

Constantinou, A. C., Liu, Y., Chobtham, K., Guo, Z., & Kitson, N. K. (2020). Large-scale empirical validation of Bayesian Network structure learning algorithms with noisy data. Retrieved from `http://arxiv.org/abs/2005.09020`

Cooper, G. M. (2000). Regulation of Protein Function. In *The cell: A molecular approach. 2nd edition*.

Davey Smith, G., & Ebrahim, S. (2003). 'Mendelian randomization': can genetic epidemiology contribute to understanding environmental determinants of disease? *International Journal of Epidemiology*. `http://doi.org/10.1093/IJE/DYG070`

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, *9*(1), 67–103. `http://doi.org/10.1089/10665270252833208`

De Smet, R., & Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nature Reviews Microbiology*. `http://doi.org/10.1038/nrmicro2419`

D'haeseleer, P., Wen, X., Fuhrman, S., & Somogyi, R. (1999). Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing*, *52*, 41–52. `http://doi.org/10.1142/9789814447300`

D'hoop, B. B., Keizer, P. L. C., Paulo, M. J., Visser, R. G. F., Eeuwijk, F. A. van, & Eck, H. J. van. (2014). Identification of agronomically important QTL in tetraploid potato cultivars using a marker-trait association analysis. *Theoretical and Applied Genetics*, *127*(3), 731–748. `http://doi.org/10.1007/s00122-013-2254-y`

Di Camillo, B., Toffolo, G., & Cobelli, C. (2009). A gene network simulator to assess reverse engineering algorithms. *Annals of the New York Academy of Sciences*, *1158*, 125–142. `http://doi.org/10.1111/j.1749-6632.2008.03756.x`

Dimitromanolakis, A., Xu, J., Krol, A., & Briollais, L. (2019). sim1000G: a user-friendly genetic variant simulator in R for unrelated individuals and family-based designs. *BMC Bioinformatics*, *20*(1), 26. `http://doi.org/10.1186/s12859-019-2611-1`

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., … Gingeras, T. R. (2013). STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, *29*(1), 15–21. `http://doi.org/10.1093/bioinformatics/bts635`

Dona, M. S. I., Prendergast, L. A., Mathivanan, S., Keerthikumar, S., & Salim, A. (2017). Powerful differential expression analysis incorporating network topology for next-generation sequencing data. *Bioinformatics*, *33*(10), 1505–1513. `http://doi.org/10.1093/bioinformatics/btw833`

Donaldson, J. (2016). tsne: T-Distributed Stochastic Neighbor Embedding for R (t-SNE).

Drton, M., & Maathuis, M. H. (2017). Structure Learning in Graphical Modeling. *Annual Review of Statistics and Its Application*, *4*(1), 365–393. `http://doi.org/10.1146/annurev-statistics-060116-053803`

Durinck, S., Spellman, P. T., Birney, E., & Huber, W. (2009). Mapping identifiers for the integration of genomic datasets with the R/ Bioconductor package biomaRt. *Nature Protocols*, *4*(8), 1184–1191. `http:`

//doi.org/10.1038/nprot.2009.97

Eicher, T., Kinnebrew, G., Patt, A., Spencer, K., Ying, K., Ma, Q., … Mathé, E. A. (2020). Metabolomics and Multi-Omics Integration: A Survey of Computational Methods and Resources. *Metabolites*, *10*(5), 202. http://doi.org/10.3390/metabo10050202

El Samad, H., Khammash, M., Petzold, L., & Gillespie, D. (2005). Stochastic modelling of gene regulatory networks. *International Journal of Robust and Nonlinear Control*, *15*(15), 691–711. http://doi.org/10.1002/rnc.1018

Emmert-Streib, F., De Matos Simoes, R., Mullan, P., Haibe-Kains, B., & Dehmer, M. (2014). The gene regulatory network for breast cancer: integrated regulatory landscape of cancer hallmarks. *Frontiers in Genetics*, *5*(FEB), 15. http://doi.org/10.3389/FGENE.2014.00015

Engreitz, J. M., Sirokman, K., McDonel, P., Shishkin, A. A., Surka, C., Russell, P., … Lander, E. S. (2014). RNA-RNA interactions enable specific targeting of noncoding RNAs to nascent pre-mRNAs and chromatin sites. *Cell*, *159*(1), 188–199. http://doi.org/10.1016/j.cell.2014.08.018

Erdös, P., & Rényi, A. (1959). On random graphs. *Publ Math Debrecen*, *6*, 290–297.

Evanno, G., Regnaut, S., & Goudet, J. (2005). Detecting the number of clusters of individuals using the software STRUCTURE: A simulation study. *Molecular Ecology*, *14*(8), 2611–2620. http://doi.org/10.1111/j.1365-294X.2005.02553.x

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., … Gardner, T. S. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, *5*(1), 0054–0066. http://doi.org/10.1371/journal.pbio.0050008

Fatima, N., & Rueda, L. (2020). iSOM-GSN: an integrative approach for transforming multi-omic data into gene similarity networks via self-organizing maps. *Bioinformatics*, *36*(15), 4248–4254. http://doi.org/10.1093/bioinformatics/btaa500

Featherstone, D. E., & Broadie, K. (2002). Wrestling with pleiotropy: Genomic and topological analysis of the yeast gene expression network. *BioEssays*, *24*(3), 267–274. http://doi.org/10.1002/bies.10054

Fellinghauer, B., Bühlmann, P., Ryffel, M., Von Rhein, M., & Reinhardt, J. D. (2013). Stable graphical model estimation with Random Forests for discrete, continuous, and mixed variables. *Computational Statistics and Data Analysis*, *64*, 132–152. http://doi.org/10.1016/j.csda.2013.02.022

Fischer, M., Schreiber, L., Colby, T., Kuckenberg, M., Tacke, E., Hofferbert, H. R., … Gebhardt, C. (2013). Novel candidate genes influencing natural variation in potato tuber cold sweetening identified by comparative proteomics and association mapping. *BMC Plant Biology*, *13*(1), 113. http://doi.org/10.1186/1471-2229-13-113

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, *7*(3-4), 601–620. http://doi.org/10.1089/106652700750050961

Friedman, N., Nachman, I., & Peér, D. (1999). Learning bayesian network structure from massive datasets:

the "sparse candidate" algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*.

Friedman, R. C., Farh, K. K. H., Burge, C. B., & Bartel, D. P. (2009). Most mammalian mRNAs are conserved targets of microRNAs. *Genome Research*, *19*(1), 92–105. http://doi.org/10.1101/gr.082701.108

Frot, B., Nandy, P., & Maathuis, M. H. (2019). Robust causal structure learning with some hidden variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. http://doi.org/10.1111/rssb.12315

Gaffney, D. J. (2013). Global properties and functional complexity of human gene regulatory variation. *PLoS Genet*, *9*(5), e1003501. http://doi.org/10.1371/journal.pgen.1003501

Gao, X., Becker, L. C., Becker, D. M., Starmer, J. D., & Province, M. A. (2010). Avoiding the high Bonferroni penalty in genome-wide association studies. *Genetic Epidemiology*, *34*(1), 100–105. http://doi.org/10.1002/GEPI.20430

Garrison, E., & Marth, G. (2012). Haplotype-based variant detection from short-read sequencing. Retrieved from http://arxiv.org/abs/1207.3907

Gebauer, F., & Hentze, M. W. (2004). Molecular mechanisms of translational control. *Nature Reviews Molecular Cell Biology*, *5*(10), 827–835. http://doi.org/10.1038/nrm1488

Geisler, S., & Coller, J. (2013). RNA in unexpected places: Long non-coding RNA functions in diverse cellular contexts. *Nature Reviews Molecular Cell Biology*, *14*(11), 699–712. http://doi.org/10.1038/nrm3679

Gibson, M. A., & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, *104*(9), 1876–1889. http://doi.org/10.1021/jp993732q

Gilad, Y., Rifkin, S. A., & Pritchard, J. K. (2008). Revealing the architecture of gene regulation: the promise of eQTL studies. *Trends in Genetics*, *24*(8), 408–415. http://doi.org/10.1016/j.tig.2008.06.001

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*(4), 403–434. http://doi.org/10.1016/0021-9991(76)90041-3

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, *81*(25), 2340–2361. http://doi.org/10.1021/j100540a008

Gillespie, D. T. (2000). Chemical Langevin equation. *The Journal of Chemical Physics*, *113*(1), 297–306. http://doi.org/10.1063/1.481811

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, *115*(4), 1716–1733. http://doi.org/10.1063/1.1378322

Gillespie, D. T. (2007). Stochastic Simulation of Chemical Kinetics. *Annual Review of Physical Chemistry*, *58*(1), 35–55. http://doi.org/10.1146/annurev.physchem.58.032806.104637

Gillespie, D. T., & Petzold, L. R. (2003). Improved leap-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics*, *119*(16), 8229–8234. http://doi.org/10.1063/1.1613254

Glymour, C., Zhang, K., & Spirtes, P. (2019). Review of Causal Discovery Methods Based on Graphical Models. *Frontiers in Genetics*, *10*(JUN), 524. http://doi.org/10.3389/fgene.2019.00524

Goeman, J. J., & Solari, A. (2014). Multiple hypothesis testing in genomics. *Statistics in Medicine*, *33*(11), 1946–1978. http://doi.org/10.1002/SIM.6082

Goldberg, A. P., Jefferson, D. R., Sekar, J. A. P., & Karr, J. R. (2020). Exact Parallelization of the Stochastic Simulation Algorithm for Scalable Simulation of Large Biochemical Networks. arXiv. Retrieved from http://arxiv.org/abs/2005.05295

Gonzalez-Zulueta, M., Bender, C. M., Yang, A. S., Nguyen, T. D., Tornout, J. M., Jones, P. A., & Beart, R. W. (1995). Methylation of the 5' CpG Island of the p16/CDKN2 Tumor Suppressor Gene in Normal and Transformed Human Tissues Correlates with Gene Silencing. *Cancer Research*, *55*(20), 4531–4535.

