NESTED TANDEM REPEAT COMPUTATION AND ANALYSIS

A thesis presented in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

in Computational Biology

at Massey University

Atheer Matroud

2013

Abstract

Biological sequences have long been known to contain many classes of *repeats*. The most studied repetitive structure is the *tandem repeat* where many approximate copies of a common segment (the *motif*) appear consecutively. In this thesis, a complex repetitive structure is investigated. This repetitive structure is called a *nested tandem repeat*. It consists of many approximate copies of two motifs interspersed with one another.

This thesis is a collection of published and in progress papers. Each paper addresses a computational problem related to the analysis of nested tandem repeats. Nested tandem repeats have been observed in the intergenic spacer of the ribosomal DNA gene in *Colocasia esculenta*. The question of whether such repeats can be found elsewhere in biological sequence databases is addressed and `NTRFinder`, a software tool to detect nested tandem repeats, is described. Another problem that arises after detecting a nested tandem repeat is the alignment of the nested tandem repeat region against its two motifs. An algorithm that guarantees an optimal solution to this problem is introduced. After detecting nested tandem repeats and identifying their structures, the identification of the motif boundaries is an unsolved problem which arises not only in nested tandem repeats but in tandem repeats as well. Heuristic solutions to this problem are implemented and tested. In order to compare two tandem repeat sequences an algorithm that aligns a hypothetical ancestral sequence of both sequences against each sequence is presented. This algorithm considers substitutions, deletions, and unidirectional duplication, namely, from ancestor to descendant.

Acknowledgements

I am sincerely thankful to my supervisors Prof. Mike Hendy and Dr. Chris Tuffley for their continuous support from day one through my candidature. It has been a privilege to work with both of them. Without their guidance and patience this work would have not been possible. I am forever grateful for their continuous support. I would also like to thank A/Prof. David Bryant for his support and help in my studies after I moved to Dunedin.

I would like to thank the Allan Wilson Centre for Molecular Ecology and Evolution for funding my study. I would also like to thank the Institute of Fundamental Sciences for the financial support they have provided to me.

I would like to thank Prof. Jens Stoye for the fruitful discussions I had with him during my visit to his lab and during our meeting in Campo Grande. Many thanks to A/Prof. Nadia El-Mabrouk for her support of my visit to her lab and for her advice. I would also like to thank A/Prof. Eric Rivals for his suggestions and comments.

I am grateful to Prof. Hamish Spencer for giving me his office to use. I am also thankful to the Department of Zoology staff for their hospitality. I am thankful to the Mathematics and Statistics department members at the University of Otago for hosting me and being so friendly.

I would like to thank all my colleagues in my office at Massey University and all colleagues in the "boffin lounge" for their company, help and support.

Last but not least, I am very grateful to my wife Zainab for her patience and strong support throughout my study. I am thankful to my children, Qaswar, Jaafar, and Durar for being such a great inspiration to me. I am grateful to my father, my mother, Naseer, Abeer, Akeel, and Aseel for their continued support.

Contents

Abstract iii

Acknowledgements v

1 Introduction **1**

1.1 Candidate's Note 2

1.2 Definitions 2

1.2.1 Sequences, edit operations and the edit distance 2

1.2.2 Classification of Tandem Repeats 3

1.2.3 Nested Tandem Repeats 4

1.3 A duplication model for tandem repeats and nested tandem repeats 6

1.4 Overview 8

2 Literature Review **11**

2.1 Motivation 11

2.2 Models of tandem repeat evolution 12

2.3 Detection of tandem repeats 13

2.4 Alignment 15

2.5 Alignment of two tandem repeat sequences 15

3 Observations on the nested tandem repeat found in taro **17**

3.1 Nested tandem repeat structures in NZ1 and JP1 18

3.2 Nested tandem repeat variants sequence 18

3.3 Variants graph 18

3.4 Expected number of parallel substitutions 23

3.5 Variants frequency distribution 24

3.6 Variants spread 25

4 NTRFinder: A Software Tool to Find Nested Tandem Repeats 27

4.1 Abstract . . . 27

4.2 Introduction . . . 27

4.3 Material and Methods . . . 28

4.4 Results . . . 31

4.4.1 Tests on simulated data . . . 31

4.4.2 Tests on real sequence data . . . 32

4.4.3 More complex structures . . . 32

4.4.4 Running time . . . 32

4.5 Discussion . . . 33

4.6 Conclusion . . . 33

5 An algorithm to solve the motif alignment problem for approximate nested tandem repeats in biological sequences 39

5.1 Abstract . . . 39

5.2 Introduction . . . 40

5.3 Definitions . . . 41

5.3.1 Alphabets and strings . . . 41

5.3.2 The edit distance . . . 41

5.3.3 Tandem repeats and nested tandem repeats . . . 42

5.3.4 Alignment . . . 43

5.4 The motif alignment problem for approximate nested tandem repeats . . . 44

5.4.1 The problem . . . 44

5.4.2 Solution to the problem via nested wrap-around dynamic programming . . . 44

5.4.3 Correctness of the algorithm . . . 46

5.4.4 Extension to nested tandem repeats with three or more motifs . . . 49

5.5 Conclusion . . . 50

6 A comparison of three heuristic methods for solving the parsing problem for tandem repeats 51

6.1 Abstract . . . 51

6.2 Introduction . . . 52

6.3 Definitions and Background 54
6.4 The importance of the parsing problem 55
6.5 Heuristic methods to estimate tandem repeat parsing 56
6.5.1 PAIR — the adjacent pairs method 58
6.5.2 VAR — the number of variants method 59
6.5.3 MST — the minimum spanning tree method 59
6.6 Results and discussion 60
6.6.1 Tests on simulated data 60
6.6.2 Tests on real sequence data 65
6.7 Conclusion 66

7 Ancestor-descendant alignment of tandemly repeated sequences 71
7.1 TR maps 72
7.2 Edit operations and edit distance 73
7.3 Ancestor-descendant repeat distance 74
7.3.1 The ancestor-descendant alignment problem for (N)TR sequences 74
7.3.2 Solution to the ancestor-descendant alignment problem 76
7.3.3 Correctness of the algorithm 77
7.4 A Longest Common Subsequence approach to estimating the most recent common ancestor 83
7.5 An application to real DNA sequences 83
7.6 Conclusion 84

8 Conclusion 87
8.1 Future work 88

A Published chapters 91

Chapter 1

Introduction

The subject of this thesis is the detection and analysis of *nested tandem repeats* (NTRs), which are repetitive structures found in DNA consisting of repeats of two different motifs interspersed with each other. The initial motivation for the work came from the discovery of NTRs in the intergenic spacer region of the rDNA in the taro *Colocasia esculenta*. Taro is a staple food crop that is widely spread around the world. The origin of taro is believed to be Southeast Asia (Matthews, 1991), and its dispersal was helped by the migration of people around the world. Taro is one of the important food sources in the Pacific and it is spread all over the Pacific islands. A population genetic study of this plant may lead to a better understanding of Polynesian migration history.

The function and the implication of NTRs in the genome are not well understood, nor is the mechanism that generates them. Finding these repetitive structures and identifying their functions are fundamental objectives of biologists. NTRs were only recently observed in DNA sequences (Newman and Cooper, 2007; Hauth and Joseph, 2002; Rolland et al., 2010), hence, there are few software tools that can help biologists to analyse them. In this thesis, our main goal is to develop tools to analyse NTRs and facilitate their use as genetic markers for evolutionary studies. However, some of our results can be used to analyse tandem repeats too.

In general, ordinary tandem repeats have been known for much longer (Hatch et al., 1976), and it is known that tandem repeats have implications for some genetic diseases such as FragileX, myotonic dystrophy, and Huntington diseases (Verkerk et al., 1991; Fu et al., 1992; Verkerk et al., 1993). It is also known that some tandem repeats have a significant role in some cancers (Buard and Jeffreys, 1997). However, most are thought

to be neutral hence their role in population genetic studies.

Tandem repeats are observed to contain polymorphism in the number of copies and copy variants. An interesting application of tandem repeat sequences is the study of human migration history. For example, (Armour et al., 1996) use the tandem repeats MS205 (D76S309) in the human Y chromosome to support the recent African origin for modern human diversity. We expect NTRs to provide yet another source of valuable genetic information.

In the next section, some definitions are introduced along with some examples followed by an overview of the thesis.

1.1 Candidate's Note

This thesis is written based on a collection of papers I have worked on through my PhD candidacy. Each chapter is in a stand-alone format. This means there is some redundancy in the contents of the thesis. I have tried to eliminate as much redundancy as possible and be as consistent as possible through the whole thesis.

1.2 Definitions

In this section, some terms that will be used throughout the thesis are defined.

1.2.1 Sequences, edit operations and the edit distance

A DNA sequence is a sequence of symbols from the nucleotide alphabet $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. We define a DNA *segment* to be a string of contiguous DNA nucleotides and define a *site* to be a position in a segment. For a DNA segment

$$\mathbf{X} = x_1 x_2 \cdots x_n,$$

$x_i \in \Sigma$ is the nucleotide at the i-th site and $|\mathbf{X}| = n$ is the length of $\mathbf{X}$.

Copying errors happen in DNA replication due to different factors. These changes are on different scales, that is changes on the nucleotides level which include substitution, insertion, and deletion of single nucleotides and changes on the segments level such as

duplication and segment deletion. We refer to these as *edit operations*. These operations may be used to define a distance function on segments, by associating a weight to each edit operation. We can then in principle find a series of edit operations, which transform segment $\mathbf{X}$ to segment $\mathbf{Y}$, of minimal total weight. We will refer to this sum as the *edit distance*, and denote it by $d(\mathbf{X}, \mathbf{Y})$. These weights represent the cost associated with the edit operation; less frequent operations will have a greater weight, corresponding to a higher cost. For the purposes of this thesis, the edit operations allowed in calculating the edit distance between segments are restricted to single nucleotide substitutions, and single nucleotide insertions or deletions (indels).

For notational purposes, let θ denote a single nucleotide transformation $\theta \in \{\alpha, \beta, \gamma\}$, where, following the Kimura 3ST substitution model (Kimura, 1981), the transformation types are

$$\alpha = \mathtt{A} \leftrightarrow \mathtt{G},\ \mathtt{C} \leftrightarrow \mathtt{T};$$

$$\beta = \mathtt{A} \leftrightarrow \mathtt{T},\ \mathtt{G} \leftrightarrow \mathtt{C};$$

$$\gamma = \mathtt{A} \leftrightarrow \mathtt{C},\ \mathtt{G} \leftrightarrow \mathtt{T}.$$

The *substitution* $i\theta$ in a segment S denotes θ applied to the nucleotide at the site i in S.

1.2.2 Classification of Tandem Repeats

Many classifications of tandem repeat schemas have been introduced in the computational biology literature. We list some classifications which are commonly used:

- **(Exact) Tandem Repeats:** An *exact tandem repeat* (TR) is a sequence comprising two or more contiguous copies $\mathbf{X}\mathbf{X}\cdots\mathbf{X}$ of identical segments $\mathbf{X}$ (referred to as the *motif*).

- **$k-$Approximate Tandem Repeats:** A *$k-$approximate tandem repeat* ($k-$TR) is a sequence comprising two or more contiguous copies $\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_n$ of similar segments, where each individual segment $\mathbf{X}_i$ is edit distance at most k from a template segment $\mathbf{X}$.

- **Multiple Length Tandem Repeats (MLTR):** A multiple length tandem repeat is a tandem repeat of the form $(\mathbf{X}\mathbf{x}^n)^m$, where n is a constant larger than one and

$d(\mathbf{X}, \mathbf{x})$ is greater than some threshold value h.

Example Below is a list of examples for each of the repeat classes:

- **Tandem repeat:**
  AGG AGG AGG AGG AGG. The motif is AGG.
- $1-$**Tandem repeat:**
  AGG AGC ATG AGG CGG. The template motif is AGG.
- **Multiple length tandem repeat**:
  GACCTTTGG ACGGT ACGGT ACGGT GACCTTTGG ACGGT ACGGT ACGGT.
  The motifs are $\mathbf{x} =$ ACGGT and $\mathbf{X} =$ GACCTTTGG, with $n = 3, m = 2$.

Approximate tandem repeats are also classified based on the length of their repeated motif. These classes are: *microsatellites*, where the length of the repeated motif is in the range 1-6 bp (Jarne and Lagoda, 1996); *minisatellites*, where the length of the repeated motif is in the range 7-100 bp (Buard and Jeffreys, 1997); and *satellites* or *megasatellites* for any repeated motif of length above 100 bp (Rolland et al., 2010).

1.2.3 Nested Tandem Repeats

In this section, a more complex repetitive structure is introduced, the nested tandem repeat (NTR), also referred to as a *variable length tandem repeat* (Hauth and Joseph, 2002). Let $\mathbf{X}$ and $\mathbf{x}$ be two segments (typically of different lengths) from the alphabet $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, such that $d(\mathbf{X}, \mathbf{x})$ is greater than some threshold value h.

Definition 1. An *exact nested tandem repeat* is a string of the form

$$\mathbf{x}^{s_0}\mathbf{X}\mathbf{x}^{s_1}\mathbf{X}\cdots\mathbf{X}\mathbf{x}^{s_n},$$

where $n > 1$, $s_i \geq 1$ for each $0 < i < n$, and $s_j \geq 2$ for some $j \in \{0, 1, \cdots, n\}$. The motif $\mathbf{x}$ is called the *tandem motif* and the motif $\mathbf{X}$ is the *interspersed motif*. The concatenations of the tandem repeats $\mathbf{x}^{s_i}$ alone, and of the interspersed motifs $\mathbf{X}$ alone, each form exact tandem repeats. We allow the possibilities $s_0 = 0$ and $s_n = 0$, so that

the NTR can start and/or finish with the interspersed motif $\mathbf{X}$, and we will describe the structure of the above NTR by specifying the $(n+1)$-tuple $(s_0, s_1, \ldots, s_n)$.

Example $\mathbf{x} = \texttt{ACGGT}$, $\mathbf{X} = \texttt{GACCTTTGG}$, $n = 7$, $s_0 = 0$, $s_1 = 3$, $s_2 = 5$, $s_3 = 2$, $s_4 = 4$, $s_5 = 1$, $s_6 = s_7 = 2$, so

$$\mathbf{x}^0 \prod_{i=1}^{7} \mathbf{X}\mathbf{x}^{s_i} = \mathbf{X}\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{X}\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{X}\mathbf{x}\mathbf{x}\mathbf{X}\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{X}\mathbf{x}\mathbf{X}\mathbf{x}\mathbf{x}\mathbf{X}\mathbf{x}\mathbf{x}$$

$$= \texttt{GACCTTTGG ACGGT ACGGT ACGGT}$$

$$\texttt{GACCTTTGG ACGGT ACGGT ACGGT ACGGT ACGGT}$$

$$\texttt{GACCTTTGG ACGGT ACGGT}$$

$$\texttt{GACCTTTGG ACGGT ACGGT ACGGT ACGGT}$$

$$\texttt{GACCTTTGG ACGGT}$$

$$\texttt{GACCTTTGG ACGGT ACGGT}$$

$$\texttt{GACCTTTGG ACGGT ACGGT}.$$

The structure of this NTR is given by (0,3,5,2,4,1,2,2).

In practice, it is expected that any nested tandem repeats occurring in DNA sequences will be approximate rather than exact. In what follows, we will write $\tilde{\mathbf{X}}$ to mean an approximate copy of the motif $\mathbf{X}$, and $\tilde{\mathbf{x}}^s$ to mean an approximate tandem repeat consisting of s (not necessarily identical) approximate copies of the motif $\mathbf{x}$.

Definition 2. A *(k_1,k_2)-approximate nested tandem repeat* is a string of the form

$$\tilde{\mathbf{x}}^{s_0}\tilde{\mathbf{X}}\tilde{\mathbf{x}}^{s_1}\tilde{\mathbf{X}}\cdots\tilde{\mathbf{X}}\tilde{\mathbf{x}}^{s_n},$$

where n and s_i satisfy the same conditions as in Definition 1, and $\tilde{\mathbf{x}}^{s_0}\tilde{\mathbf{x}}^{s_1}\cdots\tilde{\mathbf{x}}^{s_n}$ is a k_1-approximate tandem repeat with motif $\mathbf{x}$, and $\tilde{\mathbf{X}}\tilde{\mathbf{X}}\cdots\tilde{\mathbf{X}}$ is a k_2-approximate tandem repeat with motif $\mathbf{X}$.

Example Below is an exact nested tandem repeat and an example of an approximate nested tandem repeat.

- **NTR**:

  AGG AGG CTCAG AGG CTCAG AGG AGG AGG CTCAG.

  The template motifs are AGG, CTCAG.

- $(1,2)-$**NTR**:

  AGA AGG CTTCG AGG CTCAG AG AGA AGG CTTCG AGG CTCAG AAG.

  The template motifs are $\mathbf{x}$ = AGG, $\mathbf{X}$ = CTCAG.

1.3 A duplication model for tandem repeats and nested tandem repeats

Let S be a DNA sequence of symbols from the nucleotide alphabet $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, and suppose that S has the structure of a tandem repeat **TR** or nested tandem repeat **NTR**. To study Swe can map it to a macro alphabet $\Sigma_v = \{a, b, \ldots, A, B, \ldots\}$, whose symbols represent the motif variants occurring in S. In the case of a nested tandem repeat, we will use lower case letters for variants of the tandem motif, and upper case letters for variants of the interspersed motif. We define an (N)TR map of S to be the sequence S_v obtained by replacing each motif of S by the corresponding symbol in Σ_v. This process is also known as variant mapping (Berard and Rivals, 2003).

An evolution model on S_v is defined by the following edit operations:

- k**-Duplication:** the process of copying a substring of length k and placing it after the duplicated segment, for example, a 2–duplication: $a(\underline{bc}) \rightarrow a(bc)(bc)$, a 1–duplication $(\underline{a})bc \rightarrow (a)(a)bc$.

- k**–Deletion:** the process that removes k contiguous symbols from S_v. For example, a 1–deletion $a\underline{b}c \rightarrow ac$.

- **repeat copy substitution:** the process of replacing a symbol $a \in \Sigma_v$ with another symbol $b \in \Sigma_v$ by applying single nucleotide events such as deletion, insertion, and substitution.

In this thesis, the model of duplication considered is obtained from (Sammeth and Stoye, 2006), which suggests that duplications and deletions may occur at any position in the sequence S_v, and they may have any size (k-duplication and k-deletion). The

duplication operation in this model is of arity 1 (1-duplication) (Rivals, 2004b; Sammeth and Stoye, 2006; Berard and Rivals, 2003).

In the case of nested tandem repeats, where interspersed motifs $\{A, B, \dots\}$ are not observed adjacent to each other, it is assumed that duplications always start or end with a symbol from the alphabet $\{a, b, \dots\}$. Moreover, a deletion is assumed not to start at the end of one interspersed motif and end at the start of another interspersed motif.

In this model, it is also assumed that duplication and substitution events occur at a fixed relative rate, and that the motif copies remain contiguous and oriented in the same direction in the genome. Under these assumptions the duplication history of a tandem repeat can be described by a duplication history tree (DHT).

Example We illustrate in Figure 1.1 how a DHT may be inferred from a tandem repeat with chosen start and end boundaries. Consider the following sequence which contains the tandem repeat

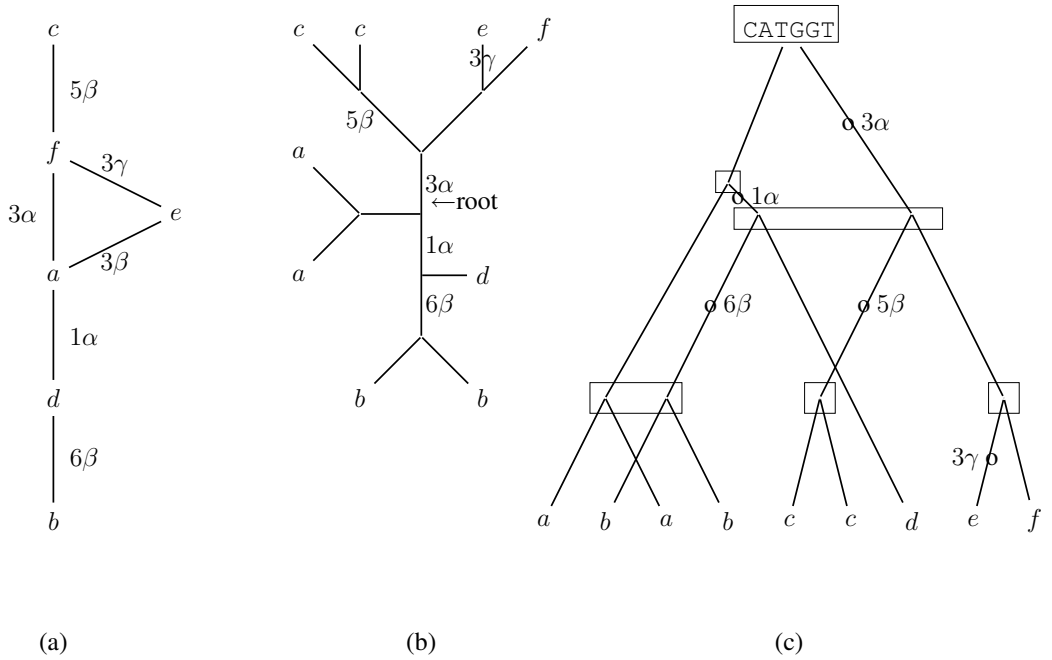
$$\begin{aligned}\texttt{TTATGT}&\texttt{CATGGT}\texttt{TATGGA}\texttt{CATGGT}\texttt{TATGGA}\texttt{CACGCT}\\ &\texttt{CACGCT}\texttt{TATGGT}\texttt{CAAGGT}\texttt{CACGGT}\texttt{CAATAG},\end{aligned} \quad (1.1)$$

which for the parsing displayed, is an approximate tandem repeat with mode motif CATGGT. There are six motif variants, in order as $ababccdef$, where

$$\begin{aligned} a &= \texttt{CATGGT}, \quad b = \texttt{TATGGA}, \quad c = \texttt{CACGCT},\\ d &= \texttt{TATGGT}, \quad e = \texttt{CAAGGT}, \quad f = \texttt{CACGGT}. \end{aligned} \quad (1.2)$$

These variants may be represented by the graph in Figure 1.1(a), in which the edge between variants u and v is labelled by the substitution $i\theta$ that transforms u into v. Figure 1.1(c) shows a DHT in which each edge is labeled with zero or more substitutions of the form $i\theta$.