Gross, T., Wongchenko, M. J., Yan, Y., & Blüthgen, N. (2019). Robust network inference using response logic. *Bioinformatics*, *35*(14), i634–i642. http://doi.org/10.1093/bioinformatics/btz326

Guelzim, N., Bottani, S., Bourgine, P., & Képès, F. (2002). Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, *31*(1), 60–63. http://doi.org/10.1038/ng873

Guil, S., & Esteller, M. (2015). RNA-RNA interactions in gene regulation: The coding and noncoding players. *Trends in Biochemical Sciences*, *40*(5), 248–256. http://doi.org/10.1016/j.tibs.2015.03.001

Hache, H., Lehrach, H., & Herwig, R. (2009). Reverse engineering of gene regulatory networks: a comparative study. *EURASIP Journal on Bioinformatics & Systems Biology*, *2009*(1), 617281. http://doi.org/10.1155/2009/617281

Hache, H., Wierling, C., Lehrach, H., & Herwig, R. (2009). GeNGe: Systematic generation of gene regulatory networks. *Bioinformatics*, *25*(9), 1205–1207. http://doi.org/10.1093/bioinformatics/btp115

Halbeisen, R. E., Galgano, A., Scherrer, T., & Gerber, A. P. (2008). Post-transcriptional gene regulation: From genome-wide studies to principles. *Cellular and Molecular Life Sciences*, *65*(5), 798–813. http://doi.org/10.1007/s00018-007-7447-6

Hara-Skrzypiec, A., Śliwka, J., Jakuczun, H., & Zimnoch-Guzowska, E. (2018). Quantitative trait loci for tuber blackspot bruise and enzymatic discoloration susceptibility in diploid potato. *Molecular Genetics and Genomics*, *293*(2), 331–342. http://doi.org/10.1007/s00438-017-1387-0

Hardy, O. J., & Vekemans, X. (2002). Spagedi: A versatile computer program to analyse spatial genetic structure at the individual or population levels. *Molecular Ecology Notes*, *2*(4), 618–620. http://doi.org/10.1046/j.1471-8286.2002.00305.x

Hasin, Y., Seldin, M., & Lusis, A. (2017). Multi-omics approaches to disease. *Genome Biology*, *18*(1), 1–15. http://doi.org/10.1186/s13059-017-1215-1

Hassani, S., Hanafi, M., Qannari, E. M., & Kohler, A. (2013). Deflation strategies for multi-block principal component analysis revisited. *Chemometrics and Intelligent Laboratory Systems*, *120*, 154–168. http:

//doi.org/10.1016/j.chemolab.2012.08.011

Haury, A. C., Mordelet, F., Vera-Licona, P., & Vert, J. P. (2012). TIGRESS: Trustful Inference of Gene REgulation using Stability Selection. *BMC Systems Biology*, *6*(1), 145. `http://doi.org/10.1186/1752-0509-6-145`

Hawe, J. S., Theis, F. J., & Heinig, M. (2019). Inferring interaction networks from multi-omics data. *Frontiers in Genetics*, *10*(JUN), 535. `http://doi.org/10.3389/fgene.2019.00535`

Haynes, B. C., & Brent, M. R. (2009). Benchmarking regulatory network reconstruction with GRENDEL. *Bioinformatics*, *25*(6), 801–807. `http://doi.org/10.1093/bioinformatics/btp068`

Heinze-Deml, C., Maathuis, M. H., & Meinshausen, N. (2018). Causal Structure Learning. *Annual Review of Statistics and Its Application*, *5*(1), 371–391. `http://doi.org/10.1146/annurev-statistics-031017-100630`

Henkin, T. M. (2008). Riboswitch RNAs: Using RNA to sense cellular metabolism. *Genes and Development*, *22*(24), 3383–3390. `http://doi.org/10.1101/gad.1747308`

Herman, J. G., & Baylin, S. B. (2003). Gene silencing in cancer in association with promoter hypermethylation. *The New England Journal of Medicine*, *349*(21), 2042–2054. `http://doi.org/10.1056/NEJMra023075`

Hernández Estévez, I., & Rodríguez Hernández, M. (2020). Plant Glutathione S-transferases: An overview, *23*, 100233. `http://doi.org/10.1016/j.plgene.2020.100233`

Higham, D. J. (2008). Modeling and Simulating Chemical Reactions. *SIAM Rev.*, *50*(2), 347–368. `http://doi.org/10.1137/060666457`

Hill, S. M., Heiser, L. M., Cokelaer, T., Unger, M., Nesser, N. K., Carlin, D. E., … Mukherjee, S. (2016). Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature Methods*, *13*(4), 310–318. `http://doi.org/10.1038/nmeth.3773`

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*. `http://doi.org/10.1080/00401706.1970.10488634`

Holoch, D., & Moazed, D. (2015). RNA-mediated epigenetic regulation of gene expression. *Nature Reviews Genetics*, *16*(2), 71–84. `http://doi.org/10.1038/nrg3863`

Hox, J. J., & Bechger, T. M. (1998). An Introduction to Structural Equation Modeling. *Family Science Review*. `http://doi.org/10.1080/10705510903008345`

Hoyer, P. O., Janzing, D., Mooij, J., Peters, J., & Schölkopf, B. (2008). Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems* (Vol. 21, pp. 689–696).

Höschele, I. (2008). Mapping Quantitative Trait Loci in Outbred Pedigrees. In *Handbook of Statistical Genetics* (pp. 623–677). Chichester, UK: John Wiley & Sons, Ltd. `http://doi.org/10.1002/9780470061619.ch19`

Huber, W., Heydebreck, A. von, Sultmann, H., Poustka, A., & Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, *18*(Suppl 1), S96–S104. http://doi.org/10.1093/bioinformatics/18.suppl_1.S96

Hunter, T. (1995). Protein kinases and phosphatases: The Yin and Yang of protein phosphorylation and signaling. *Cell*, *80*(2), 225–236. http://doi.org/10.1016/0092-8674(95)90405-0

Hutvágner, G., & Zamore, P. D. (2002). A microRNA in a multiple-turnover RNAi enzyme complex. *Science*, *297*(5589), 2056–2060. http://doi.org/10.1126/science.1073827

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., & Geurts, P. (2010). Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLoS ONE*, *5*(9), e12776. http://doi.org/10.1371/journal.pone.0012776

Ignatiadis, N., Klaus, B., Zaugg, J. B., & Huber, W. (2016). Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. *Nature Methods*, *13*(7), 577–580. http://doi.org/10.1038/nmeth.3885

Irizarry, R. A., Warren, D., Spencer, F., Kim, I. F., Biswal, S., Frank, B. C., … Yu, W. (2005). Multiple-laboratory comparison of microarray platforms. *Nature Methods*, *2*(5), 345–349. http://doi.org/10.1038/nmeth756

Jackson, R. J., & Standart, N. (2007). How do microRNAs regulate gene expression? *Science's STKE : Signal Transduction Knowledge Environment*, *2007*(367). http://doi.org/10.1126/stke.3672007re1

Jamil, I. N., Remali, J., Azizan, K. A., Nor Muhammad, N. A., Arita, M., Goh, H. H., & Aizat, W. M. (2020). Systematic Multi-Omics Integration (MOI) Approach in Plant Systems Biology. *Frontiers in Plant Science*, *11*, 944. http://doi.org/10.3389/fpls.2020.00944

Jansen, R. C. (2008). Quantitative Trait Loci in Inbred Lines. In *Handbook of Statistical Genetics* (pp. 587–622). Chichester, UK: John Wiley & Sons, Ltd. http://doi.org/10.1002/9780470061619.ch18

Jansen, R. C., & Nap, J. P. (2001). Genetical genomics: The added value from segregation. *Trends in Genetics*, *17*(7), 388–391. http://doi.org/10.1016/S0168-9525(01)02310-1

Jeong, H., Tombor, B., Albert, R., Oltval, Z. N., & Barabásl, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, *407*(6804), 651–654. http://doi.org/10.1038/35036627

Jombart, T., & Ahmed, I. (2011). adegenet 1.3-1: New tools for the analysis of genome-wide SNP data. *Bioinformatics*, *27*(21), 3070–3071. http://doi.org/10.1093/bioinformatics/btr521

Jombart, T., Devillard, S., & Balloux, F. (2010). Discriminant analysis of principal components: A new method for the analysis of genetically structured populations. *BMC Genetics*, *11*(1), 94. http://doi.org/10.1186/1471-2156-11-94

Jones, P. A., & Baylin, S. B. (2007). The Epigenomics of Cancer. *Cell*, *128*(4), 683–692. http://doi.org/10.1016/J.CELL.2007.01.029

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., & Bühlmann, P. (2012). Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, *47*(1), 1–26. http://doi.org/10.18637/jss.v047.i11

Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., & Eskin, E. (2008). Efficient control of population structure in model organism association mapping. *Genetics*, *178*(3), 1709–1723. `http://doi.org/10.1534/genetics.107.080101`

Karlebach, G., & Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, *9*(10), 770–780. `http://doi.org/10.1038/nrm2503`

Kastritis, P. L., & Bonvin, A. M. J. J. (2012). On the binding affinity of macromolecular interactions: daring to ask why proteins interact. *Journal of the Royal Society Interface*, *10*(79), 20120835–20120835. `http://doi.org/10.1098/rsif.2012.0835`

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*(3), 437–467. `http://doi.org/10.1016/0022-5193(69)90015-0`

Kim, D., Li, R., Dudek, S. M., & Ritchie, M. D. (2015). Predicting censored survival data based on the interactions between meta-dimensional omics data in breast cancer. *Journal of Biomedical Informatics*, *56*, 220–228. `http://doi.org/10.1016/j.jbi.2015.05.019`

Kimuta, H., & Yokota, K. (2004). Characterization of metabolic pathway of linoleic acid 9-hydroperoxide in cytosolic fraction of potato tubers and identification of reaction products. In *Applied Biochemistry and Biotechnology - Part A Enzyme Engineering and Biotechnology* (Vol. 118, pp. 115–132). Springer. `http://doi.org/10.1385/ABAB:118:1-3:115`

Kirk, P., Griffin, J. E., Savage, R. S., Ghahramani, Z., & Wild, D. L. (2012). Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, *28*(24), 3290–3297. `http://doi.org/10.1093/bioinformatics/bts595`

Kloosterman, B., Abelenda, J. A., Gomez, M. D. M. C., Oortwijn, M., De Boer, J. M., Kowitwanich, K., … Bachem, C. W. B. (2013). Naturally occurring allele diversity allows potato cultivation in northern latitudes. *Nature*, *495*(7440), 246–250. `http://doi.org/10.1038/nature11912`

Kong, Y. W., Cannell, I. G., Moor, C. H. de, Hill, K., Garside, P. G., Hamilton, T. L., … Bushell, M. (2008). The mechanism of micro-RNA-mediated translation repression is determined by the promoter of the target gene. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(26), 8866–71. `http://doi.org/10.1073/pnas.0800650105`

Kopylova, E., Noé, L., & Touzet, H. (2012). SortMeRNA: Fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics*, *28*(24), 3211–3217. `http://doi.org/10.1093/bioinformatics/bts611`

Korthauer, K., Chakraborty, S., Benjamini, Y., & Irizarry, R. A. (2019). Detection and accurate false discovery rate control of differentially methylated regions from whole genome bisulfite sequencing. *Biostatistics*, *20*(3), 367–383. `http://doi.org/10.1093/BIOSTATISTICS/KXY007`

Korthauer, K., Kimes, P. K., Duvallet, C., Reyes, A., Subramanian, A., Teng, M., … Hicks, S. C. (2019). A practical guide to methods controlling false discoveries in computational biology. *Genome Biology 2019 20:1*, *20*(1), 1–21. `http://doi.org/10.1186/S13059-019-1716-1`

Kozak, M. (2005). Regulation of translation via mRNA structure in prokaryotes and eukaryotes. *Gene*, *361*(1-2), 13–37. `http://doi.org/10.1016/j.gene.2005.06.037`

Kumar, D., & Chattopadhyay, S. (2018). Glutathione modulates the expression of heat shock proteins via the transcription factors BZIP10 and MYB21 in Arabidopsis. *Journal of Experimental Botany*, *69*(15), 3729–3743. `http://doi.org/10.1093/jxb/ery166`

Kuwano, Y., Kim, H. H., Abdelmohsen, K., Pullmann, R., Martindale, J. L., Yang, X., & Gorospe, M. (2008). MKP-1 mRNA Stabilization and Translational Control by RNA-Binding Proteins HuR and NF90. *Molecular and Cellular Biology*, *28*(14), 4562–4575. `http://doi.org/10.1128/MCB.00165-08`

Lachgar, A., & Achahbar, A. (2016). Network growth with preferential attachment and without "rich get richer" mechanism. *International Journal of Modern Physics C*, *27*(02), 1650020. `http://doi.org/10.1142/S0129183116500200`

Lagnado, D., & Sloman, S. (2002). Learning Causal Structure. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *24*(24), 24.

Lai, D., & Meyer, I. M. (2016). A comprehensive comparison of general RNA-RNA interaction prediction methods. *Nucleic Acids Research*, *44*(7), e61. `http://doi.org/10.1093/nar/gkv1477`

Lamikanra, O., Imam, S., & Ukuku, D. (2005). *Produce degradation: Pathways and prevention*.

Landeros, A., Stutz, T., Keys, K. L., Alekseyenko, A., Sinsheimer, J. S., Lange, K., & Sehl, M. E. (2018). BioSimulator.jl: Stochastic simulation in Julia. *Computer Methods and Programs in Biomedicine*, *167*, 23–35. `http://doi.org/10.1016/J.CMPB.2018.09.009`

Langfelder, P., & Horvath, S. (2008). WGCNA: An R package for weighted correlation network analysis. *BMC Bioinformatics*, *9*(1), 559. `http://doi.org/10.1186/1471-2105-9-559`

Langfelder, P., Zhang, B., & Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: The Dynamic Tree Cut package for R. *Bioinformatics*, *24*(5), 719–720. `http://doi.org/10.1093/bioinformatics/btm563`

Le, T. D., Hoang, T., Li, J., Liu, L., Liu, H., & Hu, S. (2019). A fast PC algorithm for high dimensional causal discovery with multi-core PCs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *16*(5), 1483–1495. `http://doi.org/10.1109/TCBB.2016.2591526`

Lecker, S. H. (2006). Protein Degradation by the Ubiquitin-Proteasome Pathway in Normal and Disease States. *Journal of the American Society of Nephrology*, *17*(7), 1807–1819. `http://doi.org/10.1681/ASN.2006010083`

Lê Cao, K. A., Boitard, S., & Besse, P. (2011). Sparse PLS discriminant analysis: Biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, *12*(1), 253. `http://doi.org/10.1186/1471-2105-12-253`

Lê Cao, K. A., Rossouw, D., Robert-Granié, C., & Besse, P. (2008). A sparse PLS for variable selection when integrating omics data. *Statistical Applications in Genetics and Molecular Biology*, *7*(1). `http://doi.org/10.2202/1544-6115.1390`

Li, B., Carey, M., & Workman, J. L. (2007). The Role of Chromatin during Transcription. *Cell*, *128*(4), 707–719. http://doi.org/10.1016/j.cell.2007.01.015

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Retrieved from http://arxiv.org/abs/1303.3997

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., … Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. http://doi.org/10.1093/bioinformatics/btp352

Li, J., Das, K., Liu, J., Fu, G., Li, Y., Tobias, C., & Wu, R. (2012). Statistical models for genetic mapping in polyploids: Challenges and opportunities. *Methods in Molecular Biology*, *871*, 245–261. http://doi.org/10.1007/978-1-61779-785-9_13

Li, L., Paulo, M. J., Strahwald, J., Lübeck, J., Hofferbert, H. R., Tacke, E., … Gebhardt, C. (2008). Natural DNA variation at candidate loci is associated with potato chip color, tuber starch content, yield and starch yield. *Theoretical and Applied Genetics*, *116*(8), 1167–1181. http://doi.org/10.1007/s00122-008-0746-y

Li, R., Tsaih, S.-W., Shockley, K., Stylianou, I. M., Wergedal, J., Paigen, B., & Churchill, G. A. (2006). Structural Model Analysis of Multiple Quantitative Traits. *PLoS Genetics*, *2*(7), e114. http://doi.org/10.1371/journal.pgen.0020114

Li, Y., Pearl, S. A., & Jackson, S. A. (2015). Gene Networks in Plant Biology: Approaches in Reconstruction and Analysis. *Trends in Plant Science*, *20*(10), 664–675. http://doi.org/10.1016/j.tplants.2015.06.013

Liang, H., & Li, W.-H. (2007). MicroRNA regulation of human protein protein interaction network. *RNA*, *13*(9), 1402–1408. http://doi.org/10.1261/rna.634607

Listgarten, J., Lippert, C., Kadie, C. M., Davidson, R. I., Eskin, E., & Heckerman, D. (2012). Improved linear mixed models for genome-wide association studies. *Nature Methods*, *9*(6), 525–526. http://doi.org/10.1038/nmeth.2037

Liu, B., De La Fuente, A., & Hoeschele, I. (2008). Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, *178*(3), 1763–1776. http://doi.org/10.1534/genetics.107.080069

Liu, Y., Beyer, A., & Aebersold, R. (2016). On the Dependency of Cellular Protein Levels on mRNA Abundance. *Cell*, *165*(3), 535–550. http://doi.org/10.1016/j.cell.2016.03.014