By removing one edge of the $a - e - f$ cycle (edge $a - e$ was chosen arbitrarily) in Figure 1.1(a) and adding leaves for each of the 9 segments, we obtain the maximum parsimony tree in Figure 1.1(b). In Figure 1.1(b) if we place the root of the tree on the edge arrowed, we get a DHT (this edge is the only edge where a root can be placed to get



(a) (b) (c)

Figure 1.1: Construction of a duplication history tree for the tandem repeat in Equation (1.2). See text for details.

a duplication tree (Gascuel et al., 2003)). Each duplication is identified by the segment to be duplicated enclosed in a rectangle. When the duplicated block encloses more than one segment, the descendant motifs alternate as shown in Figure 1.1(c). The approximate tandem repeat is fully described by the duplication tree T with 6 duplications, the ancestral motif at the root (`CATGGT`), and the 5 substitutions on the edges of T.

1.4 Overview

This thesis investigates *Nested Tandem Repeats* (NTRs). In Chapter 2, an overview of related work is presented. A detailed analysis of the nested tandem repeat in taro is introduced in Chapter 3, where some observations on the nested tandem repeat structure are presented.

After having a close look at the nested tandem repeats in taro, we set our first target to search for nested tandem repeat structures in order to understand their distribution in DNA sequences. This target was the main motivation for building the software tool `NTRFinder`. `NTRFinder` is introduced in Chapter 4. The algorithm has been tested on both real and simulated data. A list of nested tandem repeats found in some real DNA sequences is presented.

Once NTRs are found, an alignment algorithm to solve the problem of aligning two motifs against a region that contains the NTR is crucial. It is crucial not only for the verification phase in the `NTRFinder` program but also for the analysis phases. Chapter 5 describes an alignment algorithm for the verification phase of the software tool NTRFinder developed for database searches for NTRs. When the search algorithm has located a subsequence $\mathbf{T}$ containing a possible NTR, with motifs $\mathbf{X}$ and $\mathbf{x}$, a verification step aligns $\mathbf{T}$ against an exact NTR built from the templates $\mathbf{X}$ and $\mathbf{x}$, to confirm whether $\mathbf{T}$ contains an approximate NTR and determine its extent. Chapter 5 describes an algorithm to solve this alignment problem in $O(|\mathbf{T}|(|\mathbf{X}| + |\mathbf{x}|))$ space and time.

An important step before starting the analysis of an NTR is to identify the repeated motif pattern. Namely, it is important to know where the boundaries of the repeated pattern are. We call the problem of inferring the motif boundaries *the parsing problem for nested tandem repeats*. In Chapter 6, three heuristic methods for solving the parsing problem, under the assumption that the parsing is fixed throughout the duplication history of the tandem repeat, are proposed and compared. The three methods are: PAIR, which minimises the number of pairs of common mutations which span a boundary; VAR, which minimises the total number of variants of the motif; and MST, which minimises the length of the minimum spanning tree connecting the variants, where the weight of each edge is the Hamming distance of the pair of variants. These methods were tested on simulated data (for which the true parsing is known) over a range of motif lengths and relative rates of substitutions to duplications, and these tests show that all three perform better than choosing the parsing arbitrarily (note, when choosing the boundary arbitrarily, we expect to hit the true boundary with frequency $\frac{1}{\ell}$, where ℓ is the length of the repeated motif). Of the three, MST typically performs the best, followed by VAR then PAIR. To test the methods on real data, the three methods were applied on four tandem repeats that belong to two different families. Our expectation is that tandem repeats that belong to the same family will have similar parsing points.

Our main goal is to use NTRs structures as markers to build phylogenies. In Chapter 7, the problem of comparing two repeated structures is investigated. This comparison involves reconstructing an approximate ancestral sequence then aligning it against each sequence. An algorithm to align an ancestral sequence against its descendant sequence is constructed. This algorithm has quadratic time and space complexity. The algorithm

produces an asymmetric alignment, where the duplication events happen in the ancestral sequence and are observed in the descendant sequence, but not the other way around.

Chapter 2

Literature Review

2.1 Motivation

The focus of this thesis is on *nested tandem repeats*. Nested tandem repeats are a complex repetitive structure containing two motifs, which were first noticed by Matthews et al. (1992). The rapid development in sequencing technology has permitted us to get more DNA sequences, and hence given us the opportunity to explore the world of DNA rigorously. Recently, several nested tandem repeats have been reported (Hauth and Joseph, 2002; Newman and Cooper, 2007; Rolland et al., 2010) which leads to the question of how common nested tandem repeats are in biological sequences, and what are their roles.

Tandem repeats, on the other hand, have been known for much longer (Hatch et al., 1976). Tandem repeats exist in most DNA sequences and some genomes consist of more than 50% repeats. Tandem repeats are classified based on the length of their motif (e.g. microsatellite, minisatellite, satellite/megasatellite).

Microsatellites are tandem repeats with motifs of no more than 6 bp long. They are the most studied repetitive structures, due to their simple structure and their wide distribution throughout most eukaryotic genomes. It is known that the existence of some microsatellites have health implications (Verkerk et al., 1993, 1991; Boland and Goel, 2010). Microsatellites are commonly used as genetic markers in evolutionary studies, mainly due to their length polymorphism. The length polymorphism feature makes the microsatellites easy to genotype using fragment length analysis (Waters and Wallis, 2000; Nikula et al., 2011; Queller et al., 1993; Dib et al., 1996). A comprehensive literature review of microsatellites and their implications, models of evolution, and applications can

be found in (Goldstein and Schlotterer, 1999).

Minisatellites are tandem repeats with motif length in the range 7-100 bp. They are more difficult to sequence and assemble than microsatellites. They are difficult to detect due to a high evidence of polymorphism (single substitutions, single indels) amongst repeat copies. Minisatellites have also proved to have health implications (Buard and Jeffreys, 1997; Bois and Jeffreys, 1999). Due to the polymorphism in the repeated copies and the copy numbers, minisatellites are used as genetic markers for phylogenetic studies (Jeffreys et al., 1991).

Recently, advances in computational tools have facilitated the detection of larger tandem repeats such as satellites and megasatellites (Rolland et al., 2010). However, the implications of these large repeats are yet to be understood.

The implications and applications of tandem repeats raise the question of whether nested tandem repeats have similar implications and applications. The goal of this thesis is to investigate nested tandem repeats, in particular, to detect and analyse nested tandem repeats.

2.2 Models of tandem repeat evolution

In this section, we give an overview of the common evolution models for tandem repeats, with an emphasis on minisatellites.

In the literature, there are several models for tandem repeat evolution. Microsatellites have been studied more than other tandem repeat categories, and therefore a larger number of evolution models have been introduced. It is acknowledged that the main mutational mechanism that affects microsatellites is replication slippage, a process that duplicates one or more and removes one or more repeat units (Levinson and Gutman, 1987). Other mutational mechanisms may be single nucleotide substitutions and duplications (Goldstein and Schlotterer, 1999). Microsatellites are simple repetitive structures yet they have complex histories of evolution. The mutational process on microsatellites depends on many factors such as the number of copies, length of the motif, GC content and the location in the genome (Goldstein and Schlotterer, 1999).

The mechanisms that generate minisatellites are not yet well understood. The widely used minisatellite mutational mechanisms are duplication, single substitutions, single in-

dels, and segmental deletions. Duplication is the process of copying one or more motif copies in tandem. Some models allow duplication of only one copy at a time (Berard and Rivals, 2003), whereas other models allow for more than one copy to be duplicated, e.g. (Sammeth and Stoye, 2006). The start of the duplication is typically considered to fall on the motif boundary (fixed boundaries) (Fitch, 1977b; Sammeth and Stoye, 2006; Berard and Rivals, 2003), however, Benson and Dong (1999a) have proposed a different model of evolution where duplications may start at any site (dynamic boundaries).

2.3 Detection of tandem repeats

Various algorithms have been introduced to find exact tandem repeats. Such algorithms were developed mainly for theoretical purposes, namely, to solve the problem of finding squares in strings (i.e. adjacent repeats) (Apostolico and Preparata, 1983; Crochemore, 1981; Kolpakov et al., 2001; Main and Lorentz, 1984; Stoye and Gusfield, 2002). These algorithms are not easily adapted to finding the approximate tandem repeats that usually occur in DNA.

A number of algorithms (Delgrange and Rivals, 2004; Landau et al., 2001) consider motifs differing only by substitutions, using the Hamming distance as a measure of similarity.

Most algorithms used to search biological sequences take into account insertions and deletions. They generally have two phases, a scanning phase that locates candidate tandem repeats, and an analysis phase that checks the candidate tandem repeats found during the scanning phase e.g. (Benson, 1999; Hauth and Joseph, 2002; Domaniç and Preparata, 2007; Wexler et al., 2005).

Benson (1999) addressed the problem of finding tandem repeats of different lengths. Benson's program scans the sequence once looking for all exact k-tuple matches (there are 4^k possible k-tuples) and records their positions. A list of distances is created to record the differences between the indices of subsequent occurrences for each k-tuple. The list of distances is used in two tests (the sum of heads and apparent size criteria tests) to detect candidate tandem repeats. The program uses another two tests to help cut off spurious signals; these are the random walk test and waiting time test. Any candidate tandem repeats that pass these tests are then verified by an alignment algorithm in the

verification phase.

`ATRHunter` due to Wexler et al. (2005) is another program constructed to find tandem repeats. The screening phase scans the whole sequence once for each motif length. To detect tandem repeats of length ℓ, the similarity between adjacent sequences of length ℓ is tested. The similarity test is carried out by running two windows of length k through the sequence, a distance ℓ apart. The size of k depends on ℓ; for ℓ in the interval [7,100], k is in the interval [3,5]. For each pair of adjacent sequences of length ℓ, a vector of length $\ell - k + 1$ is created, where a 1 is recorded if the two windows are approximate matches, and 0 otherwise. Score and gap criteria on the vectors are tested to decide if these adjacent segments are candidate tandem repeats. The number of matches must be larger than a threshold value $S_\ell(i)$, and the number of consecutive mismatches must be smaller than $\Delta_\ell(i)$. These thresholds are determined based on random walks on a graph whose vertices represent all binary strings of length k. In the verification phase, a global alignment is done between every pair of adjacent segments which were reported as candidate tandem repeats. If the score of the alignment is larger than a given threshold value, a tandem repeat is reported, otherwise the candidate is dismissed.

Domaniç and Preparata (2007) introduced a new algorithm to find tandem repeats. The main innovation of their algorithm is in the detection phase. A window of length k is run through the sequence, and at each point the position of the immediately preceding occurrence of the current k-tuple is recorded.

To date, the only algorithm specifically designed to look for NTRs is that of Hauth and Joseph (2002), which searches for tandem motifs of length at most six nucleotides. A more general definition of tandem repeats is introduced, as well as a definition of nested tandem repeats. Their algorithm is able to find most short tandem repeats, due to their conserved structure. It also finds nested tandem repeats where the repeat motif is no more than 6 bp long. The detection component has a window of length k that screens the sequence and creates a histogram of distances by recording the distance to the previous occurrence of the k-tuple. Then peaks of the histogram are further investigated.

2.4 Alignment

String similarity problems arise in many contexts, and as a result many algorithms exist to address them. Finding the exact similarity between two strings is a fundamental computer science problem, and a number of good solutions have been introduced by several authors (see (Gusfield, 1997) for an overview). However, such exact matching algorithms generally are not useful when applied to molecular data, which tend to contain approximate rather than exact matches due to the mutations that have occurred over time.

Many string similarity problems of biological interest can be phrased as alignment problems (for a precise definition of alignment, see Section 5.3.4). These include the problem of aligning two entire strings $\mathbf{A}$ and $\mathbf{B}$ (global alignment (Needleman and Wunsch, 1970)); the problem of aligning substrings of a string $\mathbf{A}$ against substrings of $\mathbf{B}$ (local alignment (Smith and Waterman, 1981)); and the problem of finding all occurrences of string $\mathbf{B}$ within string $\mathbf{A}$. See (Navarro, 1999) for a survey. Such alignment problems are commonly solved using the technique of dynamic programming.

Of greatest interest to us is the problem of finding the substring of $\mathbf{T}$ which best matches a substring of $\mathbf{x}^s$ for some $s > 1$ (tandem repeat alignment). To solve this problem efficiently Fischetti et al. (1993) introduced wrap-around dynamic programming, which has $O(|\mathbf{T}||\mathbf{x}|)$ space and time complexity. Chapter 5 solves the motif alignment problem for nested tandem repeats by extending the algorithm of Fischetti et al. (1993).

2.5 Alignment of two tandem repeat sequences

Tandem repeats are informative markers for phylogenetic studies due to the high polymorphism in the number of motif copies as well as variation in the motif. Tandem repeat genotyping has been used for many genetically monomorphic bacterial pathogens, such as *Yersinia pestis* (Klevytska et al., 2001), *Bacillus anthracis* (Keim et al., 2000), and *Mycobacterium leprae* (Truman et al., 2004).

Comparing tandem repeats and finding the distance between them is the first step toward inferring evolutionary relationships. A pairwise tandem repeat alignment can be considered as a primary goal to achieve a multiple tandem repeat alignment. The pairwise tandem repeat alignment problem has been addressed under different tandem repeat evolution models. Benson and Dong (1999b) developed exact and heuristic algorithms for

comparing and aligning two tandem repeat sequences. Their model considers dynamic boundaries, which means a duplication can occur at any position in the nucleotide sequence. Alignment of tandem repeats under insertion, substitution, duplication and deletion of a single segment has been introduced by Behzadi and Steyaert (2003) and Berard and Rivals (2003), Their algorithms have cubic time complexity. A more general model of evolution, where a duplication of any size can occur (one or more adjacent copies of the motif are duplicated in a single duplication event), is considered by Sammeth and Stoye (2006). They introduced an algorithm to align tandem repeats; however, their algorithm has exponential time complexity.

The algorithms which have been introduced in the literature are either restricted to single duplication (such as (Behzadi and Steyaert, 2003) and (Berard and Rivals, 2003)), which cannot be easily extended to multi-duplication, or are computationally expensive (such as (Sammeth and Stoye, 2006)). In Chapter 7, we introduce an algorithm that estimates the distance between two tandem repeats. Our algorithm has quadratic time and space complexity.

Chapter 3

Observations on the nested tandem repeat found in taro

In this chapter, we discuss some observations on the nested tandem repeats in taro. Taro, *Colocasia esculenta*, is a crop which belongs to the plant family Araceae. Taro is spread from Southeast Asia to southern China, Australia and Melanesia (Matthews, 1991). The population genetic study of this plant could provide information on the spread of agriculture and trade in these regions.

The nested tandem repeats (NTRs) in taro were first observed by (Matthews et al., 1992) in a 2800bp segment. With the help of newer sequencing technology a more detailed analysis of NTRs suggests that they can be useful phylogenetic markers. The eventual goal of this study is to sequence cultivars from the Pacific region and use nested tandem repeats as a marker to build a phylogeny of those cultivars. The nested tandem repeats found in taro exist in the intergenic spacer of the nuclear ribosomal DNA gene. The rDNA genes exist in arrays of hundreds of copies in the nuclear genome.

At this stage of the study (January, 2013), I have been provided with full sequences of the NTR region for two cultivars, one from New Zealand (NZ1) and one from Japan (JP1). The two sequences show a substantial similarity (similar NTR structures and similar sets of motif variants), and this clearly shows that the NTR structure is naturally present in *Colocasia esculenta*. The wide distribution of the NTRs in taro, in diploids and triploids, and in wild and cultivated forms, makes it likely that the NTR structure is ancient in this plant. Detailed analysis of these two NTRs is presented in this chapter.

motif	length	pattern
x	11	TCGCACAGCCG
X	48	TTCTGGGCAAAACGGCTGGGCGACGTGCTGGACTGGCCAGCTGGTTCG

Table 3.1: The consensus motifs x and X which form the nested tandem repeats in taro.

3.1 Nested tandem repeat structures in NZ1 and JP1

The nested tandem repeats (NTRs) in NZ1 and JP1 show a substantial similarity which suggests that they are homologous. The NTRs in NZ1 and JP1 consist of two repeated motifs interspersed with one another. The two consensus motifs are listed in Table 3.1.

In NZ1, there are 89 approximate copies of motif x and 12 approximate copies of motif X, whereas in JP1 there are 98 approximate copies of motif x and 13 approximate copies of motif X. Recall from section 1.2.3 that the structure of an NTR of the form $\mathbf{x}^{s_0}\mathbf{X}\mathbf{x}^{s_1}\mathbf{X}\cdots\mathbf{X}\mathbf{x}^{s_n}$, may be presented in the form $(s_0, s_1, \ldots, s_n)$. Using this notation, the nested tandem repeat structures in NZ1 and JP1 are as follows:

NZ1: (5,3,1,6,10,5,10,8,13,14,13,4)

JP1: (4,3,1,6,7,8,4,10,12,10,16,13,4)

The JP1 48 bp repeats are shown in Table 3.2.

3.2 Nested tandem repeat variants sequence

There are 21 variants of the tandem motif in the NTRs in NZ1 and JP1, with the motif TCGCACAGCCG being the most frequent variant in each. These variants are listed in Table 3.3. These variants differ from each other by single nucleotide substitutions, apart from variant 'v' which is of length 13. The frequencies of the variants follow a power law (see Figure 3.1).

3.3 Variants graph

As a result of mutational events that happened in the past, at the present time, we observe an NTR containing a set of variants. Each variant may have a frequency which is the number of its occurrences in the NTR. These frequencies depends on the rate of mutation (single nucleotide mutation), rate of duplication and rate of deletion (segment duplication

	1	2	3	4	5	6	7	8	9	10	11	12	13	13a	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	[T]	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	[C]	[A]	[T]	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	[T]	G	A	C	G	T	G	C	T	G	[A]	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
B	T	T	C	T	G	G	G	C	A	A	A	[C]	[A]	-	G	G	C	T	[A]	G	G	C	G	A	C	G	T	G	C	T	G	G	[G]	C	T	G	G	C	C	A	[T]	C	T	G	G	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
B	T	T	C	T	G	G	G	C	A	A	A	A	[A]	-	G	G	C	[C]	[A]	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	[A]
B	T	T	C	T	G	G	G	C	A	A	A	A	[A]	-	G	G	C	T	[A]	G	G	[T]	G	A	C	G	T	G	C	T	G	[A]	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
A	T	T	C	T	G	G	G	C	A	A	A	A	C	-	G	G	C	T	G	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	G	C	C	A	G	C	T	G	G	T	T	C	G
B	T	T	C	T	G	G	G	C	A	A	A	A	[A]	A	G	G	C	T	[A]	G	G	C	G	A	C	G	T	G	C	T	G	G	A	C	T	G	-	C	C	A	G	C	T	G	G	T	T	C	G

Table 3.2: The twelve 48bp motifs **X** in JP1 NTR in the order they appear in the NTR sequence. Substitutions are shown in boxes. Based on sites 13 and 18, the twelve strings are categorized into two motif families **A** or **B**. The last motif contains an insertion at position 13a and a gap at position 36.

and deletion). We expect the frequencies of any two variants on average to be in the approximate ratio of 1:1 if the relative rate of mutation to duplication is high. On the other hand, if the relative rate of mutation to duplication is low we expect to have a small number of variants.

The variants are hypothesised to be homologous (to have evolved from a single ancestral segment) with an ancestral history of duplication, substitution and deletion. If we knew this history, we could illustrate it by a tree T, rooted at the common ancestor, where each vertex represents a variant (historical or contemporary), and each edge represents a set of edit operations transforming an ancestor to its descendant. We use a parsimony principle for finding a tree T connecting the variants, where the total number of edit operations to transform the connected variants is minimal.

When the variants are closely connected we consider the $1-$cluster graph $G_1 = (V, E)$

motif symbol	variant sequence											NZ1	JP1
a	T	C	G	C	A	C	A	G	C	C	G	38	39
b	T	C	G	C	**C**	C	A	G	C	C	G	18	19
c	T	C	G	C	A	C	**G**	**A**	C	C	G	7	9
d	T	C	G	C	A	C	A	G	C	C	**A**	5	7
e	T	C	G	C	**C**	C	A	**C**	C	C	G	4	3
f	T	**G**	G	C	A	C	**G**	G	C	C	**A**	3	5
g	T	C	G	C	A	C	A	G	**T**	C	G	4	3
h	T	C	G	C	A	C	A	G	**T**	C	**A**	2	2
i	T	C	G	C	**C**	C	A	**T**	C	C	G	3	1
j	T	**T**	G	C	A	C	A	G	C	C	G	1	2
k	T	C	**A**	C	A	C	A	G	C	C	G	1	3
l	T	**T**	G	C	**C**	C	A	G	C	C	G	1	1
m	T	C	G	C	A	C	A	**T**	C	C	G	1	1
n	T	C	**A**	C	A	C	A	G	C	C	**A**	1	1
o	T	C	G	**T**	A	C	A	G	C	C	G	1	1
p	T	C	G	**T**	A	C	**G**	**A**	C	C	G	1	1
q	T	C	G	C	A	C	**G**	G	C	C	G	2	0
r	T	C	G	C	A	C	A	G	C	**T**	G	0	1
s	T	C	G	C	A	C	**C**	G	C	**T**	G	0	1
t	T	C	G	C	A	C	**C**	G	C	C	G	1	0
v	T	C	**C**	**CCC**	A	C	A	G	C	C	G	1	1

Table 3.3: The 21 tandem motif variants of JP1 and NZ1. Each variant is assigned a character $a, \ldots, v$. The frequency of each variant is listed in the third and fourth columns. Variants b to t comprise 11 bp, and differ from a by 1, 2 or 3 substitutions shown in boxes. Variant v comprises 13 bp with CCCCC replacing CGC in the 2, 3, and 4 sites.

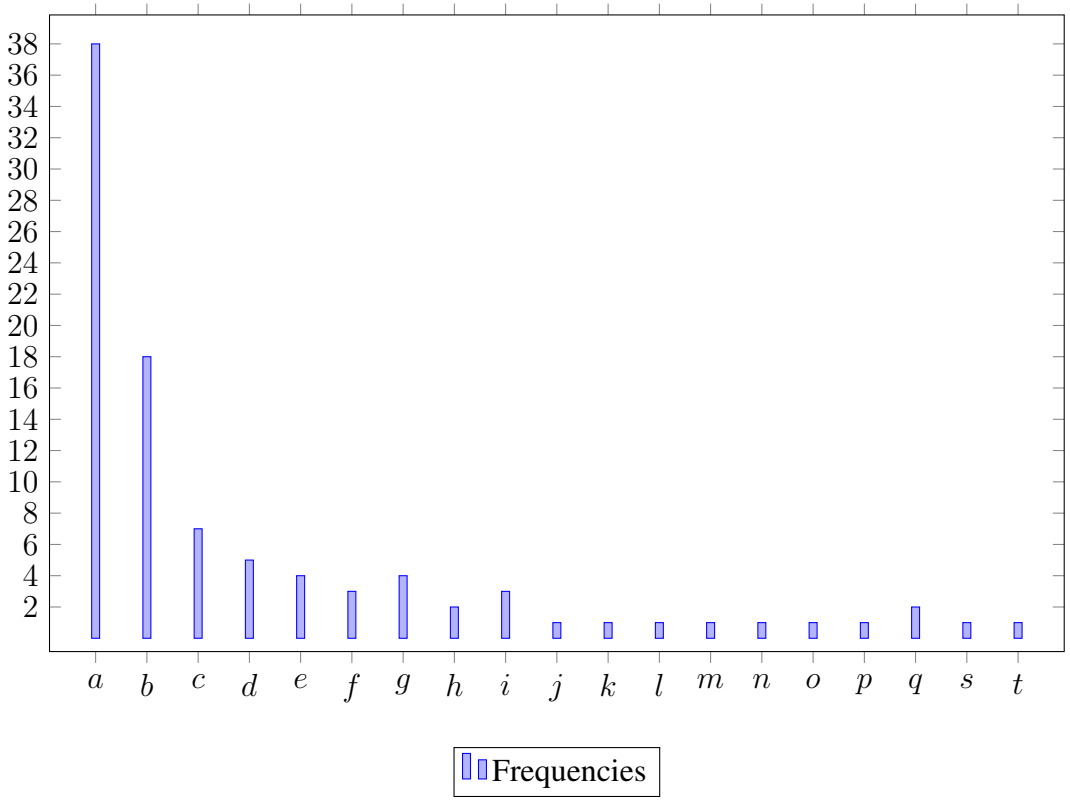


Figure 3.1: Frequencies of all variants in NZ1. The ratios of the three most common variants a, b, and c appear to follow a power law. The frequencies of the variant a, b, and c are in the approximate ratio 4:2:1.

(Hendy et al., 1980), where V is the set of contemporary variants and the set of edges $E = \{(u, v)|u, v \in V; d(u, v) = 1\}$ connects each pair of variants which differ by a single edit operation (single site substitutions and indels). However G_1 will not usually be a tree. The graph G_1 may contain circuits, for example, the four variants a, d, q, z in Figure 3.2. This may be the consequence of a parallel mutation, which can happen when the density of substitutions is high and the segments are short.

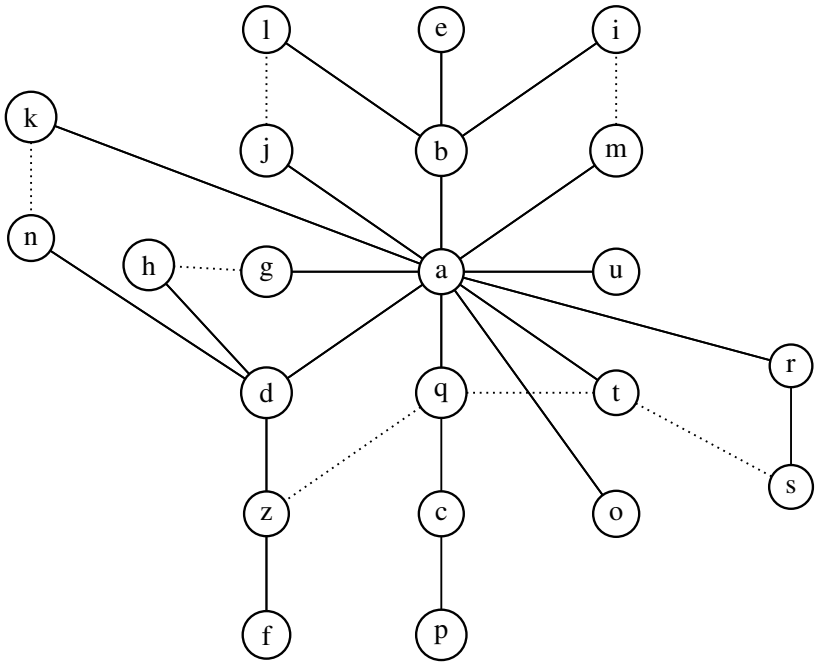


Figure 3.2: The variants graph of the taro variants with the template motif being TCGCACAGCCG, which is the variant *a*. The graph has two components, with variant *f* at distance 2 from its closest neighbours. An additional (unobserved) variant *z* is added to make the graph connected, which is not observed in the NTR region of JP1 and NZ1. Variant *v*=TCCCCCACAGCCG is not included. The set of edges in this graph is the union of the sets of edges of all minimum spanning trees. In each cycle in the graph, the edge that connect the two least frequent variants is drawn as a dashed line.

The set V may not contain all ancestral variants, as some may have been lost by deletion, and hence G_1 might not be connected. This may occur if the substitution rate is higher than the duplication rate. Connectivity can be achieved by adding edges (u, v) with $d(u, v) > 1$, where u, v are in different components of G_1, and *Steiner points* (new vertices which could represent ancestral variants) when adjacent edges share some common edit operations, to reduce the total number of edit operations across the edges. For example, the variant f differs by more than one substitution from any other observed variant, and so we add the unobserved variant z (see Figure 3.2).

We will refer to a connected graph G which connects all the variants, and which may include additional hypothetical variants, and in which every edge represents an edit operation, as a *variants graph*. The construction of the variants graph is NP-hard (Foulds and Graham, 1982).

The *variants distance graph* is the weighted complete graph with vertex set the set of variants, in which the edge between variants u and v is given weight $d(u, v)$.

3.4 Expected number of parallel substitutions

Single point substitutions happen in the duplication history of tandem repeats. These single substitutions produce a set of variants that we observe at the current time. The number of substitutions that occurred in the past is unknown, but a lower bound on the number of substitutions is the length of the Steiner tree of the variants, which can be approximated by the length of the minimum spanning tree of the distance graph. Note that the length of the Steiner tree is equal to the length of the minimum spanning tree when the variants graph is an $1-$cluster (Hendy et al., 1980). However, some parallel substitutions may have occurred during the evolution of the nested tandem repeat, whereby an existing variant is created again.

In this section, we calculate the likelihood $P(k, i)$ that i observed substitutions are the result of $k \geq i$ substitutions on a motif of size n. There are $3n$ possible substitutions, where n is the length of the motif. A substitution can either be a new substitution or it can be parallel (it duplicates an existing substitution); in the second case the number of substitutions does not increase. If we observe i substitutions after k substitutions have occurred, then the $(k+1)-$th substitution produces either a new substitution (with probability $\frac{3n-(i-1)}{3n}$), or reproduces an existing substitution (a parallel substitution with probability $\frac{i}{3n}$). Thus the probability $P(k, i)$ of observing i substitutions after $k > 0$ substitutions can be calculated using the recursive formula

$$P(k,i) = P(k-1, i-1) \times \frac{3n-(i-1)}{3n} + P(k-1,i) \times \frac{i}{3n},$$

$$\text{for } k > 0, i > 0,$$

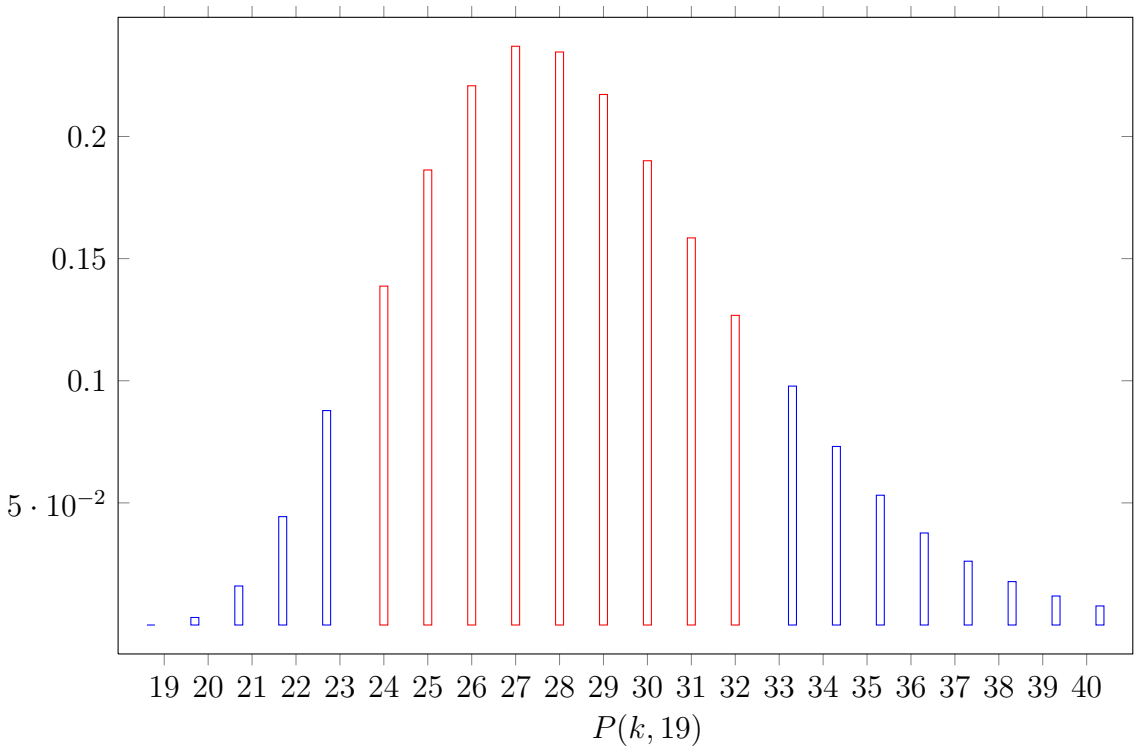


Figure 3.3: The likelihood distribution of the number of substitutions to produce 19 variants of a motif of length 11 bp. Bars representing likelihood values greater than 0.1 are coloured in red.

with initial values

$$P(0,1) = 1, P(0,i) = 0, P(k,0) = 0 \text{ for } i \neq 1.$$

Figure 3.3 plots $P(k,i)$ for $n = 11$ and $i = 19$, as is the case for the tandem motifs in NZ1. This suggests that the number of parallel substitutions in NZ1 is more likely to be in the range $24 - 19 = 5$ to $32 - 19 = 13$.

3.5 Variants frequency distribution

Nested tandem repeat copies contain variants observed in different frequencies. For example, in NZ1, variant 'a' is observed 38 times, while variant 'b' is observed 18 times. The 2:1 distributions in Figure 3.1 suggests an early distribution is 'aab', 'aba' or 'baa'. Ten possible duplication history scenarios that lead to a segment that consists of two 'a's and one 'b' are shown in Figure 3.4.

There are eight minimal paths to generate 'aab', 'aba', or 'baa' from 'a', and two to

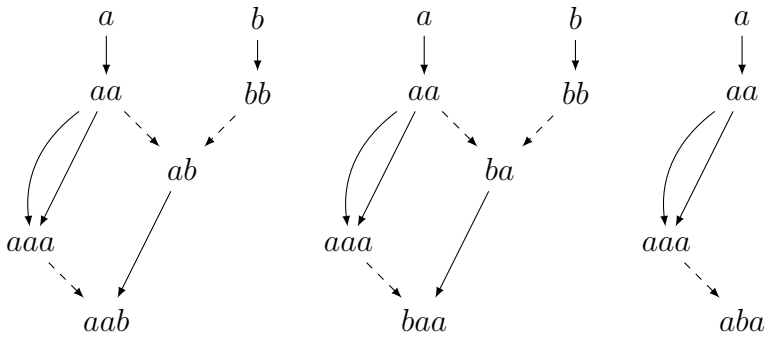


Figure 3.4: Duplication history scenarios. The dashed lines represent substitutions and the solid lines represent duplications. There are 10 possible minimal paths to produce a segment of three characters that consists of two 'a's and one 'b' from a single segment 'a' (8 paths) or 'b' (2 paths). Note that there are two ways to obtain 'aaa' from 'aa', because we may duplicate either the first or the second character.

generate them from 'b'. Therefore, we will choose 'a' as the more likely ancestral variant. We assume that the duplication mechanism has no preference on which variant to copy. Therefore, the ratio of frequencies for the more frequent variants are expected to remain similar through the later stages of the duplication history. Variants occurring with low frequencies and close to each other in the sequence suggest they are recent.

3.6 Variants spread

The distribution of variants in a repeated region is correlated with the time at which the variants were introduced. The earlier variants in the evolution of the NTR have a greater opportunity to be duplicated and therefore are likely to have both a greater spread in the sequence and a greater frequency. The spread here refers to the difference between the right most copy position and the left most copy position for a particular variant. A later variant must have a restricted spread unless there has been one or more parallel substitutions to independently produce additional copies. In Table 3.2 the variant 'a' occurs 38 times in NZ1 and 39 times in JP1 and is spread all over the NTR sequence, which leads to the conclusion that 'a' is the oldest variant and probably the ancestral variant. The frequency of variant 'b' is 18 which suggests it is more likely that the change from 'a' to 'b' happened in an early stage of the duplication history tree.

Figure 3.3 suggests that the total number of substitutions is likely to be in the range 24 to 32, and so the number of parallel substitutions is likely to be in the range $5 = 24 - 19$

to $13 = 32 - 19$. If such a parallel substitution occurs in widely separated variant copies, then these variants will have a larger spread to frequency ratio. In Figure 3.6, we note that variants h, q, g, f have larger spread to frequency ratio. This suggests that it is more likely that these variants copies were a result of parallel substitutions.

It is important to take the information on the spread of the variants into account when inferring the duplication history.

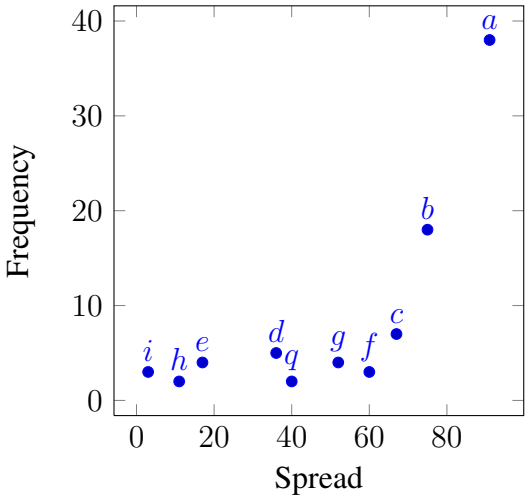


Figure 3.5: Variant frequency plotted against spread. The spread refers to the difference between the positions of the right most and left most copies.

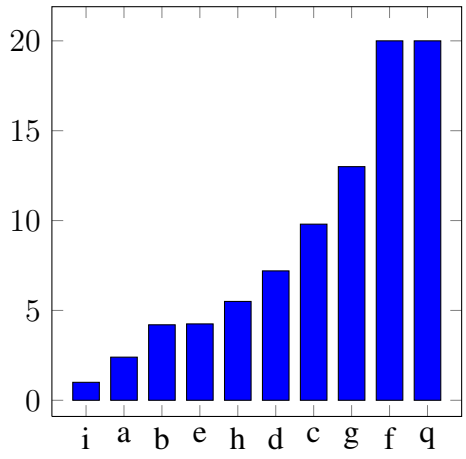


Figure 3.6: Variant spread to frequency ratio. The variants c, g, f, and q have higher spread to frequency ratios, suggesting that they are likely to be the result of parallel substitutions.

Chapter 4

NTRFinder: A Software Tool to Find Nested Tandem Repeats

This chapter reproduces the text of *NTRFinder: a software tool to find nested tandem repeats*, A. Matroud, C. Tuffley, and M. Hendy, Nucleic Acids Research (Matroud et al., 2012b). It has been reformatted for consistency with the rest of thesis, and some background and definitions have been moved to Chapter 1.

4.1 Abstract

We introduce the software tool `NTRFinder` to search for a complex repetitive structure in DNA we call a nested tandem repeat (NTR). An NTR is a recurrence of two or more distinct tandem motifs interspersed with each other. We propose that nested tandem repeats can be used as phylogenetic and population markers.

We have tested our algorithm on both real and simulated data, and present some real nested tandem repeats of interest.

`NTRFinder` can be downloaded from `http://www.maths.otago.ac.nz/~aamatroud/`.

4.2 Introduction

Genomic DNA has long been known to contain *tandem repeats:* repetitive structures in which many approximate copies of a common segment (the *motif*) appear consecu-

tively. Several studies have proposed different mechanisms for the occurrence of tandem repeats (Weitzmann et al., 1997; Wells, 1996), but their biological role is not well understood.

Recently we have observed a more complex repetitive structure in the ribosomal DNA of *Colocasia esculenta* (taro), consisting of multiple approximate copies of two distinct motifs interspersed with one another. We call such structures *nested tandem repeats* (NTRs), and the problem of finding them in sequence data is the focus of this paper. Our motivation is their potential use for studying populations: for example, a preliminary analysis suggests that changes in the NTR in taro have been occurring on a 1,000 year time scale, so a greater understanding of this NTR offers the potential to date the early agriculture of this ancient staple food crop.

The problem of locating tandem repeats is well known, as their implication for neurological disorders (Verkerk et al., 1993; Fu et al., 1992), and their use to infer evolutionary histories has urged some researchers to develop tools to find them. This has resulted in a number of software tools, each of which has its own strengths and limitations. Most of these software tools use statistical criteria on the distances between k-tuple matches (these distances are generally multiples $n \cdot \ell$ of the pattern length ℓ , where $n \in \{1, 2, \dots\}$). However, the distances between matching k-tuples in NTRs are of the form $a \cdot \alpha + b \cdot \beta$, where α and β are the lengths of the two patterns and $a, b \in \{1, 2, \dots\}$. Consequently these software tools do not generally find NTRs. In this paper we present a new software tool, `NTRFinder`, which is designed to find these more complex repetitive structures.

We report here the algorithm on which `NTRFinder` is based and report some of the NTRs it has identified, including an even more complex structure where copies of four distinct motifs are interspersed.

4.3 Material and Methods

In this section we present the algorithm we have developed to search for nested tandem repeats in a DNA sequence. The algorithm requires several preset parameters. These are: k_1 and k_2 which bound the edit distances from the tandem and interspersed motifs; and the motif length bounds $\min_{t_1}, \max_{t_1}, \min_{t_2}, \max_{t_2}$. Other input parameters are discussed below.

Search phase Our search is confined to seeking NTRs with motifs of length $l_1 \in [\min_{t_1}, \max_{t_1}]$ and $l_2 \in [\min_{t_2}, \max_{t_2}]$. A $(k_1, k_2)-$NTR must contain a k_1-TR, so we begin by scanning the sequence for approximate tandem repeats. To do this we have chosen to adapt the tandem repeat search algorithm *ATRHunter* of Wexler *et al.*, in which the sequence is searched for tandem motifs of length l_1 by scanning the sequence with two windows w_1 and w_2 of width w, at distance l_1 apart. This may be adapted to find non-adjacent copies of the tandem motif (as occur in NTRs) by holding w_1 fixed, and moving w_2 further away.

The user may set the $k_1,\ k_2$ values, preset with default values

$$k_1 = l_1(1 - p_m) + \sqrt{l_1(1 - p_m)p_m} \tag{4.1}$$

$$k_2 = l_2(1 - p_m) + \sqrt{l_2(1 - p_m)p_m}, \tag{4.2}$$

following Domaniç and Preparata (2007), with matching probability p_m given the default value $p_m = 0.8$.

Once a TR has been found and its full extent determined, the right-most copy of the repeated pattern is taken as the current TR motif $\mathbf{x}$, and further approximate copies of $\mathbf{x}$ are sought, displaced from the TR up to a distance of $\max_{t_2}$ nucleotides to the right. This is done by moving the second scanning window w_2 to the right, while holding the first fixed in the current copy of $\mathbf{x}$. If no further approximate copies of $\mathbf{x}$ are located, this TR is abandoned, and the TR search continues to the right. If a displaced approximate copy of $\mathbf{x}$ is observed, then both $\mathbf{x}$ and the interspersed segment $\mathbf{X}$ are recorded in a list, as we have found a candidate NTR. Further contiguous copies of $\mathbf{x}$ are then sought, with the rightmost copy $\mathbf{x}$ replacing the previous template motif.

The steps above are repeated with successive motifs $\mathbf{x}$ and interspersed segments copied to the list, until no additional copies of the last recorded motif $\mathbf{x}$ are found. This search phase is illustrated in Figure 4.1.

At this point the algorithm builds consensus patterns for $\mathbf{x}$ and $\mathbf{X}$ using majority rule. After constructing the two consensus patterns the algorithm moves to the verification phase.

Example: An example will help illustrate the procedure. Suppose that S contains an NTR of the form

$$\mathbf{xX_0xxxX_1xxxxxxX_2xxX_3}.$$

The algorithm will scan from the left until it locates the tandem repeat consisting of three copies of $\mathbf{x}$ between $\mathbf{X}_0$ and $\mathbf{X}_1$. It will then start searching for additional non-adjacent copies of $\mathbf{x}$ to the right, locating the first copy to the right of $\mathbf{X}_1$. Having found this it will record the intervening segment $\mathbf{X}_1$, and then continue the tandem repeat search from this point until the full extent of the tandem repeat between $\mathbf{X}_1$ and $\mathbf{X}_2$ is found.

This procedure is repeated once more, locating the tandem repeat between $\mathbf{X}_2$ and $\mathbf{X}_3$, recording the segment $\mathbf{X}_2$, and then searching for further copies to the right. At this point no more copies of $\mathbf{x}$ are found, and the process of verification begins. The segments $\mathbf{X}_0$, $\mathbf{X}_3$ and the initial copy of $\mathbf{x}$ are found during this stage.