Lizcano, J. M., & Alessi, D. R. (2002). The insulin signalling pathway. *Current Biology*, *12*(7). http://doi.org/10.1016/S0960-9822(02)00777-7

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550. http://doi.org/10.1186/s13059-014-0550-8

Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. *PLoS Computational Biology*, *13*(5). http://doi.org/10.1371/journal.pcbi.1005457

Löfstedt, T., & Trygg, J. (2011). OnPLS-a novel multiblock method for the modelling of predictive and orthogonal variation. *Journal of Chemometrics*, *25*(8), n/a–n/a. http://doi.org/10.1002/cem.1388

Lulai, E. C., Neubauer, J. D., Olson, L. L., & Suttle, J. C. (2015). Wounding induces changes in tuber polyamine content, polyamine metabolic gene expression, and enzyme activity during closing layer formation and initiation of wound periderm formation. *Journal of Plant Physiology*, *176*, 89–95. http://doi.org/10.1016/j.jplph.2014.12.010

Luo, W., Friedman, M. S., Shedden, K., Hankenson, K. D., & Woolf, P. J. (2009). GAGE: Generally applicable gene set enrichment for pathway analysis. *BMC Bioinformatics*, *10*(1), 161. http://doi.org/10.1186/1471-2105-10-161

Mackay, I., & Powell, W. (2007). Methods for linkage disequilibrium mapping in crops. *Trends in Plant Science*, *12*(2), 57–63. http://doi.org/10.1016/j.tplants.2006.12.001

Malosetti, M., Van Der Linden, C. G., Vosman, B., & Van Eeuwijk, F. A. (2007). A mixed-model approach to association mapping using pedigree information with an illustration of resistance to Phytophthora infestans in potato. *Genetics*, *175*(2), 879–889. http://doi.org/10.1534/genetics.105.054932

Mangan, S., & Alon, U. (2003). Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(21), 11980–11985. http://doi.org/10.1073/pnas.2133841100

Marbach, D., Costello, J. C., Küffner, R., Vega, N. N. M., Prill, R. J., Camacho, D. M., … Stolovitzky, G. (2012). Wisdom of crowds for robust gene network inference. *Nat Methods*, *9*(8), 796–804. http://doi.org/10.1038/nmeth.2016

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., & Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, *107*(14), 6286–6291. http://doi.org/10.1073/pnas.0913357107

Marbach, D., Schaffter, T., Mattiussi, C., & Floreano, D. (2009). Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, *16*(2), 1–8. http://doi.org/10.1089/cmb.2008.09TT

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., & Califano, A. (2006). ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*. http://doi.org/10.1186/1471-2105-7-S1-S7

Markowetz, F., & Spang, R. (2007). Inferring cellular networks - a review. *BMC Bioinformatics*, *8*(S6), S5. http://doi.org/10.1186/1471-2105-8-S6-S5

Maston, G. A., Evans, S. K., & Green, M. R. (2006). Transcriptional Regulatory Elements in the Human Genome. *Annual Review of Genomics and Human Genetics*, *7*(1), 29–59. http://doi.org/10.1146/annurev.genom.7.080505.115623

Mattick, J. S., & Makunin, I. V. (2006). Non-coding RNA. *Human Molecular Genetics*, *suppl_1*. http://doi.org/10.1093/hmg/ddl046

McAdams, H. H., & Arkin, A. (1999). It's a noisy business! Genetic regulation at the nanomolar scale. *Trends* *in Genetics*, *15*(2), 65–69. http://doi.org/10.1016/S0168-9525(98)01659-X

McCollum, J. M., Peterson, G. D., Cox, C. D., Simpson, M. L., & Samatova, N. F. (2006). The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, *30*(1), 39–49. http://doi.org/10.1016/j.compbiolchem.2005.10.007

Meek, C. (1997). *Graphical models: selecting causal and statistical models* (PhD thesis). Carnegie Mellon University.

Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, *34*(3), 1436–1462. http://doi.org/10.1214/009053606000000281

Mendes, P., Sha, W., & Ye, K. (2003). Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, *19*(SUPPL. 2). http://doi.org/10.1093/bioinformatics/btg1069

Meng, C., Helm, D., Frejno, M., & Kuster, B. (2016). MoCluster: Identifying Joint Patterns Across Multiple Omics Data Sets. *Journal of Proteome Research*, *15*(3), 755–765. http://doi.org/10.1021/acs.jproteome.5b00824

Meng, C., Kuster, B., Culhane, A. C., & Gholami, A. M. (2014). A multivariate approach to the integration of multi-omics datasets. *BMC Bioinformatics*, *15*(1), 162. http://doi.org/10.1186/1471-2105-15-162

Meng, C., Zeleznik, O. A., Thallinger, G. G., Kuster, B., Gholami, A. M., & Culhane, A. C. (2016). Dimension reduction techniques for the integrative analysis of multi-omics data. *Briefings in Bioinformatics*, *17*(4), 628–641. http://doi.org/10.1093/bib/bbv108

Mercer, T. R., Dinger, M. E., & Mattick, J. S. (2009). Long non-coding RNAs: Insights into functions. *Nature Reviews Genetics*, *10*(3), 155–159. http://doi.org/10.1038/nrg2521

Merchante, C., Stepanova, A. N., & Alonso, J. M. (2017). Translation regulation in plants: an interesting past, an exciting present and a promising future. *Plant Journal*, *90*(4), 628–653. http://doi.org/10.1111/tpj.13520

Meyer, P. E., Kontos, K., Lafitte, F., & Bontempi, G. (2007). Information-theoretic inference of large transcriptional regulatory networks. *Eurasip Journal on Bioinformatics and Systems Biology*, *2007*(1), 1–9. http://doi.org/10.1155/2007/79879

Meyer, P. E., Lafitte, F., & Bontempi, G. (2008). Minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, *9*(1), 461. http://doi.org/10.1186/1471-2105-9-461

Michaelson, J. J., Loguercio, S., & Beyer, A. (2009). Detection and interpretation of expression quantitative trait loci (eQTL). *Methods*. http://doi.org/10.1016/j.ymeth.2009.03.004

Miles, A., Bot, R., M., Ralph, P., Harding, N., Pisupati, R., … Millar, T. (2020). Cggh/scikit-allel: V1.3.2. http://doi.org/10.5281/ZENODO.3976233

Millstein, J., Zhang, B., Zhu, J., & Schadt, E. E. (2009). Disentangling molecular relationships with a causal inference test. *BMC Genetics*, *10*(1), 23. http://doi.org/10.1186/1471-2156-10-23

Milo, R., Jorgensen, P., Moran, U., Weber, G., & Springer, M. (2009). BioNumbers The database of key numbers in molecular and cell biology. *Nucleic Acids Research*, *38*(SUPPL.1). http://doi.org/10.1093/nar/gkp889

Milo, R., & Phillips, R. (2016). *Cell biology by the numbers* (p. 356). New York, NY : Garland Science, Taylor & Francis Group, LLC, [2016].

Milo, R., Shen-Orr, S. S., Itzkovitz, S., Kashtan, N., Chklovskii, D., & Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, *298*(5594), 824–827. http://doi.org/10.1126/science.298.5594.824

Misra, B. B., Langefeld, C., Olivier, M., & Cox, L. A. (2019). Integrated omics: Tools, advances and future approaches. *Journal of Molecular Endocrinology*, *62*(1), R21–R45. http://doi.org/10.1530/JME-18-0055

Mock, A., Warta, R., Dettling, S., Brors, B., Jäger, D., & Herold-Mende, C. (2018). MetaboDiff: an R package for differential metabolomic analysis. *Bioinformatics*, *34*(19), 3417–3418. http://doi.org/10.1093/bioinformatics/bty344

Montastier, E., Villa-Vialaneix, N., Caspar-Bauguil, S., Hlavaty, P., Tvrzicka, E., Gonzalez, I., … Viguerie, N. (2015). System Model Network for Adipose Tissue Signatures Related to Weight Changes in Response to Calorie Restriction and Subsequent Weight Maintenance. *PLOS Computational Biology*, *11*(1), e1004047. http://doi.org/10.1371/journal.pcbi.1004047

Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., & Schölkopf, B. (2016). Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks. *Journal of Machine Learning Research*, *17*(32), 1–102.

Morris, K. V., & Mattick, J. S. (2014). The rise of regulatory RNA. *Nature Reviews Genetics*, *15*(6), 423–437. http://doi.org/10.1038/nrg3722

Motazedi, E., Ridder, D. de, Finkers, R., Baldwin, S., Thomson, S., Monaghan, K., & Maliepaard, C. (2018). TriPoly: haplotype estimation for polyploids using sequencing data of related individuals. *Bioinformatics*, *34*(22), 3864–3872. http://doi.org/10.1093/bioinformatics/bty442

Muldoon, J. J., Yu, J. S., Fassia, M.-K., & Bagheri, N. (2019). Network inference performance complexity: a consequence of topological, experimental and algorithmic determinants. *Bioinformatics*. http://doi.org/10.1093/bioinformatics/btz105

Nagarajan, R., Scutari, M., & Lèbre, S. (2013). *Bayesian Networks in R: with Applications in Systems Biology* (pp. 1–157). Springer New York. http://doi.org/10.1007/978-1-4614-6446-4

Nalefski, E. A., Nebelitsky, E., Lloyd, J. A., & Gullans, S. R. (2006). Single-molecule detection of transcription factor binding to DNA in real time: Specificity, equilibrium, and kinetic parameters. *Biochemistry*. http://doi.org/10.1021/bi0602011

Nandy, P., Hauser, A., & Maathuis, M. H. (2018). High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, *46*(6A), 3151–3183. `http://doi.org/10.1214/17-AOS1654`

Neal, M. L., Wei, L., Peterson, E., Arrieta-Ortiz, M. L., Danziger, S. A., Baliga, N. S., … Aitchison, J. D. (2021). A systems-level gene regulatory network model for Plasmodium falciparum. *Nucleic Acids Research*, *49*(9), 4891–4906. `http://doi.org/10.1093/NAR/GKAA1245`

Neto, E. C., Ferrara, C. T., Attie, A. D., & Yandell, B. S. (2008). Inferring causal phenotype networks from segregating populations. *Genetics*, *179*(2), 1089–1100. `http://doi.org/10.1534/genetics.107.085167`

Nilsson, R., Peña, J. M., Björkegren, J., & Tegnér, J. (2007). Consistent Feature Selection for Pattern Recognition in Polynomial Time. *Journal of Machine Learning Research*, *8*, 589–612. `http://doi.org/10.5555/1314498.1314519`

Noble, W. S. (2009). How does multiple testing correction work? *Nature Biotechnology 2009 27:12*, *27*(12), 1135–1137. `http://doi.org/10.1038/nbt1209-1135`

Ogarrio, J. M., Spirtes, P., & Ramsey, J. (2016). A Hybrid Causal Search Algorithm for Latent Variable Models. In A. Antonucci, G. Corani, & C. Polpo Campos (Eds.), *Proceedings of the Eighth International Conference on Probabilistic Graphical Models* (pp. 368–379). PMLR.

Oliehoek, P. A., Windig, J. J., Van Arendonk, J. A. M., & Bijma, P. (2006). Estimating relatedness between individuals in general populations with a focus on their use in conservation programs. *Genetics*, *173*(1), 483–496. `http://doi.org/10.1534/genetics.105.049940`

Olson, T. S., & Dice, J. F. (1989). Regulation of protein degradation rates in eukaryotes. *Current Opinion in Cell Biology*, *1*(6), 1194–1200. `http://doi.org/10.1016/S0955-0674(89)80071-7`

Osborn, T. C., Chris Pires, J., Birchler, J. A., Auger, D. L., Jeffery Chen, Z., Lee, H.-S., … Martienssen, R. A. (2003). Understanding mechanisms of novel gene expression in polyploids. *Trends in Genetics*, *19*(3), 141–147. `http://doi.org/10.1016/S0168-9525(03)00015-5`

Ouma, W. Z., Pogacar, K., & Grotewold, E. (2018). Topological and statistical analyses of gene regulatory networks reveal unifying yet quantitatively different emergent properties. *PLOS Computational Biology*, *14*(4), e1006098. `http://doi.org/10.1371/JOURNAL.PCBI.1006098`

Pahle, J. (2009). Biochemical simulations: Stochastic, approximate stochastic and hybrid approaches. *Briefings in Bioinformatics*, *10*(1), 53–64. `http://doi.org/10.1093/bib/bbn050`

Pai, A. A., Pritchard, J. K., & Gilad, Y. (2015). The Genetic and Mechanistic Basis for Variation in Gene Regulation. *PLoS Genetics*, *11*(1). `http://doi.org/10.1371/journal.pgen.1004857`

Parisod, C., Holderegger, R., & Brochmann, C. (2010). Evolutionary consequences of autopolyploidy. *New Phytologist*, *186*(1), 5–17. `http://doi.org/10.1111/j.1469-8137.2009.03142.x`

Pelechano, V., Chávez, S., & Pérez-Ortín, J. E. (2010). A complete set of nascent transcription rates for yeast genes. *PLoS ONE*. `http://doi.org/10.1371/journal.pone.0015442`

Peng, Z., He, S., Gong, W., Xu, F., Pan, Z., Jia, Y., … Du, X. (2018). Integration of proteomic and transcriptomic profiles reveals multiple levels of genetic regulation of salt tolerance in cotton. *BMC Plant*

*Biology*, *18*(1), 128. http://doi.org/10.1186/s12870-018-1350-1

Peña, J. M., Björkegren, J., & Tegnér, J. (2005). Scalable, efficient and correct learning of markov boundaries under the faithfulness assumption. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 3571 LNAI, pp. 136–147). Springer Verlag. http://doi.org/10.1007/11518655_13

Peñagaricano, F., Valente, B. D., Steibel, J. P., Bates, R. O., Ernst, C. W., Khatib, H., & Rosa, G. J. (2015a). Exploring causal networks underlying fat deposition and muscularity in pigs through the integration of phenotypic, genotypic and transcriptomic data. *BMC Systems Biology*, *9*(1), 58. http://doi.org/10.1186/s12918-015-0207-6

Peñagaricano, F., Valente, B. D., Steibel, J. P., Bates, R. O., Ernst, C. W., Khatib, H., & Rosa, G. J. (2015b). Searching for causal networks involving latent variables in complex traits: Application to growth, carcass, and meat quality traits in pigs. *Journal of Animal Science*, *93*(10), 4617–4623. http://doi.org/10.2527/jas.2015-9213

Picard toolkit. (2019). *Broad Institute, GitHub repository*. http://broadinstitute.github.io/picard/; Broad Institute.

Pinna, A., Soranzo, N., Hoeschele, I., & Fuente, A. de la. (2011). Simulating systems genetics data with SysGenSIM. *Bioinformatics*, *27*(17), 2459–2462. http://doi.org/10.1093/bioinformatics/btr407

Ponting, C. P., Oliver, P. L., & Reik, W. (2009). Evolution and Functions of Long Noncoding RNAs. *Cell*, *136*(4), 629–641. http://doi.org/10.1016/j.cell.2009.02.006

Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics*.

Pržulj, N., Corneil, D. G., & Jurisica, I. (2004). Modeling interactome: Scale-free or geometric? *Bioinformatics*, *20*(18), 3508–3515. http://doi.org/10.1093/bioinformatics/bth436

Qiu, C., Yu, F., Su, K., Zhao, Q., Zhang, L., Xu, C., … Shen, H. (2020). Multi-omics Data Integration for Identifying Osteoporosis Biomarkers and Their Biological Interaction and Causal Mechanisms. *iScience*, *23*(2), 100847. http://doi.org/10.1016/j.isci.2020.100847

Quinlan, A. R., & Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, *26*(6), 841–842. http://doi.org/10.1093/bioinformatics/btq033

Quinn, J. J., & Chang, H. Y. (2016). Unique features of long non-coding RNA biogenesis and function. *Nature Reviews Genetics*, *17*(1), 47–62. http://doi.org/10.1038/nrg.2015.10

Ramsey, J., Glymour, M., Sanchez-Romero, R., & Glymour, C. (2017). A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, *3*(2), 121–129. http://doi.org/10.1007/s41060-016-0032-z

Ramsey, J., Spirtes, P., & Zhang, J. (2006). Adjacency-faithfulness and conservative causal inference. *Proc. 22th Conf. On Uncertainty in Artificial Intelligence (UAI2006)*, 401–408.

Rapoport, T. A. (2007). Protein translocation across the eukaryotic endoplasmic reticulum and bacterial plasma membranes. *Nature*, *450*(7170), 663–669. http://doi.org/10.1038/nature06384

Ravasi, T., Suzuki, H., Cannistraci, C. V., Katayama, S., Bajic, V. B., Tan, K., … Hayashizaki, Y. (2010). An Atlas of Combinatorial Transcriptional Regulation in Mouse and Man. *Cell*, *140*(5), 744–752. http://doi.org/10.1016/j.cell.2010.01.044

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., & Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, *297*(5586), 1551–1555. http://doi.org/10.1126/science.1073374

Renny-Byfield, S., & Wendel, J. F. (2014). Doubling down on genomes: Polyploidy and crop plants. *American Journal of Botany*, *101*(10), 1711–1725. http://doi.org/10.3732/ajb.1400119

Ribeiro, A. S., & Lloyd-Price, J. (2007). SGN Sim, a stochastic genetic networks simulator. *Bioinformatics*, *23*(6), 777–779. http://doi.org/10.1093/bioinformatics/btm004

Rigaill, G., Balzergue, S., Brunaud, V., Blondet, E., Rau, A., Rogier, O., … Delannoy, E. (2016). Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis. *Briefings in Bioinformatics*, *19*(1), 65–76. http://doi.org/10.1093/bib/bbw092

Rijsbergen, C. J. V. (1979). *Information retrieval* (2nd ed.). Butterworth-Heinemann.