Verification phase: Each candidate NTR is checked to determine whether it meets the NTR definition. This is accomplished by aligning the candidate NTR region, together with a margin on either side of it, against the consensus motifs $\mathbf{x}$ and $\mathbf{X}$, using the nested wrap-around dynamic programming algorithm of Matroud et al. (2011), presented in Chapter 5 of this thesis. The nested wrap-around dynamic programming parameters are set to be 2 for a match, -5 for a mismatch, and -7 for a gap. These parameters were chosen following (Wexler et al., 2005). The nested wrap-around dynamic programming algorithm has complexity $O(n|\mathbf{x}||\mathbf{X}|)$, where n is the length of the NTR region and $|\mathbf{x}|$ and $|\mathbf{X}|$ are the length of the tandem motif and the length of the interspersed motif respectively.

A remark on tandem repeat detection, and the role of verification The definition of a k-TR requires that each repeat be a distance at most k from some template motif. However, this template is unknown during the search phase. We follow ATRHunter's algorithm to compare each repeat copy with its preceding copy. Comparisons between adjacent copies will not miss any tandem repeats, provided the distance threshold is set appropriately, but may result in false positives due to "drift". Such false positives are eliminated during the verification phase, when the candidate tandem repeat is aligned against the consensus motif.

Suppose that $\mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_n$ is a k-TR with motif $\mathbf{x}$. Then since $d(\mathbf{x},\mathbf{x}_i) \leq k$ we have

$$d(\mathbf{x}_i,\mathbf{x}_j) \leq d(\mathbf{x}_i,\mathbf{x}) + d(\mathbf{x},\mathbf{x}_j) \leq 2k,$$

by the triangle inequality. It follows that a tandem repeat search that correctly detects when $d(\mathbf{x}_i,\mathbf{x}_{i+1}) \leq d$ will find all $(d/2)$-TRs.

We note however that a segment $\mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_n$ satisfying $d(\mathbf{x}_i,\mathbf{x}_{i+1}) \leq d$ for all i need not be a tandem repeat, since $\mathbf{x}_j$ may "drift" away from $\mathbf{x}_i$ as j increases. A simple example is

`aaaa aaac aacc accc cccc`,

in which adjacent copies are distance 1 apart, but the first and last copies are distance 4 apart.

4.4 Results

4.4.1 Tests on simulated data

In order to measure the accuracy of `NTRFinder`, we generated synthetic sequence data containing NTR subsequences with varying probabilities of substitution and insertion and deletion (indels), and determined the proportion of the NTRs that were found by `NTRFinder`. In our simulation we first generated one random DNA sequence of 100000 nucleotides, with each nucleotide occurring with probability 0.25. Within this sequence we embedded 100 exact NTRs with repeats of randomly generated motifs **X** and **x** of varying lengths. From this sequence we generated four additional sequences by introducing indels and substitutions. Indels were introduced to each sequence with a constant probability of 1% per site, and substitutions were introduced with varying probabilities of 1%, 2%, 3% and 4% per site. `NTRFinder` recovered 95%, 84%, 83%, 83% and 80% of the NTRs respectively. These results are plotted in Figure 4.2. No false positives were detected.

The first phase of `NTRFinder` uses a modification of the algorithm ATRHunter presented by Wexler et al. (2005). Wexler et al. report that ATRHunter has a 74%–90% success rate for finding ATRs in synthetic sequences, with average score of an ATR over all sequences being 238 with a standard deviation 116. These results suggest the accuracy

of the Wexler algorithm provides the major limitation on the accuracy of `NTRFinder`.

4.4.2 Tests on real sequence data

To test NTRFinder on real sequence data we searched all intergenic spacer (IGS) sequences available in Genbank. The IGS sequences were chosen because we already knew of an NTR in the IGS region of *C. esculenta*. We also searched the entire Human Y chromosome from (Fujita et al., 2010).

The size ranges used for this search were $[\min_{t_1}, \max_{t_1}] = [\min_{t_2}, \max_{t_2}] = [2, 100]$, with the parameters k_1 and k_2 set to their default values given in equations (4.1) and (4.2) on page 29. NTRs found in IGS sequences are listed in Table 4.1. We searched 27 IGS sequences and found NTRs in 12 of them.

NTRs found in the Human Y chromosome are listed in Table 4.2. The 11 NTRs found in the Y chromosome all appear to be in the psuedoautosomal region.

4.4.3 More complex structures

In addition to the nested tandem repeats in Table 4.2, `NTRFinder` also reported an NTR in *Linum usitatissimum* (accession number gi$|$ 164684852 $|$ gb|EU307117.1|) which on further analysis by hand turned out to have a more complex structure. The IGS region of the rDNA of this species contains an NTR with four motifs interspersed with each other. The four motifs are **w**=`GTGCGAAAAT`, **x**=`GCGCGCCAGGG`, **y**=`GCACCCATAT`, and **z**=`GCGATTTTG`, and the structure of the NTR has the form

$$\prod_{i=1}^{25} \mathbf{w}^{q_i}\mathbf{x}^{r_i}\mathbf{z}^{s_i}\mathbf{y}^{t_i},$$

where $q_i \in \{1, 2, 3\}$; $r_i \in \{1, 2\}$; $s_i \in \{0, 1\}$; $t_i \in \{0, 1\}$.

4.4.4 Running time

The running time for `NTRFinder` searching some sequences from GenBank is shown in Figure 4.3. It can be seen that the run time is approximately linear in the length of the sequence. However, it must be noted that the run time depends not only on the length of the input sequence, but also on the number of tandem and nested tandem repeats found in

the sequence. The program spends most of the time verifying any tandem repeats found.

4.5 Discussion

In the last decade a number of software tools to find tandem repeats have been introduced; however, little work exists on more complex repetitive structures such as nested tandem repeats. The problem of finding nested tandem repeats is addressed in this study. The motivation for our study is the potential use of NTRs as a marker for genetic studies of populations and of species.

We have done some analysis on the nested tandem repeat in the intergenic spacer region in *C. esculenta* (taro), noting some variation in the NTRs derived from domesticated varieties sourced from New Zealand, Australia and Japan. Further varieties are currently being analysed.

4.6 Conclusion

The nested tandem repeat structure is a complex structure that requires further analysis and study. The number of copy variants in the NTR region and the relationships between these copies might suggest a tandem repeat generation mechanism. In this paper, we have introduced a new algorithm to find nested tandem repeats. The first phase of the algorithm has $O(n(\max_{t_1})(\max_{t_2}))$ time complexity, while the second phase (the alignment) needs $O(n(\max_{t_1})(\max_{t_2}))$ space and time, where n is the length of the NTR region, and $\max_{t_1}, \max_{t_2}$ are the maximum allowed lengths of the tandem and interspersed motifs.

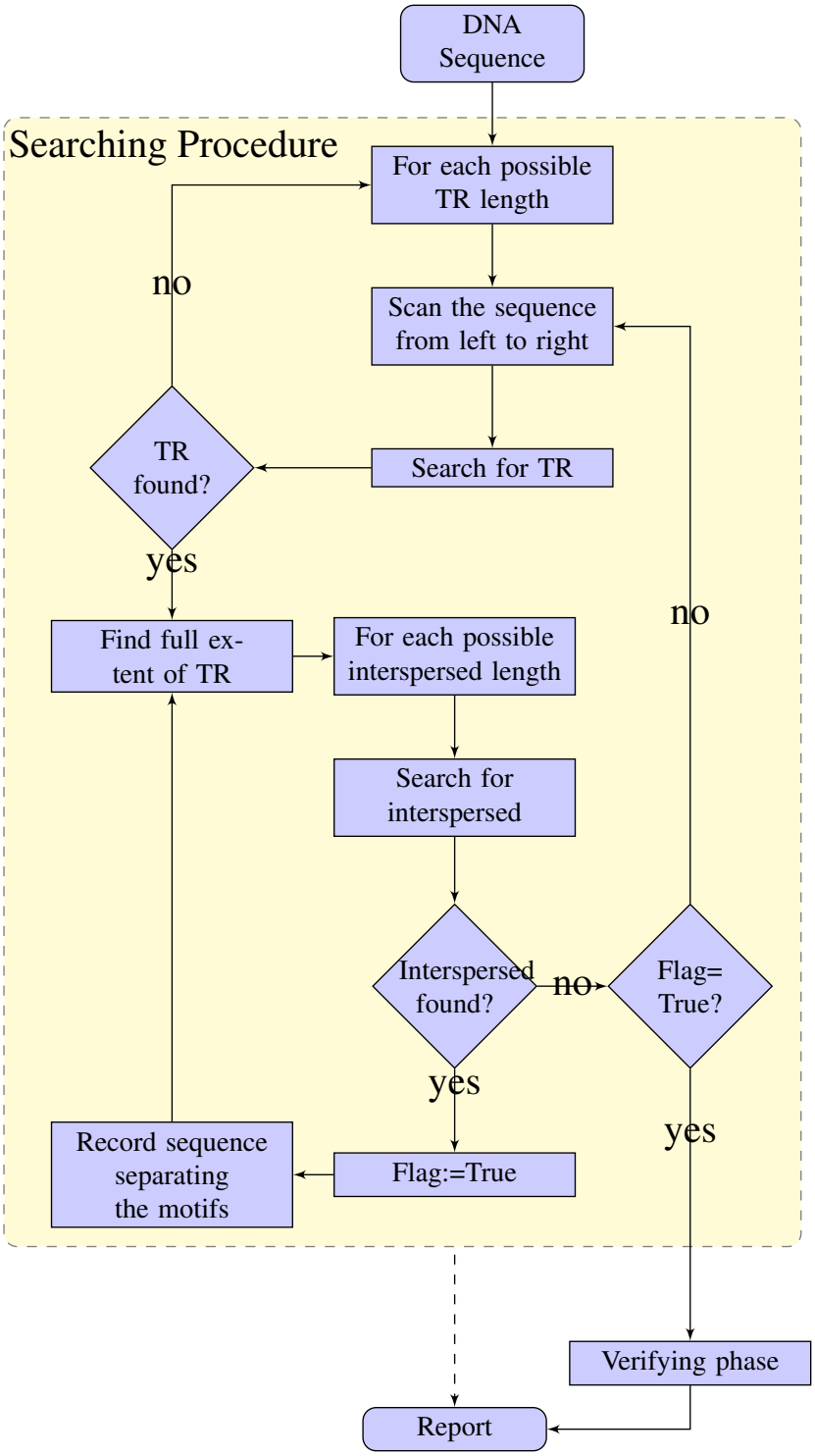


Figure 4.1: Flowchart of the `NTRFinder` algorithm.

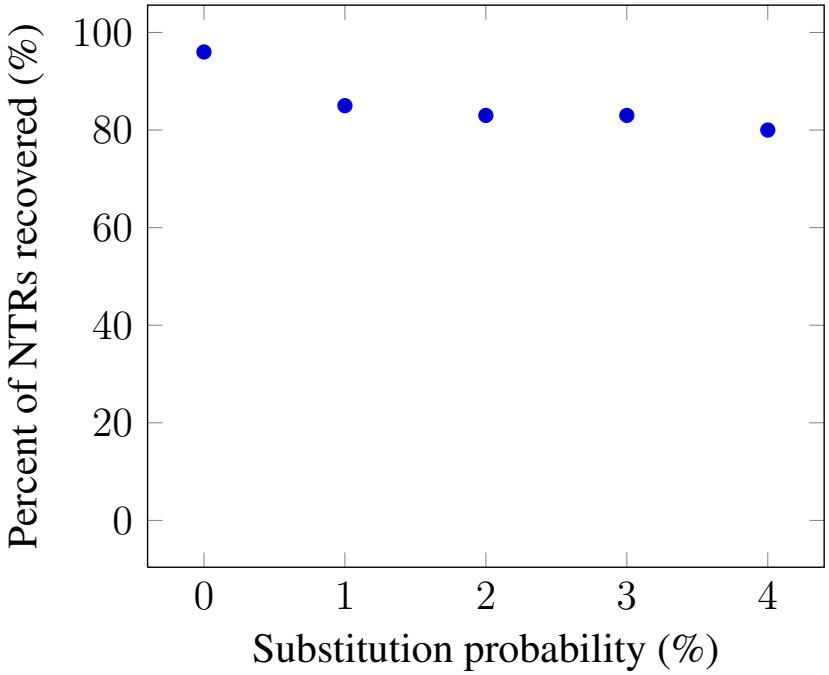


Figure 4.2: Percentage of NTRs found in the synthetic sequences.

Species Accession number	NTR structure $(s_0, s_1, ..., s_n)$	|x| |X|	start index end index
Nicotiana sylvestris X76056.1	(0,1,2,3,4,2,3,2,3,3,3,5,4,3,2,3,3,1,2,5,3,5,4,2,2,4,3)	10 13	960 2,111
Brassica juncea X73032.1	(9,12,2,6,2,5,4,1)	21 30	1,403 2,605
Brassica olerecea X56978.1	(1,1,1,3,3,1,3,2,1,1,2,1,3,1,1,2,1,2,1,1,2)	30 44	1,031 2,902
Brassica olerecea X60324.1	(1,1,1,3,3,1,3,1,3,2,1,1,2,1,3,1,1,2,1,2,1,1,2)	30 44	1,036 3,133
Brassica rapa S78172.1	(1,2,2,3,3,2,2,2,3)	12 45	385 1,337
Brassica campestris X73031.1	(6,4,4,7,4,4,4,3,1)	21 51	1,558 2,580
Colocasia esculenta Not published	(5,3,1,6,10,5,10,9,13,14,15,4)	11 48	725 2,384
Nicotiana tomentosiformis Y08427.1	(1,1,2,2,1,2,4,2,1)	20 46	1,016 1,969
Arabidopsis thaliana CP002685.1	(3,2,1,1)	13 17	32,189 32,365
Zea mays AJ309824.2	(2,2,1)	19 52	2,984 3,113
Olea europaea AJ865373.1	(3,1,3,6,5,6,4,3,4)	75 11	961 3,743
Herdmania momus X53538.1	(3,1,1,1,1,0)	107 91	6,363 7,642

Table 4.1: Nested tandem repeats found in some IGS sequences searched from GenBank and an additional unpublished sequence (*C. esculenta*).

NTR structure $(s_0, s_1, ..., s_n)$	$\|\mathbf{x}\|$ $\|\mathbf{X}\|$	start index end index
(1,2,2,1,2,1,2,1,1,2,2,2,2,1)	12 56	143,865 144,880
(7,22,23,12,14,4)	2 88	234,183 234,767
(1,1,2,1,1,1,1,,1,1,1,1,1,1,1,1,1,1,1,1)	15 14	465,369 466,397
(1,1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,1,1,1,1)	11 16	647,659 649,721
(17,15,31,28,72,62)	2 49	901,237 902,037
(3,6,8,11,7,6,4,4,5,4,11)	12 32	1,279,754 1,280,875
(26,27,25,25,25,20,17,13,26)	1 48	1,397,128 1,397,735
(1,2,1,2,1,2,2,2,2,2,1,2,1,1,1,2,2,2,1,2,2,1,1)	16 22	1,516,157 1,517,560
(1,1,2,6,2,2,2,1,2)	19 35	1,626,578 1,627,258
(1,1,1,0,2,1)	19 56	2,102,194 2,102,594
(2,2,2,1,2,1,1,1,1,2,6)	21 15	2,164,541 2,165,091

Table 4.2: Nested tandem repeats found in the Human Y chromosome (accession number NC_000024).

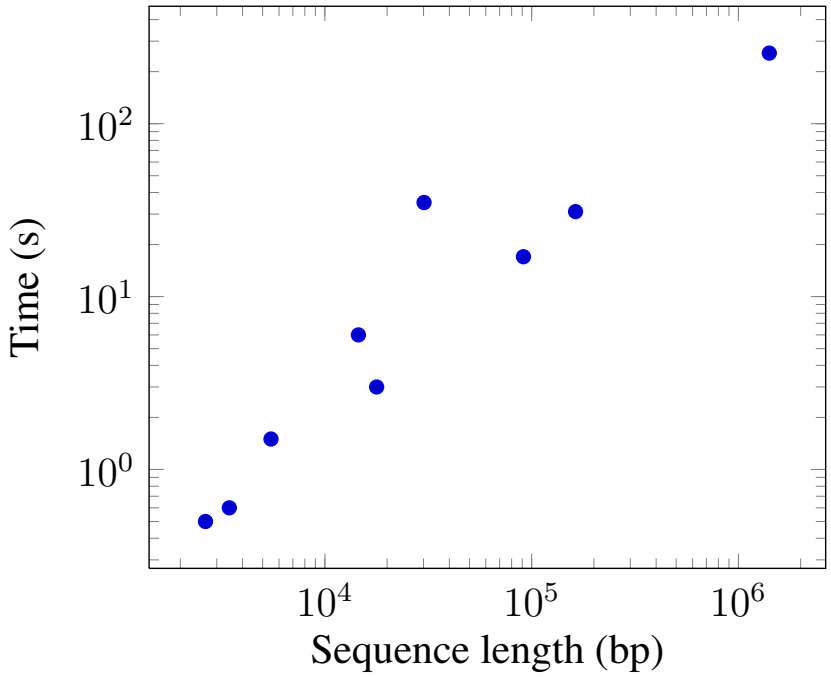


Figure 4.3: Running time of NTRFinder (on a Pentium Dual core T4300 2.1 GHz) plotted against segment length on a log-log scale. The search was performed on segments of different lengths, with the minimum and maximum tandem repeat lengths set to 8 and 50 respectively. The distribution suggests the running time is approximately linear with sequence length.

Chapter 5

An algorithm to solve the motif alignment problem for approximate nested tandem repeats in biological sequences

This chapter reproduces the text of *An algorithm to solve the motif alignment problem for approximate nested tandem repeats in biological sequences*, A. Matroud, C. Tuffley, and M. Hendy, Journal of Computational Biology (Matroud et al., 2011). It has been reformatted for consistency with the rest of thesis.

5.1 Abstract

An *approximate nested tandem repeat* (NTR) in a string $\mathbf{T}$ is a complex repetitive structure consisting of many approximate copies of two substrings $\mathbf{x}$ and $\mathbf{X}$ ("motifs") interspersed with one another. NTRs have been found in real DNA sequences and are expected to be important in evolutionary biology, both in understanding evolution of the ribosomal DNA (where NTRs can occur), and as a potential marker in population genetic and phylogenetic studies. This chapter describes an alignment algorithm for the verification phase of the software tool NTRFinder developed for database searches for NTRs. When the search algorithm has located a subsequence containing a possible NTR, with motifs $\mathbf{X}$ and $\mathbf{x}$, a verification step aligns this subsequence against an exact NTR built from the

templates $\mathbf{X}$ and $\mathbf{x}$, to determine whether the subsequence contains an approximate NTR and its extent. This chapter describes an algorithm to solve this alignment problem in $O(|\mathbf{T}|(|\mathbf{X}| + |\mathbf{x}|))$ space and time. The algorithm is based on the wrap-around dynamic programming of Fischetti et al. (1993).

5.2 Introduction

An *approximate nested tandem repeat* (NTR) in a string $\mathbf{T}$ is a complex repetitive structure consisting of many approximate copies of two substrings $\mathbf{x}$ and $\mathbf{X}$ ("motifs") interspersed with one another. The name derives from the fact that an NTR may be thought of as two tandem repeats nested within one another.

Approximate nested tandem repeats have been found in real DNA sequences, such as that of *Colocasia esculenta*, the ancient staple food crop taro (Matroud et al., 2012b). The intergenic spacer (IGS) region in the taro ribosomal DNA contains an NTR consisting of eleven approximate copies of a 48 bp motif, interspersed within a tandem repeat consisting of 96 approximate copies of an 11 bp motif. The NTR found in taro, used as a genetic marker, offers the potential to elucidate the prehistory of the early agriculture of this ancient food crop, as mutation events appear to be accumulating on a thousand-year time scale. NTRs in general also offer an opportunity to investigate concerted evolution whereby mutations are propagated throughout the many hundreds of copies of the IGS region in the taro genome.

To develop a fuller understanding of the nature of NTRs, we have developed software to find them (Matroud et al., 2012b). This comprises two phases. The first phase is the detection phase, where the sequence is scanned to locate candidate NTRs and construct their consensus motifs $\mathbf{X}$ and $\mathbf{x}$. The second phase is the verification phase where a subsequence containing a possible NTR, with motifs $\mathbf{X}$ and $\mathbf{x}$, is aligned against all patterns of the form

$$\mathbf{x}^{s_0}\mathbf{X}^{t_0}\mathbf{x}^{s_1}\mathbf{X}^{t_1}\cdots\mathbf{x}^{s_k}\mathbf{X}^{t_k}.$$

Such an alignment is needed to find the extent and structure of the NTR (that is, to find the exponents s_i, t_i occurring above), and may also be used to evaluate the fit of the template motifs $\mathbf{x}$ and $\mathbf{X}$. We call this problem the *motif alignment problem* for NTRs, to distinguish it from the mapping problem (*variants alignment problem*) that arises at later

stages of the analysis.

The purpose of this chapter is to present an algorithm to solve the motif alignment problem for approximate NTRs, given a sequence $\mathbf{T}$, and the motifs $\mathbf{x}$ and $\mathbf{X}$ identified by our NTR search algorithm `NTRFinder` (Matroud et al., 2012b). Our alignment algorithm runs in $O(|\mathbf{T}|(|\mathbf{x}| + |\mathbf{X}|))$ space and time, and plays a key rule in the verification phase of NTRFinder. It is based on the wrap-around dynamic programming technique introduced by (Fischetti et al., 1993) to solve the corresponding problem for (ordinary) tandem repeats. We show it can be readily adapted for use with more complex repetitive structures built from three or more motifs.

5.3 Definitions

5.3.1 Alphabets and strings

An *alphabet* is a nonempty set Σ of symbols or *characters*, and a *string* over Σ is a finite sequence of elements of Σ. We write Σ^* for the set of all strings over the alphabet Σ, and $|\mathbf{S}|$ for the length of the string $\mathbf{S}$.

Given a string $\mathbf{S}$ and integers i, j such that $0 < i \leq j \leq |\mathbf{S}|$, we will write $\mathbf{S}[i]$ for the ith character of $\mathbf{S}$, and $\mathbf{S}[i, j]$ for the substring consisting of the ith to jth characters of $\mathbf{S}$. Given a second string $\mathbf{T}$, the *concatenation* of $\mathbf{S}$ and $\mathbf{T}$ is the string $\mathbf{ST}$, where

$$(\mathbf{ST})[i] = \begin{cases} \mathbf{S}[i] & \text{if } i \leq |\mathbf{S}|, \\ \mathbf{T}[i - |\mathbf{S}|] & \text{if } i > |\mathbf{S}|. \end{cases}$$

In applications to DNA sequences Σ is typically the set $\{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$, and we will use this alphabet in examples. However, our algorithm is not restricted to this case.

5.3.2 The edit distance

In order to compare two strings $\mathbf{X}$ and $\mathbf{Y}$ it is useful to have some measure of the extent to which they differ. For the purposes of this chapter we will use the *edit distance*, where the edit operations we permit are the insertion of a single character; the substitution of a single character; or the deletion of a single character.

Given a set of allowed edit operations, such as those listed above, the edit distance from $\mathbf{X}$ to $\mathbf{Y}$, $d(\mathbf{X}, \mathbf{Y})$, is the minimum number of allowable edit operations needed to transform $\mathbf{X}$ into $\mathbf{Y}$. With the choice of permitted edit operations made above, it is straight forward to verify that d is a metric.

5.3.3 Tandem repeats and nested tandem repeats

An *exact tandem repeat* is a string of the form $\mathbf{X}^l$ for some $l \geq 2$. Thus, an exact tandem repeat is a string comprised of two or more contiguous exact copies of the same substring $\mathbf{X}$. This substring is called the *motif* of the tandem repeat. We obtain an *approximate tandem repeat* by allowing approximate rather than exact copies of the template motif $\mathbf{X}$. More precisely, an approximate tandem repeat is a string of the form $\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_l$, where $d(\mathbf{X}, \mathbf{X}_i) \leq k|\mathbf{X}|$ for each i, for some fixed $k < 1$ and template motif $\mathbf{X}$. Where the value of the parameter k is important we may say that we have a *k-approximate tandem repeat* (k-TR). For simplicity of notation, we will write $\tilde{\mathbf{X}}^l$ to mean an approximate tandem repeat, consisting of l approximate copies of $\mathbf{X}$.

Given two motifs $\mathbf{X}$ and $\mathbf{x}$ such that $d(\mathbf{X}, \mathbf{x}) \gg 0$, an *exact nested tandem repeat* is a string of the form

$$\mathbf{x}^{s_0}\mathbf{X}^{t_0}\mathbf{x}^{s_1}\mathbf{X}^{t_1}\cdots\mathbf{x}^{s_n}\mathbf{X}^{t_n},$$

where $n > 1$, $s_i \geq 1$ for each $i > 0$, and $t_i \geq 1$ for each $i < n$. We again obtain an *approximate nested tandem repeat* by allowing the copies of the motifs $\mathbf{X}$ and $\mathbf{x}$ to be approximate rather than exact. Thus, an approximate nested tandem repeat is a string of the form

$$\tilde{\mathbf{x}}^{s_0}\tilde{\mathbf{X}}^{t_0}\tilde{\mathbf{x}}^{s_1}\tilde{\mathbf{X}}^{t_1}\cdots\tilde{\mathbf{x}}^{s_n}\tilde{\mathbf{X}}^{t_n},$$

where $n > 1$, $s_i \geq 1$ for each $i > 0$, and $t_i \geq 1$ for each $i < n$, and such that $\tilde{\mathbf{x}}^{s_0}\tilde{\mathbf{x}}^{s_1}\cdots\tilde{\mathbf{x}}^{s_n}$ is an approximate tandem repeat with motif $\mathbf{x}$, and $\tilde{\mathbf{X}}^{t_0}\tilde{\mathbf{X}}^{t_1}\cdots\tilde{\mathbf{X}}^{t_n}$ is an approximate tandem repeat with motif $\mathbf{X}$.

Note that the definition of an approximate nested tandem repeat includes exact nested tandem repeats as a special case. "Nested tandem repeat" or "NTR" by itself will always mean an *approximate* nested tandem repeat, unless explicitly stated otherwise.

Remark. The definition of an NTR given here is slightly more general than that given in Chapter 4. In Chapter 4, a nested tandem repeat is required to satisfy $t_i \leq 1$ for each i.

5.3.4 Alignment

Given an alphabet Σ, let $\bar{\Sigma}$ be the alphabet $\Sigma \cup \{-\}$, where "$-$" ("*gap*") is a character that does not belong to Σ. We define $\phi : \bar{\Sigma}^* \to \Sigma^*$ to be the function that deletes all gaps.

Given two strings $\mathbf{A}, \mathbf{B} \in \Sigma^*$, an *alignment* of $\mathbf{A}$ and $\mathbf{B}$ is a choice of a pair of strings $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) \in \bar{\Sigma}^* \times \bar{\Sigma}^*$ satisfying the following conditions:

A1. $\phi(\bar{\mathbf{A}}) = \mathbf{A}$ and $\phi(\bar{\mathbf{B}}) = \mathbf{B}$;

A2. $|\bar{\mathbf{A}}| = |\bar{\mathbf{B}}|$; and

A3. there is no index i for which $\bar{\mathbf{A}}[i] = \bar{\mathbf{B}}[i] = -$.

Thus, $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are obtained from $\mathbf{A}$ and $\mathbf{B}$ respectively by inserting gaps in such a way that the resulting strings have the same length, and do not both have a gap in the same position.

σ	$-$	A	C	G	T
$-$	$-\infty$	-2	-2	-2	-2
A	-2	1	-1	-1	-1
C	-2	-1	1	-1	-1
G	-2	-1	-1	1	-1
T	-2	-1	-1	-1	1

Table 5.1: A sample scoring matrix for DNA sequences. This matrix rewards matching characters from Σ with a score of $+1$, and penalises mis-matching characters from Σ with a score of -1. The penalty for aligning a gap against a character from Σ is -2. The value $\sigma(-,-) = -\infty$ reflects condition A3, which prohibits a gap being aligned against a gap.

To score an alignment we use a scoring matrix σ, which specifies the reward or penalty for aligning any two characters of $\bar{\Sigma}$ against each other. See Table 5.1 for an example. We will assume throughout that σ penalises gaps (that is, $\sigma(-, \alpha)$ and $\sigma(\alpha, -)$ are both negative for all $\alpha \in \bar{\Sigma}$), and we set $\sigma(-,-) = -\infty$ to reflect condition A3 above. Given an alignment $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ for which $|\bar{\mathbf{A}}| = |\bar{\mathbf{B}}| = L$, the *alignment score* of $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ is then defined to be

$$\sigma(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \sum_{i=1}^{L} \sigma(\bar{\mathbf{A}}[i], \bar{\mathbf{B}}[i]).$$

An *optimal global alignment* is an alignment of $\mathbf{A}$ and $\mathbf{B}$ which maximises the alignment score over all such alignments. See (Navarro, 1999) for a survey of this and other alignment problems.

5.4 The motif alignment problem for approximate nested tandem repeats

5.4.1 The problem

The *motif alignment problem for approximate nested tandem repeats* is the following:

Given

1. a string $\mathbf{T}$ and motifs $\mathbf{x}$ and $\mathbf{X}$ over the alphabet Σ, and
2. a scoring matrix σ defined over $\bar{\Sigma} \times \bar{\Sigma}$,

find an optimal alignment of $\mathbf{T}$ against substrings of strings of the form

$$\mathbf{x}^{s_0}\mathbf{X}^{t_0}\mathbf{x}^{s_1}\mathbf{X}^{t_1}\cdots\mathbf{x}^{s_k}\mathbf{X}^{t_k}.$$

Thus, given a string $\mathbf{T}$ that is presumed to contain an approximate nested tandem repeat with motifs $\mathbf{x}$ and $\mathbf{X}$, and a scoring matrix σ, the problem is to find an optimal alignment of $\mathbf{T}$ against substrings of *exact* nested tandem repeats with motifs $\mathbf{x}$ and $\mathbf{X}$.

5.4.2 Solution to the problem via nested wrap-around dynamic programming

The motif alignment problem for NTRs is closely related to the corresponding problem for tandem repeats, which was solved by (Fischetti et al., 1993) using wrap-around dynamic programming. We solve the problem by adapting their technique. The key differences are the introduction of a second matrix, to hold information relating to the second motif, and a modification to the update rule used between the first and second passes.

In what follows we let $n = |\mathbf{T}|$, $m = |\mathbf{x}|$, and $l = |\mathbf{X}|$. An optimal alignment will be calculated using two matrices $D^{(1)}$ and $D^{(2)}$. The matrix $D^{(1)}$ is $(m+1) \times (n+1)$, and will record scores related to aligning portions of $\mathbf{T}$ against $\mathbf{x}$, while the matrix $D^{(2)}$ is $(l+1) \times (n+1)$, and will record scores related to aligning portions of $\mathbf{T}$ against $\mathbf{X}$. Both matrices will be indexed starting from 0, and we will denote the (i,j) entry of $D^{(k)}$ by $D^{(k)}[i,j]$. We write $D^{(k)}_{i,j}$ for the upper-left $(i+1) \times (j+1)$ submatrix of $D^{(k)}$.

The score matrices $D^{(1)}$ and $D^{(2)}$ are filled as follows:

1. We initialise the two matrices by setting

$$D^{(k)}[0, j] := 0, \qquad D^{(k)}[i, 0] := 0$$

for all i, j and k.

2. We compute each column of the matrices (starting from $j = 1$) in two rounds. In the first round we compute $D^{(1)}[i, j]$ using the recursive function

$$D^{(1)}[i, j] := \max \left\{ \begin{array}{c} D^{(1)}[i-1, j-1] + \sigma(\mathbf{x}[i], \mathbf{T}[j]), \\ D^{(1)}[i-1, j] + \sigma(\mathbf{x}[i], -), \\ D^{(1)}[i, j-1] + \sigma(-, \mathbf{T}[j]) \end{array} \right\}.$$

We then compute $D^{(2)}[i, j]$ in a similar fashion.

In the second round, we update both matrix entries $D^{(1)}[0, j]$ and $D^{(2)}[0, j]$ with the value $\max\{D^{(1)}[m, j],\ D^{(2)}[l, j]\}$, and then update $D^{(1)}[i, j]$ for $1 \leq i \leq m$ using the formula above, which simplifies to

$$D^{(1)}[i, j] := \max\{D^{(1)}[i, j], D^{(1)}[i-1, j] + \sigma(\mathbf{x}[i], -)\}$$

during the second round. The entries $D^{(2)}[i, j]$ for $1 \leq i \leq l$ are then updated in a similar fashion.

The visualisation of the algorithm is shown in Figure 5.4.2. Pseudo-code for the matrix-filling algorithm appears below.

Once the matrices are filled, an optimal alignment is found using the standard trace-back procedure for dynamic programming (see for example (Fischetti et al., 1993)), beginning from the largest entry in the righthand columns of $D^{(1)}$, $D^{(2)}$. The algorithm clearly has space complexity $O(n(m + l))$, and the matrices $D^{(1)}$ and $D^{(2)}$ are filled in time $O(n(m + l))$.

Data: Strings $\mathbf{T}$, $\mathbf{X}$, $\mathbf{x}$ and scoring matrix σ
Result: Matrices $D^{(1)}$, $D^{(2)}$ containing optimal alignment scores with respect to σ of alignments of $\mathbf{T}$ against substrings of exact NTRs with motifs $\mathbf{X}$ and $\mathbf{x}$

for $j = 0$ **to** $|\mathbf{T}|$ **do**
  for $i = 0$ **to** $|\mathbf{x}|$ **do**
    $D^{(1)}[i, j] := 0$
  end
  for $i = 1$ **to** $|\mathbf{X}|$ **do**
    $D^{(2)}[i, j] := 0$
  end
end
for $j = 1$ **to** $|\mathbf{T}|$ **do**
  for $i = 1$ **to** $|\mathbf{x}|$ **do**
    $D^{(1)}[i, j] := \max\{D^{(1)}[i-1, j-1] + \sigma(\mathbf{x}[i], \mathbf{T}[j]),$
    $D^{(1)}[i-1, j] + \sigma(\mathbf{x}[i], -), D^{(1)}[i, j-1] + \sigma(-, \mathbf{T}[j])\}$
  end
  for $i = 1$ **to** $|\mathbf{X}|$ **do**
    $D^{(2)}[i, j] := \max\{D^{(2)}[i-1, j-1] + \sigma(\mathbf{X}[i], \mathbf{T}[j]),$
    $D^{(2)}[i-1, j] + \sigma(\mathbf{X}[i], -), D^{(2)}[i, j-1] + \sigma(-, \mathbf{T}[j])\}$
  end
  $D^{(1)}[0, j] := \max\{D^{(1)}[|\mathbf{x}|, j], D^{(2)}[|\mathbf{X}|, j]\}$
  $D^{(2)}[0, j] := \max\{D^{(1)}[|\mathbf{x}|, j], D^{(2)}[|\mathbf{X}|, j]\}$
  for $i = 1$ **to** $|\mathbf{x}|$ **do**
    $D^{(1)}[i, j] := \max\{D^{(1)}[i, j], D^{(1)}[i-1, j] + \sigma(\mathbf{x}[i], -)\}$
  end
  for $i = 1$ **to** $|\mathbf{X}|$ **do**
    $D^{(2)}[i, j] := \max\{D^{(2)}[i, j], D^{(2)}[i-1, j] + \sigma(\mathbf{X}[i], -)\}$
  end
end

Algorithm 1: Pseudo-code for our nested wrap-around dynamic programming algorithm for the motif alignment problem for NTRs.

5.4.3 Correctness of the algorithm

We now prove by induction that the matrices $D^{(1)}$ and $D^{(2)}$ have been calculated correctly to produce the optimal alignment. In what follows let $NTR(\mathbf{x}, \mathbf{X})$ denote the set of all strings $\mathbf{N}$ that occur as substrings of exact NTRs with motifs $\mathbf{x}$ and $\mathbf{X}$.

Suppose that the two sub-matrices $D^{(1)}_{m,j-1}$ and $D^{(2)}_{l,j-1}$ have been correctly computed for some $j \geq 1$. That is, assume that $D^{(1)}[i, j-1]$ is the optimal alignment score of any alignment of $\mathbf{T}[1, j-1]$ against a string $\mathbf{N} \in NTR(\mathbf{x}, \mathbf{X})$ that ends with a suffix of $\mathbf{x}[1, i]$, and similarly that $D^{(2)}[i, j-1]$ is the optimal alignment score of any alignment of $\mathbf{T}[1, j-1]$ against a string $\mathbf{N} \in NTR(\mathbf{x}, \mathbf{X})$ that ends with a suffix of $\mathbf{X}[1, i]$. When

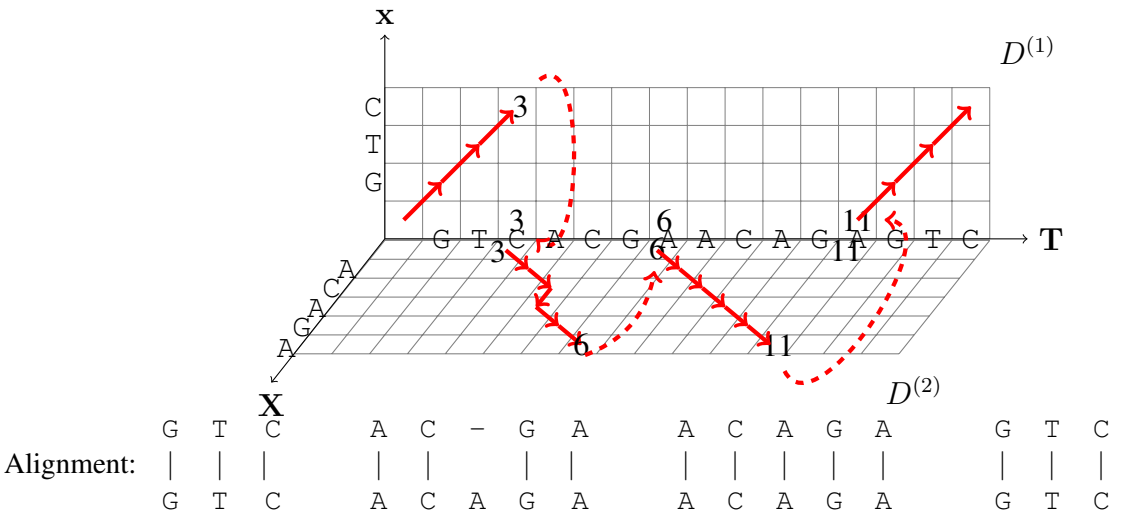


Figure 5.1: Visualisation of the algorithm applied to the string $\mathbf{T} = \texttt{GTCACGAACAGAGTC}$, with template motifs $\mathbf{x} = \texttt{GTC}$, $\mathbf{X} = \texttt{ACAGA}$. The matrix $D^{(1)}$ lies in the $(\mathbf{x}, \mathbf{T})$ plane, while $D^{(2)}$ lies in the $(\mathbf{X}, \mathbf{T})$ plane. The majority of the matrix entries have been omitted for clarity. Solid arrows represent the optimal alignment path, while dashed arrows indicate that the value at its tail is fed to the location at its head. The corresponding alignment appears below the diagram.

$i = 0$ our assumption is that

$$D^{(1)}[0, j-1] = D^{(2)}[0, j-1] = \max\{D^{(1)}[m, j-1], D^{(2)}[l, j-1]\},$$

so that this common value is the optimal score of an alignment of $\mathbf{T}[1, j-1]$ against a string $\mathbf{N} \in NTR(\mathbf{x}, \mathbf{X})$ ending in either $\mathbf{x}[m]$ or $\mathbf{X}[l]$.

Consider an alignment $(\bar{\mathbf{N}}, \bar{\mathbf{S}})$ of $\mathbf{S} = \mathbf{T}[1, j]$ against a string $\mathbf{N} \in NTR(\mathbf{x}, \mathbf{X})$ ending in $\mathbf{x}[1, i]$ or $\mathbf{X}[1, i]$. We consider three cases, according to the final characters of $\bar{\mathbf{S}}$ and $\bar{\mathbf{N}}$:

1. If $\bar{\mathbf{S}}$ ends in $\mathbf{T}[j]$ and $\bar{\mathbf{N}}$ in $\mathbf{x}[i]$, then deleting these characters gives an alignment of $\mathbf{T}[1, j-1]$ against a string $\mathbf{N}' \in NTR(\mathbf{x}, \mathbf{X})$ ending in $\mathbf{x}[i-1]$ if $i > 1$, or in either $\mathbf{x}[m]$ or $\mathbf{X}[l]$ if $i = 1$. It follows that

   $$\sigma\left(\bar{\mathbf{N}}, \bar{\mathbf{S}}\right) \leq D^{(1)}[i-1, j-1] + \sigma(\mathbf{x}[i], \mathbf{T}[j]),$$

   with equality when $\bar{\mathbf{N}}$ and $\bar{\mathbf{S}}$ are obtained by appending $\mathbf{x}[i]$ and $\mathbf{T}[j]$ to an optimal alignment at $D^{(1)}[i-1, j-1]$. A similar argument applies if $\bar{\mathbf{N}}$ ends in $\mathbf{X}[i]$.

2. If $\bar{\mathbf{S}}$ ends in $\mathbf{T}[j]$ and $\bar{\mathbf{N}}$ in a gap, then deleting these characters gives an alignment

of $\mathbf{T}[1, j-1]$ against $\mathbf{N}$. If $\mathbf{N}$ ends in $\mathbf{x}[i]$ then

$$\sigma\left(\bar{\mathbf{N}}, \bar{\mathbf{S}}\right) \leq D^{(1)}[i, j-1] + \sigma(-, \mathbf{T}[j]),$$

with equality when $\bar{\mathbf{N}}$ and $\bar{\mathbf{S}}$ are obtained by appending "$-$" and $\mathbf{T}[j]$ to an optimal alignment at $D^{(1)}[i, j-1]$. A similar argument applies if $\mathbf{N}$ ends in $\mathbf{X}[i]$.

3. If $\bar{\mathbf{S}}$ ends in a gap then we may express $\bar{\mathbf{S}}$ in the form

$$\bar{\mathbf{S}} = \bar{\mathbf{S}}'(-)^s,$$

where $s \geq 1$ is as large as possible. Let $\bar{\mathbf{N}} = \bar{\mathbf{N}}'\mathbf{M}$ with $|\mathbf{M}| = s$. Then $\left(\bar{\mathbf{N}}', \bar{\mathbf{S}}'\right)$ is an alignment of one the types considered in cases 1 and 2 above, so

$$\begin{aligned}\sigma\left(\bar{\mathbf{N}}, \bar{\mathbf{S}}\right) &= \sigma\left(\bar{\mathbf{N}}', \bar{\mathbf{S}}'\right) + \sigma(\mathbf{M}, (-)^s) \\ &\leq D^{(k')}[i', j] + \sigma(\mathbf{M}, (-)^s)\end{aligned}$$

for integers $i' \geq 1$ and $k' \in \{1, 2\}$ determined by the tail of $\mathbf{N}'$.

For conciseness let $\mathbf{Y}_1 = \mathbf{x}$ and $\mathbf{Y}_2 = \mathbf{X}$. Then the string $\mathbf{M}$ is an element of $NTR(\mathbf{x}, \mathbf{X})$ of length s ending with $\mathbf{Y}_k[i]$ and beginning with

$$\mathbf{M}[1] = \begin{cases} \mathbf{Y}_{k'}[i'+1] & \text{if } i' < |\mathbf{Y}_{k'}|, \\ \mathbf{x}[1] \text{ or } \mathbf{X}[1] & \text{if } i' = |\mathbf{Y}_{k'}|. \end{cases}$$

So what we must show is that for such strings we have

$$D^{(k)}[i, j] \geq D^{(k')}[i', j] + \sum_{a=1}^{s} \sigma(\mathbf{M}[a], -).$$

By the update rules we have

$$D^{(k'')}[a, j] \geq D^{(k'')}[a-1, j] + \sigma(\mathbf{Y}_{k''}[a], -)$$

for $k'' = 1, 2$ and $a \geq 1$, so the necessary inequality will be true by induction

provided we can show that we still have

$$D^{(k'')}[0,j] = \max\{D^{(1)}[m,j], D^{(2)}[l,j]\} \tag{5.1}$$

after the second update round. This equality follows from the fact that the larger of $D^{(1)}[m,j], D^{(2)}[l,j]$ is unchanged during the second round. Indeed, if the value $D^{(1)}[m,j]$ is changed during the second round then it must have been *increased* to

$$D^{(1)}[m,j] = D^{(1)}[0,j] + \sum_{b=1}^{m} \sigma(\mathbf{x}[b], -),$$

and this is strictly less than $D^{(1)}[0,j]$, because $\sigma(\alpha, -) < 0$ for all α. A similar argument applies to $D^{(2)}[l,j]$, so the larger of these is unchanged and remains the maximum.