Rinn, J. L., & Chang, H. Y. (2012). Genome Regulation by Long Noncoding RNAs. *Annual Review of Biochemistry*, *81*(1), 145–166. http://doi.org/10.1146/annurev-biochem-051410-092902

Rocke, D. M., & Durbin, B. (2001). A Model for Measurement Error for Gene Expression Arrays. *Journal of Computational Biology*, *8*(6), 557–569. http://doi.org/10.1089/106652701753307485

Rockman, M. V. (2008). Reverse engineering the genotype-phenotype map with natural genetic variation. *Nature*, *456*(7223), 738–744. http://doi.org/10.1038/nature07633

Rohart, F., Gautier, B., Singh, A., & Lê Cao, K.-A. (2017). mixOmics: An R package for 'omics feature selection and multiple data integration. *PLOS Computational Biology*, *13*(11), e1005752. http://doi.org/10.1371/journal.pcbi.1005752

Rosenfeld, N., Elowitz, M. B., & Alon, U. (2002). Negative autoregulation speeds the response times of transcription networks. *Journal of Molecular Biology*, *323*(5), 785–793. http://doi.org/10.1016/S0022-2836(02)00994-4

Ross, I. L., Browne, C. M., & Hume, D. A. (1994). Transcription of individual genes in eukaryotic cells occurs randomly and infrequently. *Immunology and Cell Biology*, *72*(2), 177–185. http://doi.org/10.1111/j.1440-1711.1994.tb03774.x

Rosyara, U. R., De Jong, W. S., Douches, D. S., & Endelman, J. B. (2016). Software for Genome-Wide Association Studies in Autopolyploids and Its Application to Potato. *The Plant Genome*, *9*(2), 0. http://doi.org/10.3835/plantgenome2015.08.0073

Roy, S., Werner-Washburne, M., & Lane, T. (2008). A system for generating transcription regulatory networks with combinatorial control of transcription. *Bioinformatics*, *24*(10), 1318–1320. http://doi.org/10.

1093/bioinformatics/btn126

Salari, R., Backofen, R., & Sahinalp, S. C. (2010). Fast prediction of RNA-RNA interaction. *Algorithms for Molecular Biology*, *5*(1), 5. http://doi.org/10.1186/1748-7188-5-5

Sanguinetti, G., Noirel, J., & Wright, P. C. (2008). MMG: A probabilistic tool to identify submodules of metabolic pathways. *Bioinformatics*, *24*(8), 1078–1084. http://doi.org/10.1093/bioinformatics/btn066

Schadt, E. E., Lamb, J., Yang, X., Zhu, J., Edwards, S., GuhaThakurta, D., … Lusis, A. J. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, *37*(7), 710–717. http://doi.org/10.1038/ng1589

Schaffter, T., Marbach, D., & Floreano, D. (2011). GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, *27*(16), 2263–2270. http://doi.org/10.1093/bioinformatics/btr373

Schilstra, M. J., & Nehaniv, C. L. (2008). Bio-Logic: Gene Expression and the Laws of Combinatorial Logic. *Artificial Life*, *14*(1), 121–133. http://doi.org/10.1162/artl.2008.14.1.121

Schönhals, E. M., Ding, J., Ritter, E., Paulo, M. J., Cara, N., Tacke, E., … Gebhardt, C. (2017). Physical mapping of QTL for tuber yield, starch content and starch yield in tetraploid potato (Solanum tuberosum L.) by means of genome wide genotyping by sequencing and the 8.3 K SolCAP SNP array. *BMC Genomics*, *18*(1), 642. http://doi.org/10.1186/s12864-017-3979-9

Schönhals, E. M., Ortega, F., Barandalla, L., Aragones, A., Ruiz de Galarreta, J. I., Liao, J. C., … Gebhardt, C. (2016). Identification and reproducibility of diagnostic DNA markers for tuber starch and yield optimization in a novel association mapping population of potato (Solanum tuberosum L.). *Theoretical and Applied Genetics*, *129*(4), 767–785. http://doi.org/10.1007/s00122-016-2665-7

Schreiber, G., Haran, G., & Zhou, H.-X. (2009). Fundamental aspects of protein- protein association kinetics. *Chemical Reviews*, *109*(3), 839–860. http://doi.org/10.1021/cr800373w

Schreiber, L., Nader-Nieto, A. C., Schönhals, E. M., Walkemeier, B., & Gebhardt, C. (2014). SNPs in genes functional in starch-sugar interconversion associate with natural variation of tuber starch and sugar content of potato (Solanum tuberosum L.). *G3: Genes, Genomes, Genetics*, *4*(10), 1797–1811. http://doi.org/10.1534/g3.114.012377

Schumacker, R. E., & Lomax, R. G. (2016). *A Beginner's Guide to Strutural Equation Modeling*. http://doi.org/10.1198/tech.2005.s328

Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., … Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, *473*(7347), 337–342. http://doi.org/10.1038/nature10098

Scutari, M., Graafland, C. E., & Gutiérrez, J. M. (2019). Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, *115*, 235–253. http://doi.org/10.1016/j.ijar.2019.10.003

Sedgewick, A. J., Buschur, K., Shi, I., Ramsey, J. D., Raghu, V. K., Manatakis, D. V., … Benos, P. V. (2019). Mixed graphical models for integrative causal analysis with application to chronic lung disease diagnosis and prognosis. *Bioinformatics*, *35*(7), 1204–1212. `http://doi.org/10.1093/bioinformatics/bty769`

Sehl, M., Alekseyenko, A. V., & Lange, K. L. (2009). Accurate stochastic simulation via the step anticipation $\tau$-leaping (SAL) algorithm. *Journal of Computational Biology*, *16*(9), 1195–1208. `http://doi.org/10.1089/cmb.2008.0249`

Serganov, A., & Patel, D. J. (2012). Metabolite Recognition Principles and Molecular Mechanisms Underlying Riboswitch Function. *Annual Review of Biophysics*, *41*(1), 343–370. `http://doi.org/10.1146/annurev-biophys-101211-113224`

Sharma, S. K., Bolser, D., Boer, J. de, Sønderkær, M., Amoros, W., Carboni, M. F., … Bryan, G. J. (2013). Construction of reference chromosome-scale pseudomolecules for potato: Integrating the potato genome with genetic and physical maps. *G3: Genes, Genomes, Genetics*, *3*(11), 2031–2047. `http://doi.org/10.1534/g3.113.007153`

Sharma, S. K., MacKenzie, K., McLean, K., Dale, F., Daniels, S., & Bryan, G. J. (2018). Linkage disequilibrium and evaluation of genome-wide association mapping models in tetraploid potato. *G3: Genes, Genomes, Genetics*, *8*(10), 3185–3202. `http://doi.org/10.1534/g3.118.200377`

Shen, J., Li, Z., Chen, J., Song, Z., Zhou, Z., & Shi, Y. (2016). SHEsisPlus, a toolset for genetic studies on polyploid species. *Scientific Reports*, *6*(1), 1–10. `http://doi.org/10.1038/srep24095`

Shen, R., Olshen, A. B., & Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, *25*(22), 2906–2912. `http://doi.org/10.1093/bioinformatics/btp543`

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, *31*(1), 64–68. `http://doi.org/10.1038/ng881`

Shi, W. J., Zhuang, Y., Russell, P. H., Hobbs, B. D., Parker, M. M., Castaldi, P. J., … Kechris, K. (2019). Unsupervised discovery of phenotype-specific multi-omics networks. *Bioinformatics*, *35*(21), 4336–4343. `http://doi.org/10.1093/bioinformatics/btz226`

Shimizu, S., Hoyer, P. O., Hyvärinen, A., & Kerminen, A. (2006). *A Linear Non-Gaussian Acyclic Model for Causal Discovery Antti Kerminen* (No. 72) (Vol. 7, pp. 2003–2030).

Shin, S. Y., Fauman, E. B., Petersen, A. K., Krumsiek, J., Santos, R., Huang, J., … Soranzo, N. (2014). An atlas of genetic influences on human blood metabolites. *Nature Genetics*, *46*(6), 543–550. `http://doi.org/10.1038/ng.2982`

Shpitser, I., & Pearl, J. (2006). Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proceedings of the National Conference on Artificial Intelligence*.

Siegmund, D. O., Zhang, N. R., & Yakir, B. (2011). False discovery rate for scanning statistics. *Biometrika*, *98*(4), 979–985. `http://doi.org/10.1093/BIOMET/ASR057`

Sima, C., Hua, J., & Jung, S. (2009). Inference of Gene Regulatory Networks Using Time-Series Data: A

Survey. *Current Genomics*, *10*(6), 416–429. http://doi.org/10.2174/138920209789177610

Simoes, R. de M., Dehmer, M., & Emmert-Streib, F. (2013). B-cell lymphoma gene regulatory networks: Biological consistency among inference methods. *Frontiers in Genetics*, *4*, 281. http://doi.org/10.3389/fgene.2013.00281

Singh, A., Shannon, C., Gautier, B., Rohart, F., Vacher, M., Tebbutt, S., & Lê Cao, K.-A. (2016). DIABLO: from multi-omics assays to biomarker discovery, an integrative approach. *bioRxiv*, 067611. http://doi.org/10.1101/067611

Siwiak, M., & Zielenkiewicz, P. (2010). A Comprehensive, Quantitative, and Genome-Wide Model of Translation. *PLoS Computational Biology*, *6*(7), e1000865. http://doi.org/10.1371/journal.pcbi.1000865

Sonenberg, N., & Hinnebusch, A. G. (2009). Regulation of Translation Initiation in Eukaryotes: Mechanisms and Biological Targets. *Cell*, *136*(4), 731–745. http://doi.org/10.1016/j.cell.2009.01.042

Spirtes, P., & Glymour, C. (1991). An Algorithm for Fast Recovery of Sparse Causal Graphs. *Social Science Computer Review*. http://doi.org/10.1177/089443939100900106

Spirtes, P., & Glymour, C. N. (1990). Causal structure among measured variables preserved with unmeasured variables. http://doi.org/10.1184/R1/6491087.V1

Spirtes, P., Glymour, C. N., & Scheines, R. (2001). *Causation, Prediction, and Search* (2nd edn, p. 543). MIT Press.

Spirtes, P., Meek, C., & Richardson, T. (1999). An Algorithm for Causal Inference in the Presence of Latent Variables and Selection Bias. In G. F. Cooper & C. Glymour (Eds.), *Computation, Causation, and Discovery* (pp. 211–252). Cambridge, MA: The MIT Press. http://doi.org/10.7551/mitpress/2006.003.0009

Srivastava, V., Obudulu, O., Bygdell, J., Löfstedt, T., Rydén, P., Nilsson, R., … Wingsle, G. (2013). OnPLS integration of transcriptomic, proteomic and metabolomic data shows multi-level oxidative stress responses in the cambium of transgenic hipI- superoxide dismutase Populus plants. *BMC Genomics*, *14*(1), 1–16. http://doi.org/10.1186/1471-2164-14-893

Stolovitzky, G., Monroe, D., & Califano, A. (2007). Dialogue on reverse-engineering assessment and methods: The DREAM of high-throughput pathway inference. In *Annals of the New York Academy of Sciences* (Vol. 1115, pp. 1–22). http://doi.org/10.1196/annals.1407.021

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(3), 479–498. http://doi.org/10.1111/1467-9868.00346

Storz, G., Altuvia, S., & Wassarman, K. M. (2005). An abundance of RNA regulators. *Annual Review of Biochemistry*, *74*, 199–217. http://doi.org/10.1146/annurev.biochem.74.082803.133136

Subramanian, I., Verma, S., Kumar, S., Jere, A., & Anamika, K. (2020). Multi-omics Data Integration, Interpretation, and Its Application. *Bioinformatics and Biology Insights*, *14*, 117793221989905. http://doi.org/10.1177/1177932219899051

Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., … Von Mering, C. (2017). The STRING database in 2017: Quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Research*, *45*(D1), D362–D368. http://doi.org/10.1093/nar/gkw937

Tam, V., Patel, N., Turcotte, M., Bossé, Y., Paré, G., & Meyre, D. (2019). Benefits and limitations of genome-wide association studies. *Nature Reviews Genetics*, 1. http://doi.org/10.1038/s41576-019-0127-1

Tayrac, M. de, Lê, S., Aubry, M., Mosser, J., & Husson, F. (2009). Simultaneous analysis of distinct Omics data sets with integration of biological knowledge: Multiple Factor Analysis approach. *BMC Genomics*, *10*(1), 32. http://doi.org/10.1186/1471-2164-10-32

Tenenhaus, M., Tenenhaus, A., & Groenen, P. J. F. (2017). Regularized Generalized Canonical Correlation Analysis: A Framework for Sequential Multiblock Component Methods. *Psychometrika*, *82*(3), 737–777. http://doi.org/10.1007/s11336-017-9573-x

Thomas, D. C., & Conti, D. V. (2004). Commentary: The concept of 'Mendelian randomization'. *International Journal of Epidemiology*, *33*(1), 21–25. http://doi.org/10.1093/ije/dyh048

Tibshirani, R. (1996). Regression Selection and Shrinkage via the Lasso. *Journal of the Royal Statistical Society B*. http://doi.org/10.2307/2346178

Tripathi, S., Lloyd-Price, J., Ribeiro, A., Yli-Harja, O., Dehmer, M., & Emmert-Streib, F. (2017). sgnesR: An R package for simulating gene expression data from an underlying real gene network structure considering delay parameters. *BMC Bioinformatics*, *18*(1), 325. http://doi.org/10.1186/s12859-017-1731-8

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003a). Algorithms for Large Scale Markov Blanket Discovery. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*.

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003b). Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 673–678). New York, New York, USA: ACM Press. http://doi.org/10.1145/956750.956838

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78. http://doi.org/10.1007/s10994-006-6889-7

Tsirlis, K., Lagani, V., Triantafillou, S., & Tsamardinos, I. (2018). On scoring Maximal Ancestral Graphs with the Max–Min Hill Climbing algorithm. *International Journal of Approximate Reasoning*, *102*, 74–85. http://doi.org/10.1016/J.IJAR.2018.08.002

Tu, K., Yu, H., Hua, Y. J., Li, Y. Y., Liu, L., Xie, L., & Li, Y. X. (2009). Combinatorial network of primary and secondary microRNA-driven regulatory mechanisms. *Nucleic Acids Research*, *37*(18), 5969–5980. http://doi.org/10.1093/nar/gkp638

Turner, T. E., Schnell, S., & Burrage, K. (2004). Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, *28*(3), 165–178. http://doi.org/10.1016/j.compbiolchem.2004.05.001

Urbany, C., Colby, T., Stich, B., Schmidt, L., Schmidt, J., & Gebhardt, C. (2012). Analysis of Natural Variation of the Potato Tuber Proteome Reveals Novel Candidate Genes for Tuber Bruising. *Journal of Proteome Research*, *11*(2), 703–716. http://doi.org/10.1021/pr2006186

Urbany, C., Stich, B., Schmidt, L., Simon, L., Berding, H., Junghans, H., … Gebhardt, C. (2011). Association genetics in Solanum tuberosum provides new insights into potato tuber bruising and enzymatic tissue discoloration. *BMC Genomics*, *12*(1), 7. http://doi.org/10.1186/1471-2164-12-7

Valencia-Sanchez, M. A., Liu, J., Hannon, G. J., & Parker, R. (2006). Control of translation and mRNA degradation by miRNAs and siRNAs. *Genes and Development*, *20*(5), 515–524. http://doi.org/10.1101/gad.1399806

Van den Broeck, L., Gordon, M., Inzé, D., Williams, C., & Sozzani, R. (2020). Gene Regulatory Network Inference: Connecting Plant Biology and Mathematical Modeling. *Frontiers in Genetics*, *11*, 457. http://doi.org/10.3389/fgene.2020.00457

Van Den Burg, H. A., Tsitsigiannis, D. I., Rowland, O., Lo, J., Rallapalli, G., MacLean, D., … Jones, J. D. G. (2008). The F-box protein ACRE189/ACIF1 regulates cell death and defense responses activated during pathogen recognition in tobacco and tomato. *Plant Cell*, *20*(3), 697–719. http://doi.org/10.1105/tpc.107.056978

Van Der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. In *Journal of machine learning research* (Vol. 9, pp. 2579–2605).

Van Harsselaar, J. K., Lorenz, J., Senning, M., Sonnewald, U., & Sonnewald, S. (2017). Genome-wide analysis of starch metabolism genes in potato (Solanum tuberosum L.). *BMC Genomics*, *18*(1), 37. http://doi.org/10.1186/s12864-016-3381-z

VanRaden, P. M. (2008). Efficient methods to compute genomic predictions. *Journal of Dairy Science*, *91*(11), 4414–4423. http://doi.org/10.3168/jds.2007-0980

Varshavsky, A. (2005). Regulated protein degradation. In *Trends in Biochemical Sciences* (Vol. 30, pp. 283–286). http://doi.org/10.1016/j.tibs.2005.04.005

Väremo, L., Nielsen, J., & Nookaew, I. (2013). Enriching the gene set analysis of genome-wide data by incorporating directionality of gene expression and combining statistical hypotheses and methods. *Nucleic Acids Research*, *41*(8), 4378–4391. http://doi.org/10.1093/nar/gkt111

Veyrieras, J. B., Kudaravalli, S., Kim, S. Y., Dermitzakis, E. T., Gilad, Y., Stephens, M., & Pritchard, J. K. (2008). High-resolution mapping of expression-QTLs yields insight into human gene regulation. *PLoS Genetics*, *4*(10). http://doi.org/10.1371/journal.pgen.1000214

Vignes, M., Vandel, J., Allouche, D., Ramadan-Alban, N., Cierco-Ayrolles, C., Schiex, T., … Givry, S. de. (2011). Gene regulatory network reconstruction using bayesian networks, the dantzig selector, the lasso and their meta-analysis. *PLoS ONE*, *6*(12), 29165. http://doi.org/10.1371/journal.pone.0029165

Vogel, C., & Marcotte, E. M. (2012). Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nature Reviews Genetics*, *13*(4), 227–232. http://doi.org/10.1038/nrg3185

Voorrips, R. E., & Maliepaard, C. A. (2012). The simulation of meiosis in diploid and tetraploid organisms using various genetic models. *BMC Bioinformatics*, *13*(1), 248. `http://doi.org/10.1186/1471-2105-13-248`

Wagner, A., & Fell, D. A. (2001). The small world inside large metabolic networks. *Proceedings of the Royal Society B: Biological Sciences*, *268*(1478), 1803–1810. `http://doi.org/10.1098/rspb.2001.1711`