By the above we have $\sigma(\bar{\mathbf{N}}, \bar{\mathbf{S}}) \leq D^{(k)}[i,j]$. It remains to show that there is in fact an alignment with score $D^{(k)}[i,j]$ when $D^{(k)}[i,j] = D^{(k)}[i-1,j] + \sigma(\mathbf{Y}_k[i], -)$. Consider the trace back procedure beginning from $D^{(k)}[i-1,j]$. This must eventually reach an (i',j)-entry of either $D^{(1)}$ or $D^{(2)}$ that derives from column $j-1$ (since for example the largest entry in each column of each matrix must be derived this way), and we obtain the desired alignment by appending suitable strings to an optimal alignment at this point.

Cases 1–3 above show that $D^{(1)}[i,j]$ and $D^{(2)}[i,j]$ have been correctly computed for $i \geq 1$, and equation (5.1) shows that $D^{(1)}[0,j]$ and $D^{(2)}[0,j]$ have been too. It follows by induction that both matrices $D^{(1)}$ and $D^{(2)}$ have been correctly computed.

5.4.4 Extension to nested tandem repeats with three or more motifs

Our algorithm is easily adapted to the motif alignment problem for more complex NTRs built from three or more motifs $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_r$. For each $k = 1, \ldots, r$ we introduce an $|\mathbf{X}_k| \times |\mathbf{T}|$ matrix $D^{(k)}$,where $|\mathbf{T}|$ is the text containing the NTR, and we initialise these as in Section 5.4.2. After the jth column of each matrix has been filled as in the first round above we update $D^{(k)}[0,j]$ with $\max\{D^{(i)}[|\mathbf{X}_i|, j] | i = 1, \ldots, r\}$ for each k, and then run a second round as above to update the jth column of each matrix. Once the matrices have been filled, an optimal alignment may then be found using the standard

trace-back procedure. The time and space complexity for the r-motif alignment algorithm is $O(|\mathbf{T}|(|\mathbf{X}_1|+|\mathbf{X}_2|+\cdots+|\mathbf{X}_r|)$ as it takes $O(|\mathbf{T}||\mathbf{X}_k|)$ time and space to fill each matrix $D^{(k)}$. In the case where the motifs have the same length $|\mathbf{X}|$ then the complexity would be $O(|\mathbf{T}|(k|\mathbf{X}|))$.

5.5 Conclusion

In this chapter, we presented an algorithm to solve the problem of the alignment of nested tandem repeats. This algorithm has $O(|\mathbf{T}|(|\mathbf{x}|+|\mathbf{X}|))$ time complexity. The nested WDP alignment is incorporated in the program NTRFinder (Matroud et al., 2012b) which is described in Chapter 4, as part of the verification phase.

Chapter 6

A comparison of three heuristic methods for solving the parsing problem for tandem repeats

This chapter is an extended version of the text of *A Comparison of Three Heuristic Methods for Solving the Parsing Problem for Tandem Repeats*, A. Matroud, C. Tuffley, D. Bryant, and M. Hendy, Advances in Bioinformatics and Computational Biology (Matroud et al., 2012a). It has been reformatted for consistency with the rest of thesis. Some definitions have been removed to avoid redundancy. A list of the materials and text modified is given below:

- Comments on how the methods could test for dynamic boundaries and the result of a further simulation with fixed and dynamic boundaries were added in section 6.6.1.
- The tests on real sequence data in 6.6.2 were added.
- Several corrections and consistency matters were addressed.

6.1 Abstract

In many applications of tandem repeats the outcome depends critically on the choice of boundaries (beginning and end) of the repeated motif: for example, different choices of pattern boundaries can lead to different duplication history trees. However, the best

choice of boundaries or *parsing* of the tandem repeat is often difficult to determine: in real biological sequences it is frequently observed that the flanking regions before and after the tandem repeat contain partial approximate copies of the motif, making it difficult to determine where the tandem repeat (and hence the motif) begins and ends. We define the *parsing problem* for tandem repeats to be the problem of discriminating among the possible choices of parsing.

In this paper we propose and compare three heuristic methods for solving the parsing problem, under the assumption that the parsing is fixed throughout the duplication history of the tandem repeat. The three methods are PAIR, which minimises the number of pairs of common mutations which span a boundary; VAR, which minimises the total number of variants of the motif; and MST, which minimises the length of the minimum spanning tree connecting the variants, where the weight of each edge is the Hamming distance of the pair of variants. We test the methods on simulated data over a range of motif lengths and relative rates of substitutions to duplications, and show that all three perform better than choosing the parsing arbitrarily. Of the three MST typically performs the best, followed by VAR then PAIR.

6.2 Introduction

Genomic DNA has long been known to contain *tandem repeats* repetitive structures in which many approximate copies of a common segment (the *motif*) appear consecutively. The copies of the motif are usually polymorphic, which makes tandem repeats a useful tool for phylogenetics and for inter-population studies (Rivals, 2004a); in addition, highly polymorphic tandem repeats can be used to discriminate among individuals within a population, and have proved to be useful for DNA fingerprint techniques (Jeffreys et al., 1980). Because of this, many algorithms have been developed to find tandem repeats; align tandem repeats; compare DNA sequences containing tandem repeats (Behzadi and Steyaert, 2003; Berard and Rivals, 2003; Sammeth and Stoye, 2006); and construct the duplication history tree (DHT) of a tandem repeat (Rivals, 2004a; Lajoie et al., 2007; Bertrand et al., 2008; Chauve et al., 2008).

In many important applications of tandem repeats the outcome depends critically on the choice of boundaries (beginning and end) of the repeated motif. We will refer to a

choice of boundaries as a *parsing* of the tandem repeat. For a tandem repeat with motif of length ℓ there are ℓ possible parsings, and different choices of parsing can for example lead to different duplication history trees. However, the "true" parsing is often ambiguous, as the flanking regions (the ℓ nucleotides immediately preceding and following the tandem repeat) often contain partial approximate copies of the motif, making it difficult to decide where the tandem repeat (and consequently the motif) begins and ends. It is therefore highly desirable to find methods to discriminate among the possible parsings, and we will refer to the problem of doing so as the *parsing problem* for tandem repeats.

The parsing problem does not appear to have received a great deal of attention to date in the literature, and in many tandem repeat search tools the criteria for setting boundaries appear to be subjective or arbitrary (for example (Crochemore, 1981; Matroud et al., 2012b; Hauth and Joseph, 2002; Stoye and Gusfield, 2002; Benson, 1999; Sagot and Myers, 1998). To the best of our knowledge the only reference on this problem to date is by (Benson and Dong, 1999b), who propose a method to solve the parsing problem based on a tandem repeat duplication model which allows dynamic boundaries (that is, duplications may occur on different boundaries throughout the duplication history). The purpose of this paper is to present three new criteria to select the parsing, under the assumption that the pattern boundaries are fixed throughout the duplication process, which was suggested by Fitch (1977a).

One possible method for selecting the parsing is to choose the parsing that minimises the parsimony score of the resulting DHT. However, obtaining the maximum parsimony DHT can be computationally expensive (especially when the motif is long or there are many copies) (Foulds and Graham, 1982), and in some cases the maximum parsimony tree cannot be expressed as a duplication tree (Gascuel et al., 2003). In these circumstances it may be preferable to find more tractable measures for comparing parsings. The three criteria we propose are heuristic, and are intended as easily computed surrogates for the score of the maximum parsimony tree. They are each based on the observation that the histories of two nucleotide substitutions that occur at nearby sites less than ℓ bases apart at some stage in the evolution of the tandem repeat can be different depending on whether they occur in the same or adjacent copies of the motif.

The three methods are

1. PAIR, which minimises the number of pairs of common mutations which span a

boundary,

2. VAR, which minimises the total number of variants of the motif, and

3. MST, which minimises the length of the minimum spanning tree connecting the variants, with each edge length being the Hamming distance of the pair of variants.

These three methods work on tandem repeats of the same length. However, if the tandem repeat copies do contain insertion and deletion then these copies should be aligned using available alignment algorithms such as (Fischetti et al., 1993) for tandem repeats and (Matroud et al., 2011) for nested tandem repeats.

We test the methods on simulated data (for which the parsing used to generate the tandem repeat is known) over a range of motif lengths and relative rates of substitutions to duplications. We show that all three methods perform better than choosing the parsing arbitrarily, and that of the three MST typically performs the best, followed by VAR then PAIR.

We have applied our three methods on four tandem repeats taken from the microorganisms tandem repeats database (Denœud and Vergnaud, 2004) and (Visca et al., 2011). These four tandem repeats belong to two families. We show that the three methods suggest a parsing region instead of a unique parsing point. There is a consistency in the results using the MST and the PAIR methods.

6.3 Definitions and Background

An *exact tandem repeat* is a string comprising two or more contiguous exact copies of a substring $\mathbf{X}$, called the tandem repeat *motif*. We obtain an *approximate tandem repeat* by allowing approximate rather than exact copies of the template motif $\mathbf{X}$. We will refer to each copy of the motif as a *segment*. We define a *mode* motif to be a sequence of length ℓ where the $i-$th nucleotide is a most common nucleotide at the $i-$th site among the segments, for all $1 \leq i \leq \ell$. Note the mode motif is not necessarily unique. We define the set of *variants* of a tandem repeat with motif $\mathbf{X}$ to be the set of distinct segments that are observed in the sequence.

We define the *distance graph* of the variants to be the weighted graph with vertex set the set of variants, and an edge between each pair of variants with weight equal to their

Hamming distance.

Let $\mathbf{X}[i] = x_i x_{i+1} \dots x_\ell x_1 \dots x_{i-1}$ be the i^{th} cyclic permutation of the motif $\mathbf{X}$ (also referred as the *parsing point* i of the motif $\mathbf{X}$), where ℓ is the length of $\mathbf{X}$. We define the *parsing problem* for tandem repeats to be the problem of determining which of the ℓ possible cyclic permutations of $\mathbf{X}$ produces the minimal parsimony score of the tandem repeat duplication history tree.

6.4 The importance of the parsing problem

In order to use a tandem repeat region in a phylogenetic study we need to infer the number and size of the edit operations that occurred in this region, transforming a single segment into the observed tandem repeat. However, the number and size of the inferred edit operations depend on the parsing we select. In the following example, we illustrate the implications of having two different parsing points on the inferred number and size of the edit operations.

Example Consider the following sequence which contains an approximate tandem repeat with periodicity 4:

$$\texttt{AGACCACGAACGTACGAACGTATTA}. \tag{6.1}$$

There are 4 possible parsings. If we set the first boundary point such that the first repeat copy is `ACGA` we obtain

$$\texttt{AGACC}\boxed{\texttt{ACGA}}\boxed{\texttt{ACGT}}\boxed{\texttt{ACGA}}\boxed{\texttt{ACGT}}\texttt{ATTA}, \tag{6.2}$$

with mode motif `ACGA` (note that in this case `ACGT` is another mode motif). If we shift the frame one nucleotide to the left we obtain a different parsing

$$\texttt{AGAC}\boxed{\texttt{CACG}}\boxed{\texttt{AACG}}\boxed{\texttt{TACG}}\boxed{\texttt{AACG}}\texttt{TATTA}, \tag{6.3}$$

with a unique mode motif `AACG`.

The DHT that best describes the tandem repeat depends on the parsing. The minimal DHT for the parsing of (6.2) is shown in Figure 6.4(a), and the minimal DHT for the parsing of (6.3) is shown in Figure 6.4(b). The two trees involve different sets of edit

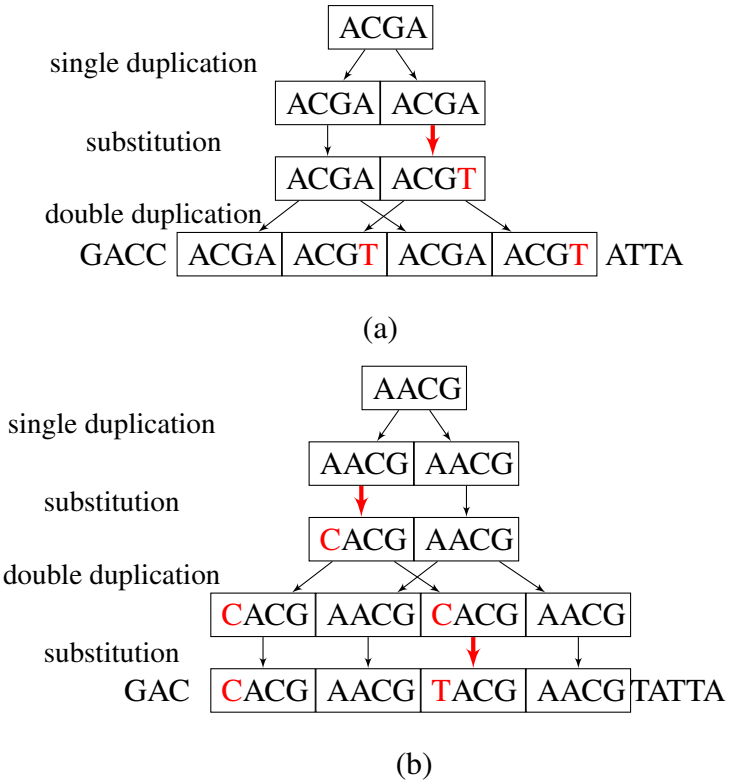


Figure 6.1: The DHTs inferred from the two parsing of the example in Section 6.4. The parsing (a) has a DHT with two duplications and a single substitution, in parsing (b) the DHT has two duplications and two substitutions. In both cases we see that the number of events is minimal for that parsing. By the parsimony principle, we prefer parsing (a) over parsing (b) as its DHT requires fewer mutational events.

operations; the first requires fewer, and so is to be preferred on parsimony grounds. Note that the parsing with mode motif `CGAA` that results from a frame shift one nucleotide to the right in (6.2) gives the same DHT as Figure 6.4(a).

Similarly, duplication of nested tandem repeats can be modeled in the same way as tandem repeats. However, a nested tandem repeat starts with two motifs $\mathbf{x}$ and $\mathbf{X}$, then a series of duplications, deletions, substitutions occurs on the two motifs which results in a nested tandem repeat sequence.

6.5 Heuristic methods to estimate tandem repeat parsing

In Section 6.4, we were able to discriminate between the parsings of (6.2) and (6.3) on the basis of the parsimony scores of their duplication history trees. However, when considering large and long tandem repeats, obtaining the maximum parsimony duplication

history tree of the motif copies can be computationally expensive, and in some cases the maximum parsimony tree cannot be expressed as a duplication tree (Gascuel et al., 2003). It may be preferable to avoid these constructions when comparing different parsings.

Below, we describe three heuristic approaches to discriminating between the different possible parsings. They are each based on the observation that the histories of two nucleotide substitutions that occur at nearby sites less than ℓ bases apart at some stage in the evolution of the tandem repeat can be different depending on whether they occur in the same segment or in adjacent segments.

Recall that $i\theta$ denotes a substitution of the type $\theta \in \{\alpha, \beta, \gamma\}$ at site i. Suppose a substitution $i\theta_1$ producing nucleotide ν occurs at site i in one segment in the sequence. The variant containing ν may be duplicated a number of times before a second substitution $j\theta_2$ producing nucleotide ω occurs at site j within ℓ bases of ν at site $i_1 \equiv i \bmod \ell$, with $j = i_1 + k,\ 0 < |k| < \ell$. Now suppose there are further duplications producing further copies of variants containing ν and ω.

If ω is in the same segment as ν, and there are no subsequent parallel substitutions producing ω at any other site $j_1 \equiv j \bmod \ell$, then we will observe variants with ν, variants with ν and ω together, but no variants with ω alone. Hence each ω at site j_2 will have a companion ν at site $i_2 = j_2 - k$. Figure 6.2 illustrates this scenario.

However, if the substitution producing ω were in an adjacent segment, and there were subsequent duplications, then we can observe variants containing ν alone, and variants containing ω alone. In some of these cases there may be pairs of ν and ω still k nucleotides apart, but there can also be copies of ω with no adjacent ν. For the purpose of the arguments, we will assume that ν is to the left of ω in the segments that contain both.

The three methods we have introduced to exploit this observation are listed below. We note the performance of these methods is dependent on the ratio m of substitutions to duplications. If m is small there will only be a small number of (or possibly no) pairs of close substitutions to indicate the likely parsing. In these cases, there may be multiple locations that are optimal on some of our scores, and some additional criteria (or perhaps random choice) would be required to identify a preferred parsing.

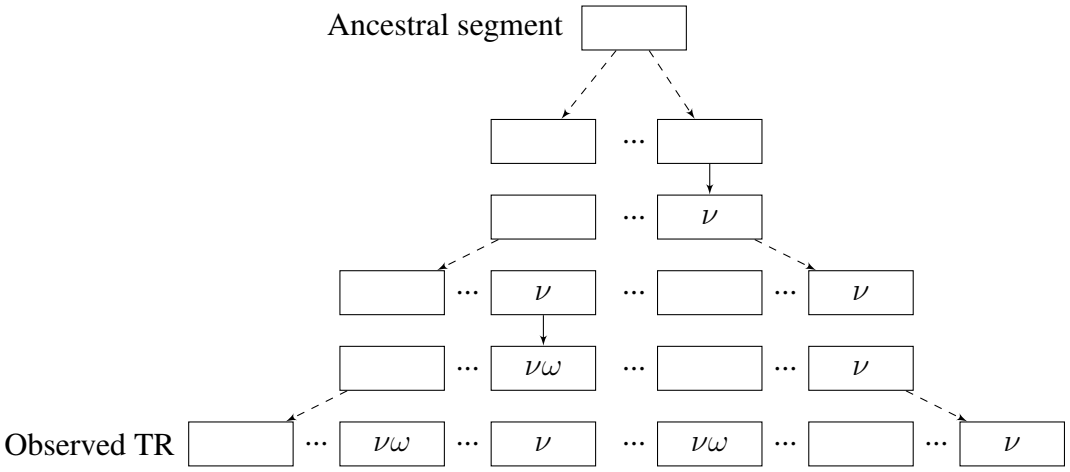


Figure 6.2: A schematic of the duplication process. The dashed arrows represent series of duplication events of different sizes, and the solid arrows represent single nucleotide substitutions.

6.5.1 PAIR — the adjacent pairs method

Here we consider all occurrences of a pair of substitutions ν and ω which occur at least twice (to restrict attention to pairs which may have been duplicated) at sites less than ℓ bases apart, with the ν's at some sites $i \bmod \ell$, and the ω's at some sites $j \bmod \ell$, and with each ω always adjacent to an ν. We then note all the sites mod ℓ between each adjacent ν and ω and record their frequency. For the method PAIR we select those sites mod ℓ which are counted in this way with lowest frequency as our preferred location for the boundary of the motifs. Provided there are sufficient substitutions so that multiple substitutions occur in some ancestral segment which is subsequently duplicated, then this should discriminate between sites. This discrimination should remain as the ratio m of substitutions to duplications grows, as the frequency of parallel substitutions should always be lower than unique substitutions.

To illustrate this method, consider the tandem repeat in the example on page 7 in Section 1.3. Indexing from site 1 in the first box, the substitutions from the mode motif are: 7α, 12β, 19α, 24β, 27α, 29β, 33α, 35β, 37α, 45β, 51α. From this list we observe the only sets of pairs at most 5 bases apart which occur more than once are $(1\alpha, 6\beta)$ at sites $(7, 12)$ and $(19, 24)$, and $(3\alpha, 5\beta)$ at sites $(27, 29)$ and $(33, 35)$. These pairs do not straddle the parsing boundary in (1.1), but either one or both pairs will straddle any other proposed parsing boundary, and contribute 2 (the frequency of the pair) to the score of the corresponding parsing. The resulting score for each parsing is shown in Table 6.1. The

Method	Consensus pattern of parsing					
	CATGGT	TCATGG	GTCATG	GGTCAT	TGGTCA	ATGGTC
PAIR	0	2	2	4	4	2
VAR	6	7	8	7	7	7
MST	5	6	7	7	6	6

Table 6.1: The score under each of our three methods for each parsing of the tandem repeat in the example on page 7 in Section 1.3. See Sections 6.5.1 to 6.5.3 for details. Each method returns the first parsing CATGGT as the preferred parsing, since this minimises the score with respect to each method.

method selects the generating parsing CATGGT of (1.1) as the preferred parsing.

6.5.2 VAR — the number of variants method

For this method we consider each of the ℓ possible parsings, and for each parsing we count the number of variants. When the proposed parsing is correct, then the variants containing ν and containing ν and ω together will be counted, leading to 2 additional variants observed. However, if we propose a parsing which separates these, then we may find variants containing ν alone, variants containing ω alone, and variants containing both, with ω to the left of ν. Hence the VAR method selects the parsing or parsings which minimise the number of distinct variants.

As the ratio m of substitutions to duplications grows, so too will the number of distinct variants, and for m large, they may be almost all distinct, irrespective of the location of the proposed parsing. Hence for larger values of m, the VAR method may lose its discriminatory power.

To illustrate the method we again consider the tandem repeat in the example on page 7 in Section 1.3. For the parsing shown there are six variants, as listed in (1.2), so the score for this parsing is 6. The scores for the other six parsings are given in Table 6.1. The method selects the first parsing CATGGT as the preferred parsing, as this minimises the score.