Walsh, C. T., Garneau-Tsodikova, S., & Gatto, G. J. (2005). Protein posttranslational modifications: The chemistry of proteome diversifications. *Angewandte Chemie - International Edition*, *44*(45), 7342–7372. `http://doi.org/10.1002/anie.200501023`

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., … Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, *11*(3), 333–337. `http://doi.org/10.1038/nmeth.2810`

Wang, K. C., & Chang, H. Y. (2011). Molecular Mechanisms of Long Noncoding RNAs. *Molecular Cell*, *43*(6), 904–914. `http://doi.org/10.1016/j.molcel.2011.08.018`

Wang, L., & Michoel, T. (2017). Efficient and accurate causal inference with hidden confounders from genome-transcriptome variation data. *PLOS Computational Biology*, *13*(8), e1005703. `http://doi.org/10.1371/journal.pcbi.1005703`

Wang, Y., Liu, C. L., Storey, J. D., Tibshirani, R. J., Herschlag, D., & Brown, P. O. (2002). Precision and functional specificity in mRNA decay. *Proceedings of the National Academy of Sciences*, *99*(9), 5860–5865. `http://doi.org/10.1073/pnas.092538799`

Wang, Y. X. R., & Huang, H. (2014). Review on statistical methods for gene network reconstruction using expression data. *Journal of Theoretical Biology*, *362*, 53–61. `http://doi.org/10.1016/j.jtbi.2014.03.040`

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. `http://doi.org/10.1038/30918`

Werij, J. S., Kloosterman, B., Celis-Gamboa, C., De Vos, C. H., America, T., Visser, R. G. F., & Bachem, C. W. B. (2007). Unravelling enzymatic discoloration in potato through a combined approach of candidate genes, QTL, and expression analysis. *Theoretical and Applied Genetics*, *115*(2), 245–252. `http://doi.org/10.1007/s00122-007-0560-y`

Wery, M., Kwapisz, M., & Morillon, A. (2011). Noncoding RNAs in gene regulation. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, *3*(6), 728–738. `http://doi.org/10.1002/wsbm.148`

Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, *10*(2), 122–133. `http://doi.org/10.1038/nrg2509`

Wilkinson, D. J. (2012). *Stochastic Modelling for Systems Biology* (2nd edition). CRC Press. `http://doi.org/10.1080/09332480.2012.752295`

Wingett, S. W., & Andrews, S. (2018). FastQ Screen: A tool for multi-genome mapping and quality control. *F1000Research*, *7*, 1338. `http://doi.org/10.12688/f1000research.15931.2`

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. In *Multivariate analysis*. Retrieved from http://arxiv.org/abs/arXiv:1011.1669v3

Wongchokprasitti, C. (2019). rcausal: R-Causal Library.

Woodhouse, M., Burkart-Waco, D., & Comai, L. (2009). Polyploidy. *Nature Education*.

Wright, P. R., Georg, J., Mann, M., Sorescu, D. A., Richter, A. S., Lott, S., … Backofen, R. (2014). CopraRNA and IntaRNA: Predicting small RNA targets, networks and interaction domains. *Nucleic Acids Research*, *42*(W1). http://doi.org/10.1093/nar/gku359

Wu, L., & Belasco, J. G. (2008). Let Me Count the Ways: Mechanisms of Gene Regulation by miRNAs and siRNAs. *Molecular Cell*, *29*(1), 1–7. http://doi.org/10.1016/j.molcel.2007.12.010

Xu, X., Pan, S., Cheng, S., Zhang, B., Mu, D., Ni, P., … Visser, R. G. F. (2011). Genome sequence and analysis of the tuber crop potato. *Nature*, *475*(7355), 189–195. http://doi.org/10.1038/nature10158

Xu, Y., Li, P., Yang, Z., & Xu, C. (2017). Genetic mapping of quantitative trait loci in crops. *The Crop Journal*, *5*(2), 175–184. http://doi.org/10.1016/j.cj.2016.06.003

Yan, J., Risacher, S. L., Shen, L., & Saykin, A. J. (2017). Network approaches to systems biology analysis of complex disease: integrative methods for multi-omics data. *Briefings in Bioinformatics*, *19*(6), 1370–1381. http://doi.org/10.1093/bib/bbx066

Yang, E., Nimwegen, E. van, Zavolan, M., Rajewsky, N., Schroeder, M., Magnasco, M., & Darnell, J. E. (2003). Decay rates of human mRNAs: Correlation with functional characteristics and sequence attributes. *Genome Research*, *13*(8), 1863–1872. http://doi.org/10.1101/gr.1272403

Yang, Z., & Michailidis, G. (2015). A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, *32*(1), btv544. http://doi.org/10.1093/bioinformatics/btv544

Yang, Z., Wu, N., Liang, Y., Zhang, H., & Ren, Y. (2020). SMSPL: Robust Multimodal Approach to Integrative Analysis of Multiomics Data. *IEEE Transactions on Cybernetics*, 1–14. http://doi.org/10.1109/tcyb.2020.3006240

Yeung, M. K. S., Tegnér, J., & Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(9), 6163–8. http://doi.org/10.1073/pnas.092576199

Yip, A. M., & Horvath, S. (2007). Gene network interconnectedness and the generalized topological overlap measure. *BMC Bioinformatics*, *8*(1), 22. http://doi.org/10.1186/1471-2105-8-22

Yu, J., Pressoir, G., Briggs, W. H., Bi, I. V., Yamasaki, M., Doebley, J. F., … Buckler, E. S. (2006). A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature Genetics*, *38*(2), 203–208. http://doi.org/10.1038/ng1702

Zeng, I. S. L., & Lumley, T. (2018). Review of Statistical Learning Methods in Integrated Omics Studies (An Integrated Information Science). *Bioinformatics and Biology Insights*, *12*, 117793221875929. http://doi.org/10.1177/1177932218759292

Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, *4*(1). http://doi.org/10.2202/1544-6115.1128

Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, *172*(16-17), 1873–1896. http://doi.org/10.1016/j.artint.2008.08.001

Zhang, K., & Chan, L. W. (2006). Extensions of ICA for causality discovery in the Hong Kong stock market. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vols. 4234 LNCS - III, pp. 400–409). Springer Verlag. http://doi.org/10.1007/11893295_45

Zhang, K., Schölkopf, B., Spirtes, P., & Glymour, C. (2018). Learning causality and causality-related learning: some recent progress. *National Science Review*, *5*(1), 26–29. http://doi.org/10.1093/nsr/nwx137

Zhang, Z., Ersoz, E., Lai, C. Q., Todhunter, R. J., Tiwari, H. K., Gore, M. A., … Buckler, E. S. (2010). Mixed linear model approach adapted for genome-wide association studies. *Nature Genetics*, *42*(4), 355–360. http://doi.org/10.1038/ng.546

Zhao, K., Aranzana, M. J., Kim, S., Lister, C., Shindo, C., Tang, C., … Nordborg, M. (2007). An Arabidopsis Example of Association Mapping in Structured Samples. *PLoS Genetics*, *3*(1), e4. http://doi.org/10.1371/journal.pgen.0030004

Zhao, K., Tung, C. W., Eizenga, G. C., Wright, M. H., Ali, M. L., Price, A. H., … McCouch, S. R. (2011). Genome-wide association mapping reveals a rich genetic architecture of complex traits in Oryza sativa. *Nature Communications*, *2*(1), 1–10. http://doi.org/10.1038/ncomms1467

Zhou, X., & Stephens, M. (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics*, *44*(7), 821–824. http://doi.org/10.1038/ng.2310

Zhu, D., & Qin, Z. S. (2005). Structural comparison of metabolic networks in selected single cell organisms. *BMC Bioinformatics*, *6*. http://doi.org/10.1186/1471-2105-6-8

Zhu, J., Lum, P. Y., Lamb, J., GuhaThakurta, D., Edwards, S. W., Thieringer, R., … Schadt, E. E. (2004). An integrative genomics approach to the reconstruction of gene networks in segregating populations. *Cytogenetic and Genome Research*, *105*(2-4), 363–374. http://doi.org/10.1159/000078209

Zhu, J., Sova, P., Xu, Q., Dombek, K. M., Xu, E. Y., Vu, H., … Schadt, E. E. (2012). Stitching together multiple data dimensions reveals interacting metabolomic and transcriptomic networks that modulate cell regulation. *PLoS Biology*, *10*(4), e1001301. http://doi.org/10.1371/journal.pbio.1001301

Zhu, J., Zhang, B., Smith, E. N., Drees, B., Brem, R. B., Kruglyak, L., … Schadt, E. E. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, *40*(7), 854–861. http://doi.org/10.1038/ng.167

Zierer, J., Pallister, T., Tsai, P. C., Krumsiek, J., Bell, J. T., Lauc, G., … Kastenmüller, G. (2016). Exploring the molecular basis of age-related disease comorbidities using a multi-omics graphical model. *Scientific Reports*, *6*(1), 1–10. http://doi.org/10.1038/srep37646

Zlatanova, J., & Van Holde, K. E. (2016). *Molecular Biology: Structure and Dynamics of Genomes and Proteomes* (p. 624). Garland Sciences.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429. `http://doi.org/10.1198/016214506000000735`

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*. `http://doi.org/10.1111/j.1467-9868.2005.00503.x`