6.5.3 MST — the minimum spanning tree method

When m is small, so the number of variants is small, then the Maximum Parsimony (MP) tree connecting the variants is likely to be $1-$connected, and the length of the MP tree will be the number of variants minus one. However as m grows, the MP tree may require

Steiner points (representing ancestral variants that are no longer present in the extant set of variants), and the length of the MP tree may be more discriminating in determining the parsing. However, as the determination of the MP tree can be NP-hard, we can use the length of the minimum spanning tree, as a quick measure of the relative relatedness of the variants. To avoid the issue of connectedness and the requirement of Steiner points, we take the minimum spanning tree of the distance graph of the variants. Recall that this is the graph with vertex set the set of variants, and an edge between two variants a and b with weight equal to their Hamming distance.

We propose the MST, the length of the minimum spanning tree of the variants as our third measure. We expect MST to agree with VAR, but to be more accurate for larger values of m.

We illustrate this method in Figure 6.3, which shows minimum spanning trees of the variants distance graphs for two of the six parsings of the tandem repeat in the example on page 7 in Section 1.3. The first parsing `CATGGT` has a minimum spanning tree of length 5, while the fourth parsing `GGTCAT` has a minimum spanning tree of length 7. Note that in this latter case the minimum spanning tree length is greater than $\mathrm{VAR} - 1$, illustrating the potential for MST to be more discriminating than VAR. The scores for all six parsings appear in Table 6.1, and the method again selects the first parsing `CATGGT` as the preferred parsing.

6.6 Results and discussion

6.6.1 Tests on simulated data

We generated 90000 synthetic DNA sequences to compare the accuracy of the three proposed parsing methods. Each simulated sequence contained an approximate tandem repeat of around 100 copies of an ancestral motif of ℓbp ($\ell = 10, 50, 100$). These were generated by a stochastic evolutionary process of motif duplication (where the frequency of duplicating a segment of κ motif copies was proportional to $\frac{1}{\kappa}$), with nucleotide substitutions accumulating at a frequency of m substitutions per duplication. We applied each of the three proposed parsing methods to each simulated sequence and recorded whether the predicted parsing agreed with the parsing used to generate the tandem repeat.

We generated 100 samples for each value of $m = 0.1, 0.2, \ldots, 30.0$ and we report the

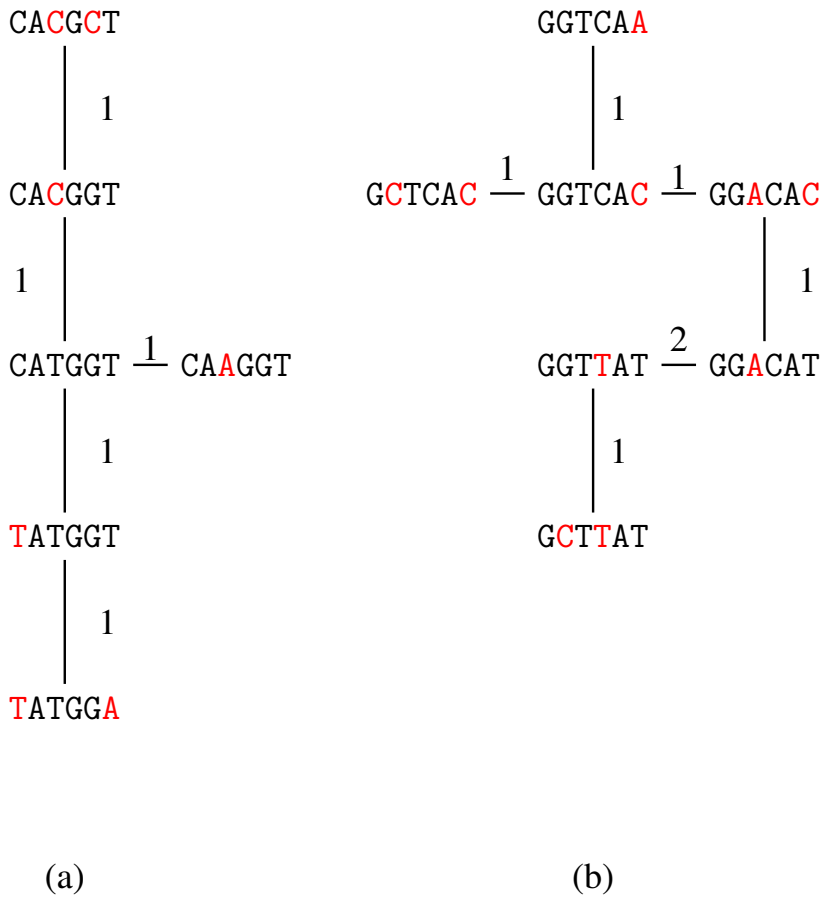
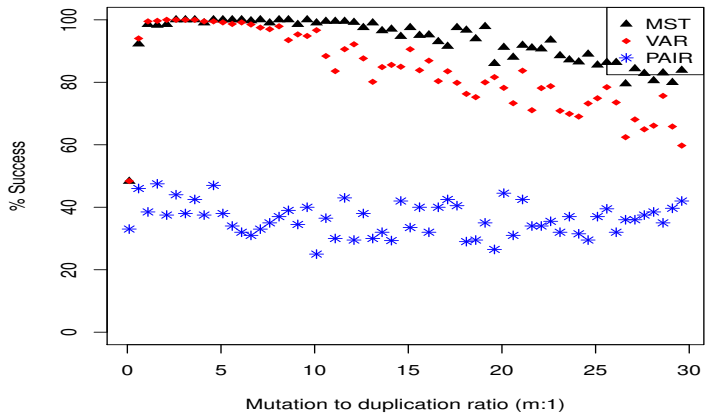


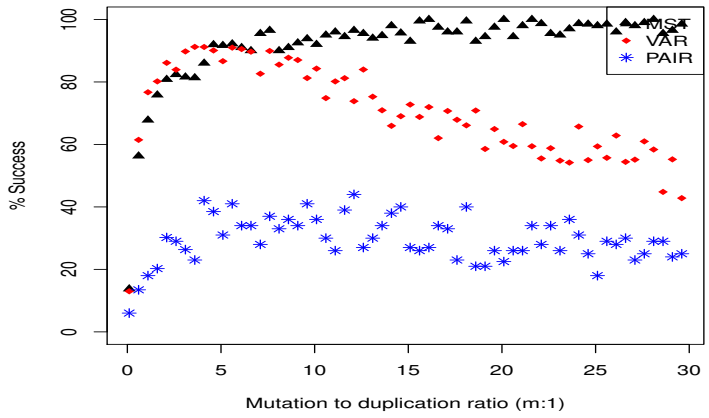
Figure 6.3: Minimum spanning trees of the variants distance graphs for two of the six parsings of the tandem repeat in the example on page 7 in Section 1.3. (a) A minimum spanning tree of length 5 for the parsing CATGGT. (b) A minimum spanning tree of length 7 for the parsing GGTCAT.

percentage success for each method and value of m in Figure 6.4. For the purposes of this plot, "success" means that the set of minima reported by the method contains the true parsing, or one of the two parsings adjacent to the true parsing. The average number of minima returned by each method is plotted in Figure 6.5 (plotted as a percentage of the number of possible parsings (the motif length ℓ)), which shows that each method typically returned only a small fraction of the possible parsings — often only one or two in the case of PAIR and MST. Figure 6.6 shows the number of times each method reports minima that do not include the true parsing.

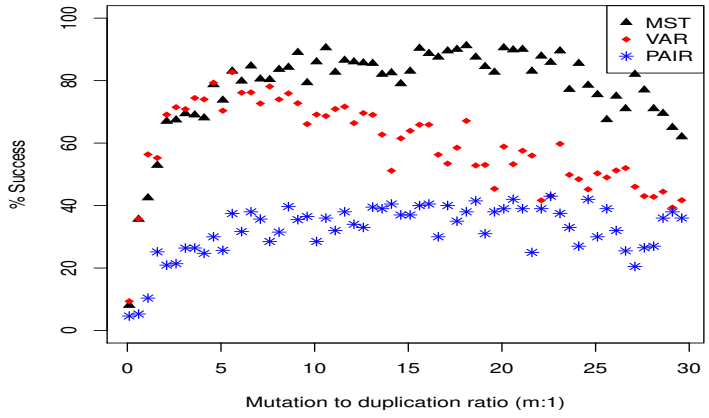
We note that the PAIR method performed poorly (at about 35% accuracy) over the range of values of m, whereas VAR and MST showed above 90% accuracy for m in the range of about 0.3 to 5. Nevertheless all three methods performed better than setting the parsing arbitrarily, as the null method of randomly assigning a parsing would be expected



(a) Motif length=10.

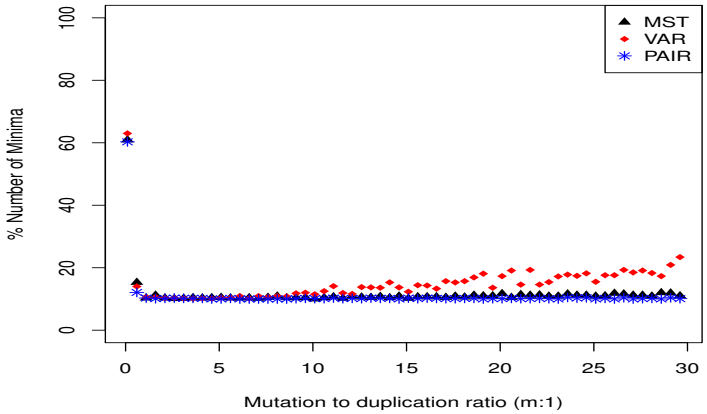


(b) Motif length=50.

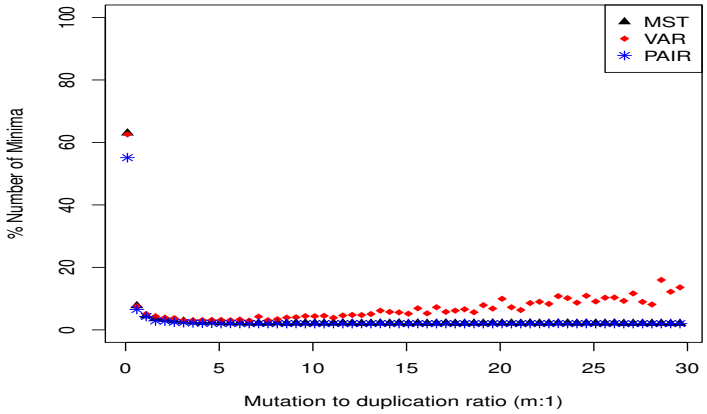


(c) Motif length=100.

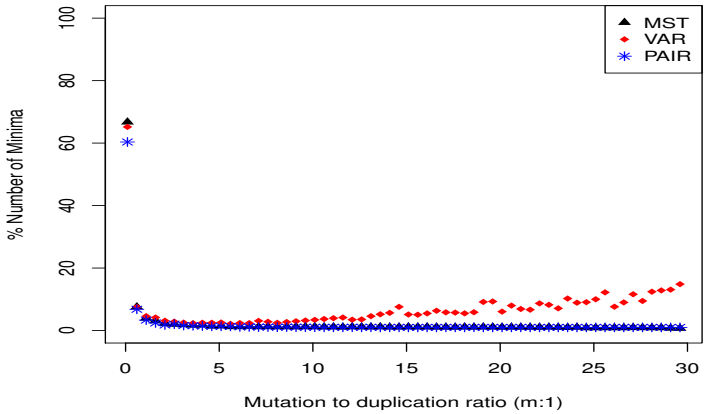
Figure 6.4: Percentage success plotted against the relative mutation rate for each of the three methods. The y-axis represents the percentage of simulations for which the set of minima contains the true boundary point or the points that are one step away from the true boundary, plotted against each relative mutation rate $m = 0.1, 0.2, \ldots, 30$. The motif length is (a) $\ell = 10$, (b) $\ell = 50$ and (c) $\ell = 100$.



(a) Motif length=10.

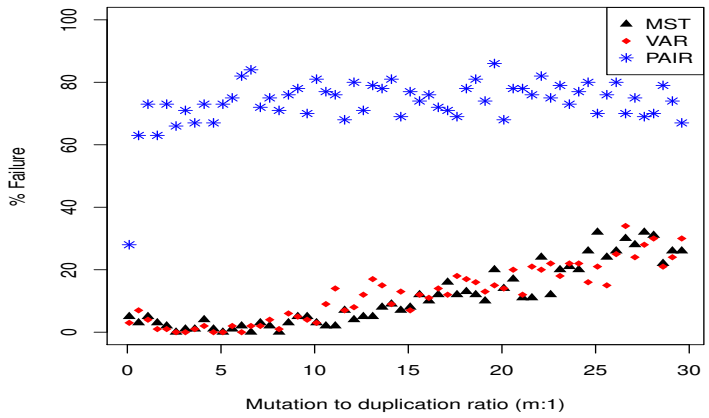


(b) Motif length=50.

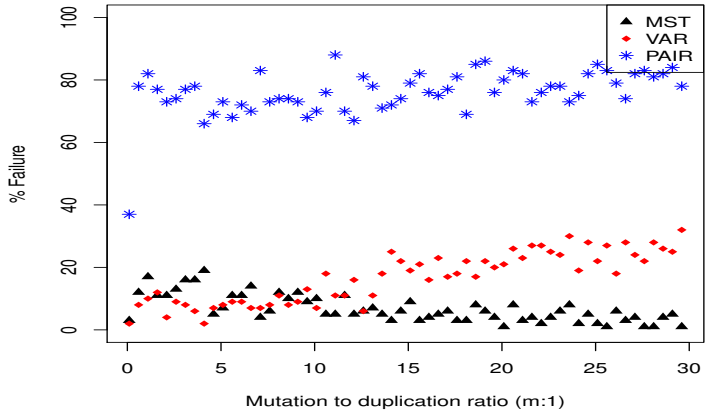


(c) Motif length=100.

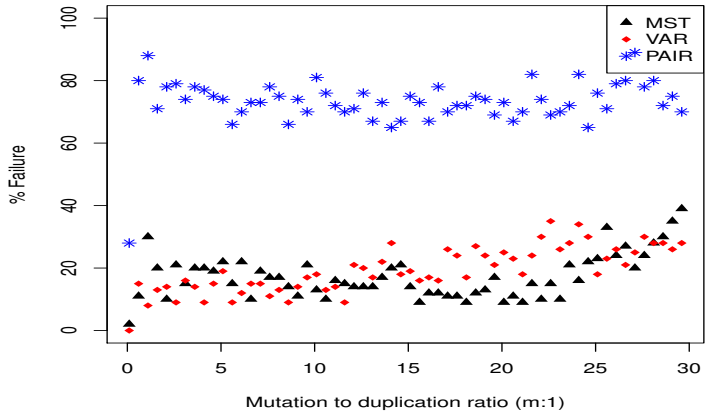
Figure 6.5: The number of minima plotted against the relative mutation rate. The y-axis represents the average number of minima reported by each method, expressed as a percentage of the number of possible parsings (the motif length ℓ), plotted against each relative mutation rate $m = 0.1, 0.2, \ldots, 30$. The motif length is (a) $\ell = 10$, (b) $\ell = 50$ and (c) $\ell = 100$.



(a) Motif length=10.



(b) Motif length=50.



(c) Motif length=100.

Figure 6.6: Percentage failure plotted against the relative mutation rate for each of the three methods. The y-axis represents the percentage of simulations for which the true boundary is not among the reported minima, plotted against each relative mutation rate $m = 0.1, 0.2, \ldots, 30$. The motif length is (a) $\ell = 10$, (b) $\ell = 50$ and (c) $\ell = 100$.

to achieve accuracy of $3/\ell = 30\%, 6\%, 3\%$ for $\ell = 10, 50, 100$. We also note that the PAIR method has lower sensitivity to the motif length than the other two methods. For much of this range the MST method performs better than VAR. The VAR method does not perform better than MST in terms of accuracy (Figure 6.4), and it also produces solutions containing a larger number of optimal parsings (Figure 6.5).

The PAIR method has the computational advantage that the scores for all ℓ possible parsings can be computed simultaneously, whereas the VAR and MST scores must be computed for each possible parsing in turn. Nevertheless, their observed accuracy indicates they are preferred to the PAIR method, when the motif length is not large. MST performed better on average than VAR, so our results suggest MST to be the preferred method of predicting parsing.

To test the assumption that the motif boundaries are fixed throughout the duplication history of the tandem repeat, we ran a simulation that generated tandem repeats with dynamic boundaries. Several tandem repeats of motif length 10 were generated, in which the boundaries were either (a) fixed at parsing point 5 throughout the duplication history; (b) randomly located at parsing points 0 and 5, with equal probabilities; or (c) uniformly distributed throughout the motif. Each tandem repeat had around 100 copies of the motif, and the substitution frequency used was 1 substitution per 10 duplications. The three methods produce flat score distributions when the boundaries are not fixed as shown in Figure 6.7(b) and Figure 6.7(c).

6.6.2 Tests on real sequence data

To test the three methods on real data we applied them on four tandem repeats that are grouped into two families. Within each family, the tandem repeats are taken from the same or closely related sequences, and have similar motifs of the same length.

Our expectation is that tandem repeats within the same family are likely to be homologous and so are likely to have the same parsing points. The tandem repeats used in this test are taken from the minisatellite database in (Denœud and Vergnaud, 2004). The details of the two datasets are as follows.

- Two tandem repeats with a common motif of length 18bp from two different locations (609417–610022 (SA1) and 604971–605498 (SA2)) in the DNA sequence of *Staphylococcus aureus*_04_02981. The number of motif copies are 33 and 30

respectively.

- Two tandem repeats with a common motif of length 18bp at locations (634730 to 635239 (SAC1) and 643148 to 643627 (SAC2)) in the DNA sequence of *Staphylococcus aureus* COL. The number of motif copies are 28 and 27 respectively.

The results of applying our methods to each family are plotted in Figures 6.8 and 6.9.

We start our analysis by parsing the tandem repeats of the two *Staphylococcus aureus* strains at the same point. This can be done by aligning the two tandem repeats. In this data, the mode motifs of the tandem repeat of each strain are `TCAGATAGCGATTCAGAT` and `TCAGATAGCGACTCAGAT`.

The parsing points on the *Staphylococcus aureus* data set using the VAR methods are $\{13, 14, 15, 16, 17, 18\}$ as in Figure 6.8(a). The MST method suggests the parsing points to be at any of the following points $\{16, 17, 18\}$, see Figure 6.8(b). The PAIR method shows a unique minimum at point 7 and local minima at $\{16, 17, 18\}$ (Figure 6.8(c)). The three methods show some consistency in having minima at $\{16, 17, 18\}$ and also a drop in the score around the seventh site. The three methods also suggest that the tandem repeat in both strains of *Staphylococcus aureus* can be parsed at similar regions, namely $\{16, 17, 18\}$.

The *Staphylococcus aureus* COL tandem repeat data set shows consistency between the PAIR and the MST methods in terms of minima produced by both methods (6.9(b) and (c)). Both MST and PAIR suggest the parsing to be at any point between 10 and 12. The VAR methods suggest the points between 13–15 (Figure 6.9(a)).

6.7 Conclusion

Three heuristic methods to solve the parsing problem have been proposed and tested. These methods are intended as easily computed surrogates for the score of the duplication history tree as a means of discriminating between alternate parsings.

Based on the assumption that the boundaries are fixed through the evolution history we expect the scores of these three methods to discriminate among all possible parsing points. We showed by a simulation that when this assumption is not imposed, the distribution of minima for each of the three methods may indicate that the parsing point varied during the evolution of the tandem repeats.

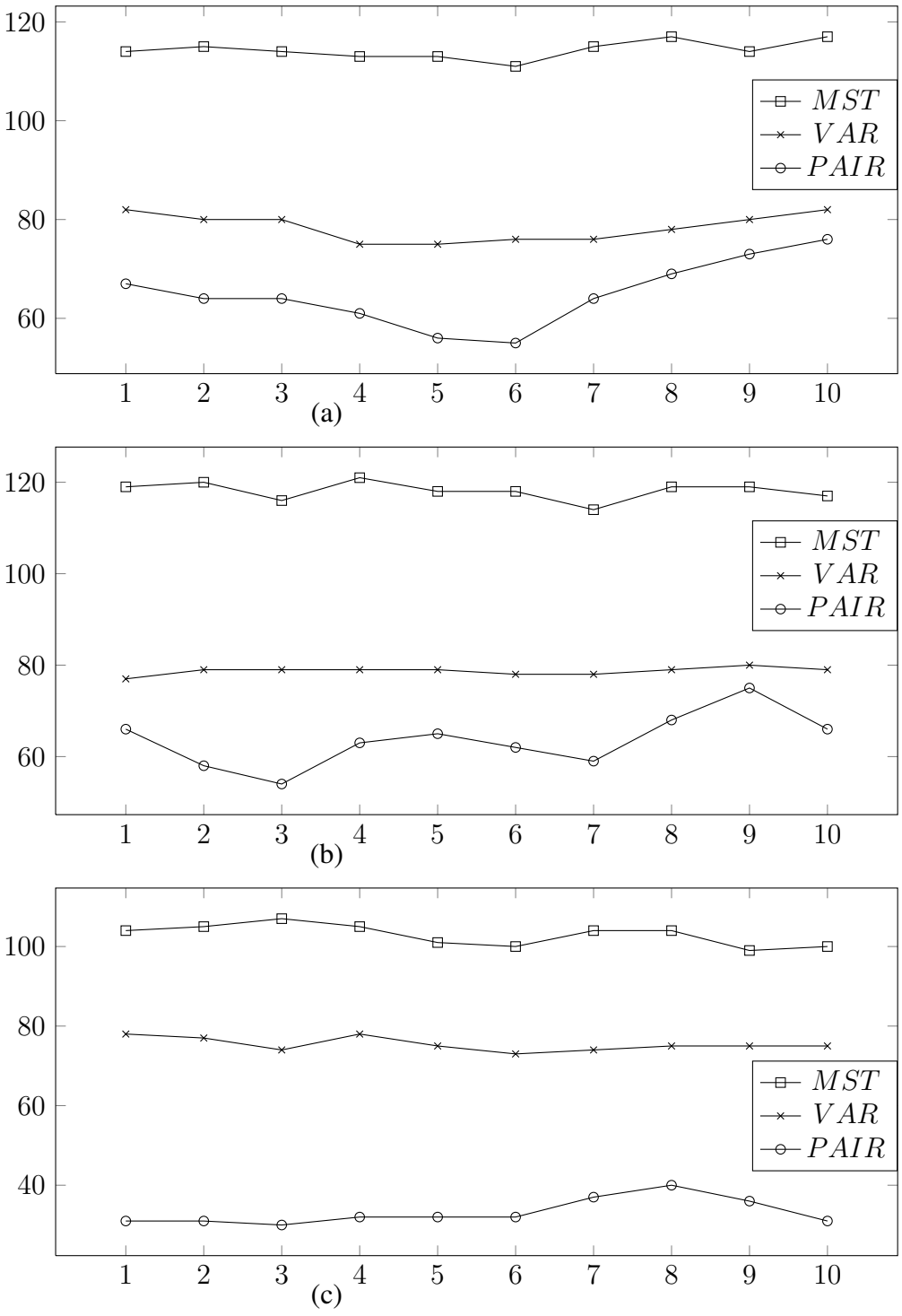


Figure 6.7: The scores of the three methods when the motif boundaries are (a) fixed at point 5 through the duplication history tree; (b) randomly located at either at 0 or 5 with equal probabilities at each duplication; (c) uniformly distributed through the motif. The motif length was 10.

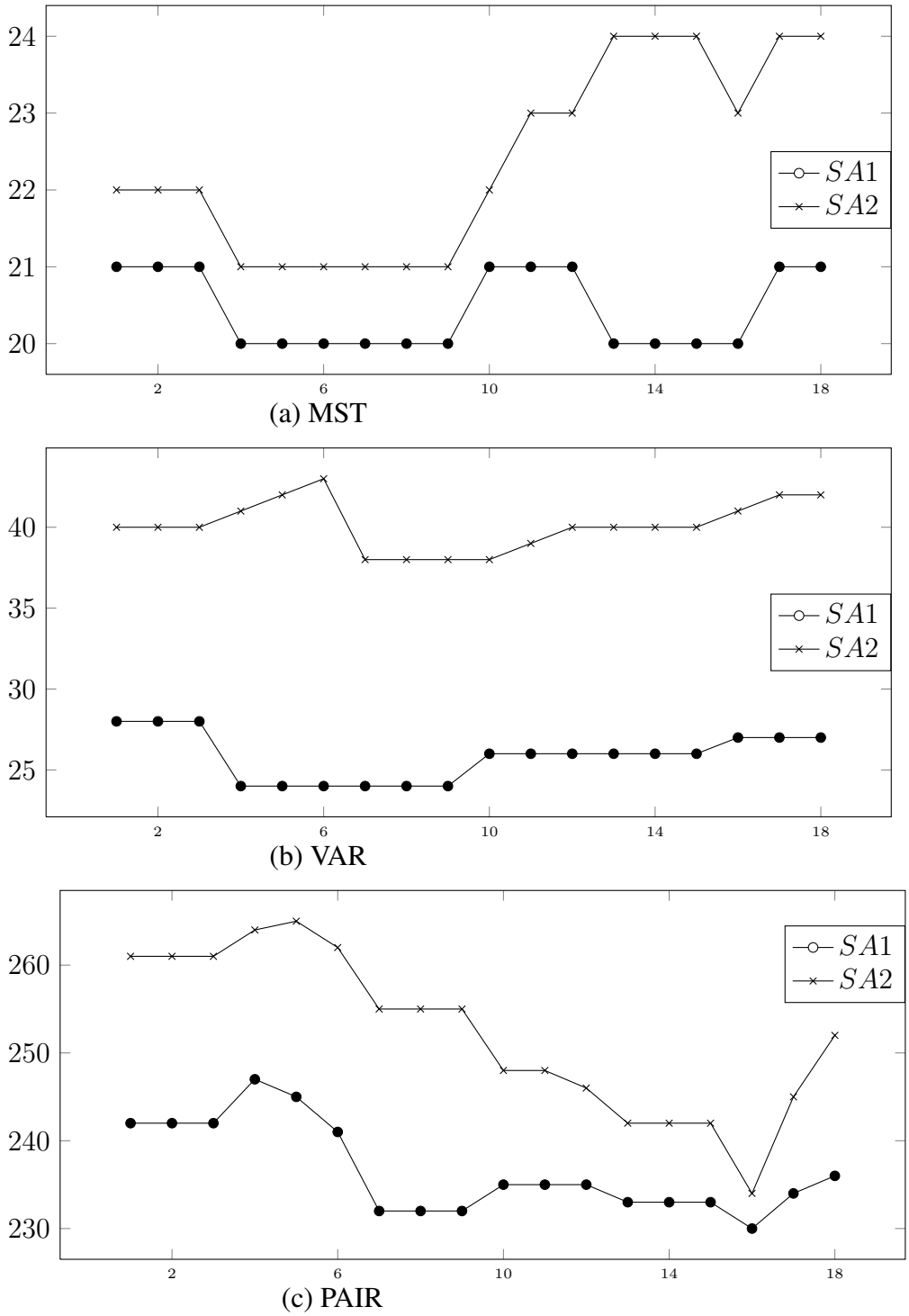
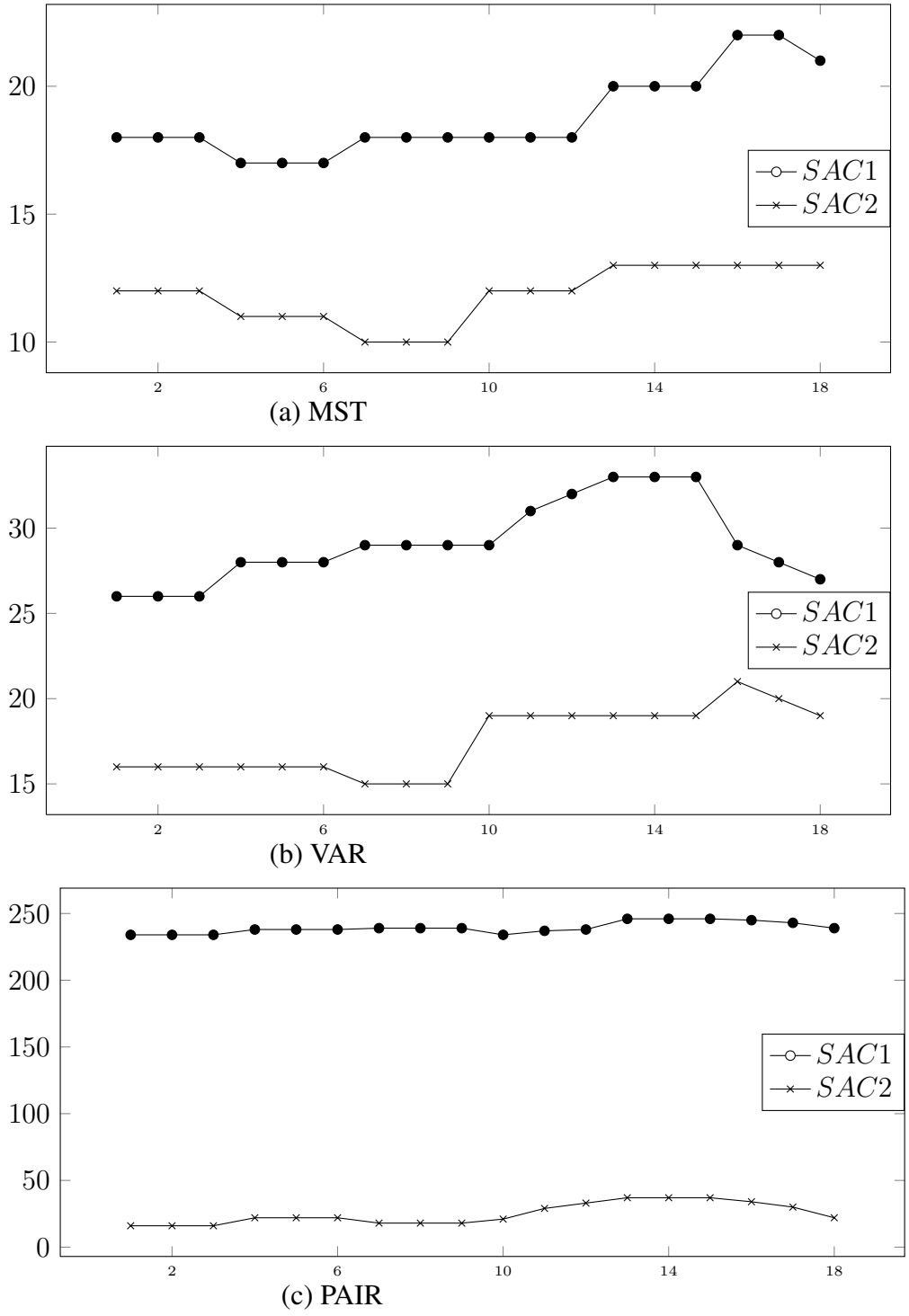


Figure 6.8: The score of the three methods (a) MST, (b) VAR, (c) PAIR on two tandem repeats from the same family in the DNA sequences of *Staphylococcus aureus*. The X axis on each plot corresponds to different parsing points relative to the motif as parsed initially. Each method shows a strong correlation between the two repeats, with both VAR and MST indicating the best parsing point in the range of 4–9, whereas PAIR has a minimum at 16.



(a) MST

(b) VAR

(c) PAIR

Figure 6.9: The score of the three methods (a) MST, (b) VAR, (c) PAIR on two tandem repeats from the same family in the DNA sequences of *Staphylococcus aureus* COL. The X axis on each plot corresponds to different parsing points. The methods show a weak correlation between the two repeats. VAR and MST show minima at points 7–9 for SAC1 and 1–3 for SAC2, so it suggests these two families may have evolved with a different parsing point.

Chapter 7

Ancestor-descendant alignment of tandemly repeated sequences

Tandem repeats and nested tandem repeats can serve as important markers for phylogenetic and population genetic studies. Comparing two tandem repeats and approximating the distance between them is a first step toward solving the problem of building phylogenies using tandem repeats.

When comparing two tandem repeats S_1 and S_2, which appear to be homologous with closely related motifs, we assume that they are descendants of some most recent common ancestral sequence A. We define the distance

$$d(S_1, S_2) := e(A, S_1) + e(A, S_2),$$

where $e(A, S_i)$ is the minimum weight evolution by which S_i evolves from A, namely, $e(A, S_i)$ is the edit distance from A to S_i under the operations of single nucleotide substitution, and duplication or deletion of entire copies of the motif. As we do not allow for insertions, e is not symmetric, but by the definition $d(S_1, S_2) = d(S_2, S_1)$.

As A will generally not be known, the problem of estimating the distance between S_1 and S_2 can be broken into two steps:

1. Estimate a possible common ancestral sequence A.

2. Estimate the edit distance from A to S_i for each i, under the operations of single nucleotide substitutions, tandem duplication and tandem deletion.

The main contribution of this chapter is a solution to the second problem, based on an asymmetric alignment algorithm. To address the first problem, we have chosen to use a Longest Common Subsequence (LCS) approach, modified to allow approximate matches between characters. We describe this in section 7.4. In solving both problems, we first replace the DNA tandem repeat sequence with a sequence whose characters represent the motif variants. This is known as an *(N)TR map*, and is described below in Section 7.1.

The problem we consider at step 2 above is an instance of what is known as the *comparison (mapping)* problem for tandem sequences (also called the alignment problem for tandem repeat sequences). Recently, a number of algorithms to address this problem under various models of tandem repeat evolution have been introduced. Benson (1997) developed exact and heuristic algorithms for comparing and aligning two tandem repeat sequences. The exact algorithm has $O(n^5)$ time complexity and $O(n^2)$ space complexity, where n is the length of the tandem repeat sequences. Algorithms to align tandem repeats under insertion, substitution, duplication and deletion of a single segment have been introduced by Behzadi and Steyaert (2003) and Berard and Rivals (2003). Their algorithms have time complexity $O(\max(m,n)^3.\rho\sigma)$ and $O(\max(m,n)^4)$ respectively, where m and n are the lengths of the two sequences, σ is the number of variants, and ρ (maximal arity) is the maximum number of single duplications in one event. A more general model of evolution, where a duplication of any size can occur (one or more adjacent copies are duplicated in one single duplication event), is considered by Sammeth and Stoye (2006). They introduced an algorithm to align tandem repeats; however, their algorithm has exponential time complexity.

7.1 TR maps

A TR *map* is a process by which a tandem repeat sequence S is replaced by a sequence S_v whose characters represent the motif variants occurring in S. Given a tandem repeat or nested tandem repeat S, let $\Sigma_v = \{a, b, \ldots, A, B, \ldots\}$ be an alphabet whose symbols represent the observed motif variants that occur in S. In the case of a nested tandem repeat, we will use lower case letters $a, b, \ldots$ to represent variants of the tandem motif, and upper case letters $A, B, \ldots$ to represent variants of the interspersed motif.

We define an (N)TR map of S to be the sequence S_v obtained by replacing each motif

variant of S by the corresponding symbol in Σ_v. This process is also known as variant mapping (Berard and Rivals, 2003).

The process of (N)TR mapping on the sequence S involves the following steps:

1. Set the boundaries of the tandem repeat motifs.
2. Build a consensus (modal) motif using the majority rules on the aligned motif copies.
3. Identify all variants of the consensus motif using the motif alignment algorithm in Chapter 5.
4. Assign a symbol from the alphabet Σ_v to each variant.
5. Replace the motif copies in S by their associated symbols.

The example on page 7 in Section 1.3 includes an example of a TR map. The set of variants is $\Sigma_v = \{a, b, c, d, e, f\}$ and the TR map is $S_v = ababccdef$.

7.2 Edit operations and edit distance

Let S_v be an (N)TR map. We define the following edit operations.

- **k-Duplication:** the process of copying a substring of length k and placing it after the duplicated segment, for example, a 2–duplication: $a(\underline{bc}) \rightarrow a(bc)(bc)$, 1–duplication $(\underline{a})bc \rightarrow (a)(a)bc$.
- **k–Deletion:** the process that removes k contiguous symbols from S_v. For example, a 1–deletion $a\underline{b}c \rightarrow ac$.
- **Variant substitution:** the process of replacing a variant a with another variant by applying one or more of the following single nucleotide events:
  - **Substitution:**

    TCGCAC$\underline{\text{A}}$GCCG $\rightarrow$ TCGCACGGCCG .
  - **Deletion:**

    TCGCACAGC$\underline{\text{T}}$G $\rightarrow$ TCGCACAGCG .

– **Insertion**:

TCGCACAG$\underline{\text{C}}$CG $\rightarrow$ TCGCACAGCACG.

Consider two tandem repeat sequences S_1 and S_2 with TR maps $(S_1)_v$ and $(S_2)_v$. Given the set of allowed edit operations listed above, and by setting weights for each edit operation, we define the edit distance between $(S_1)_v$ and $(S_2)_v$, $d((S_1)_v, (S_2)_v)$, to be the minimum weight needed to transform the ancestor A_v to both $(S_1)_v$ and $(S_2)_v$:

$$d(S_1, S_2) := e(A_v, (S_1)_v) + e(A_v, (S_2)_v),$$

where $e(A_v, (S_i)_v)$ is the minimum weight evolution by which $(S_i)_v$ evolves from A_v under the edit operations above.

7.3 Ancestor-descendant repeat distance

In this section, we introduce an algorithm that approximates the distance between two TR maps A and D, where A is hypothesized to be an ancestor and D is the observed descendant sequence.

7.3.1 The ancestor-descendant alignment problem for (N)TR sequences

Let $A = a_1 a_2 \cdots a_n$ be a string from the alphabet Σ_v. We define G_A to be a directed graph with n vertices, where the i-th vertex represents the i-th character a_i appearing in A. There is an edge from a_i to all a_j, for $1 \leq j \leq i+1$, $1 \leq i < n$, and a_n is connected to all vertices (see Figure 7.1). We can define a string S of length m as a sequence of symbols generated by a walk of length m in the directed graph G_A. A walk of length k starting at vertex a_1 and ending at vertex a_n represents a string generated from A by a number of duplications of substrings of A. Let A_d denote the set of all strings that occur as a result of walks on G_A.

We define $\vec{\delta}$=$(\delta_0, \delta_1, \ldots, \delta_n)$ to be a vector which holds the cost of a duplication, where a duplication of $a_i a_{i+1} \cdots a_j$ costs δ_{j-i}. So for example, a 1-duplication $a_i \rightarrow a_i a_i$ costs δ_0, and a 2-duplication $a_i a_{i+1} \rightarrow a_i a_{i+1} a_i a_{i+1}$ costs δ_1. We assign weights to all edges of the graph G_A as follows.

- The weight of the edge (a_i,a_{i+1}) is 0.

- The weight of the edge (a_i,a_j) is $\delta_{(i-j)}$ if $i \geq j$.

We require $\vec{\delta}$ to satisfy the following conditions:

- $\delta_j \leq \delta_i$ for all $i < j$ (longer duplications should be at least as expensive as shorter ones).
- $\delta_i + \delta_j \leq \delta_{(i+j)}$ (loosely speaking, one duplication should be no more expensive than breaking it into two smaller duplications).

An example of a vector $\vec{\delta}$ that satisfies the above conditions is $\delta_i = d+(i\times\epsilon)$, $d < 0, \epsilon < 0$.

Any string S generated by a set of duplications applied to the string A carries a cost of duplication $\Delta(S)$ equal to the minimal weight of a walk that generates the string S in the graph G_A. Our goal is to align the descendant sequence D against such a string S.

Recall the definition of an alignment in Chapter 5. Given an alphabet Σ_v, let $\bar{\Sigma}_v$ be the alphabet $\Sigma_v \cup \{-\}$, where "$-$" ("*gap*") is a character that does not belong to Σ_v. We define $\phi : \bar{\Sigma}_v^* \rightarrow \Sigma_v^*$ to be the function that deletes all gaps. Given two strings $A, D \in \Sigma_v^*$, where A represents the ancestor and D represents the descendant, an *ancestor-descendant alignment* of A and D is a choice of a pair of strings $(\bar{S}, \bar{D}) \in \bar{\Sigma}_v^* \times \bar{\Sigma}_v^*$ satisfying the following conditions:

A1. S is the result of a walk in G_A;

A2. $\phi(\bar{S}) = S$ and $\phi(\bar{D}) = D$;

A3. $|\bar{S}| = |\bar{D}|$; and

A4. there is no index i for which $\bar{S}[i] = \bar{D}[i] = -$.

Thus, $\bar{S}$ and $\bar{D}$ are obtained from S and D respectively by inserting gaps in such a way that the resulting strings have the same length, and do not both have a gap in the same position.

To score an alignment we use a scoring matrix σ, which specifies the reward or penalty for aligning any two characters of $\bar{\Sigma}_v$ against each other, and we use a duplication cost vector $\vec{\delta}$ to hold the cost of duplications. Note σ holds the cost of converting one variant to another (edit distance). In NTRs, the upper case symbols represent variants of the interspersed motif, and the lower case symbols represent variants of the tandem motif, so the edit distance between any upper case symbol and a lower case symbol is large.

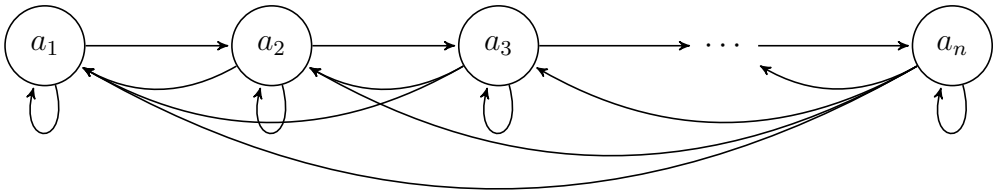


Figure 7.1: The directed graph G_A with n vertices, where the i-th vertex represents the i-th character a_i appearing in A. A walk on G starting at a_1 and ending at a_n represents a sequence derived from A by k duplications where $k = 0, 1, \cdots$.

We will assume throughout that σ penalises gaps (that is, $\sigma(-, \alpha)$ and $\sigma(\alpha, -)$ are both negative for all $\alpha \in \bar{\Sigma}_v$), and we set $\sigma(-, -) = -\infty$ to reflect condition A4 above. Given an alignment $(\bar{S}, \bar{D})$ for which $|S| = |\bar{D}| = L$, the *alignment score* of $(\bar{S}, \bar{D})$ is then defined to be

$$\sigma(\bar{S}, \bar{D}) = \sum_{i=1}^{L} \sigma(\bar{S}[i], \bar{D}[i]) + (\Delta(\bar{S})).$$

An *optimal global alignment* is an alignment of A and D which maximises the alignment score over all such alignments.

The new alignment problem can be defined as follows:

Given

1. two strings $A = a_1 a_2 \cdots a_n$ and $D = d_1 d_2 \cdots d_m$ over the alphabet Σ_v,
2. a scoring matrix σ defined over $\bar{\Sigma}_v \times \bar{\Sigma}_v$, and
3. a duplication cost vector cost $\vec{\delta} = (\delta_0, \delta_1, \ldots, \delta_n)$,

find a global optimal alignment of D against substrings of strings generated by a walk in G_A that starts at a_1 and ends at a_n.

7.3.2 Solution to the ancestor-descendant alignment problem

The ancestor-descendant alignment problem is solved by using a similar technique to the motif alignment technique of Matroud et al. (2011), presented here in Chapter 5. In this problem, there will be two matrices M and Ω. M is $(m + 1) \times (n + 1)$, and holds the score of aligning the string D against A, and Ω is of size $(m + 1) \times (n + 1)$ and holds the score after the update round.

The score matrices are filled as follows:

- Initialisation:
  - $M[i][0] = -\infty$ for all i.
  - $M[0][j] = 0$ for all j.
  - $\Omega[i][j] = 0$ for all i and j.
- We compute each column of the matrix M in two rounds. In the first round we compute $M[i][j]$ using the recursive formula

$$M[i][j] := \max \left\{ \begin{array}{c} M[i-1][j-1] + \sigma(D[i], A[j]), \\ M[i-1][j] + \sigma(D[i], -), \\ M[i][j-1] + \sigma(-, A[j]), \\ \Omega[i-1][j] + \sigma(D[i], A[j]) \end{array} \right\}.$$

  In the second round, we update $\Omega[i][j]$ with the value $\max_{j \leq k \leq n} \{M[i][k] + \delta_{(k-j)}\}$, and then update $M[i][\ell]$ for $1 \leq \ell \leq n$, using the following formula

$$\begin{aligned} M[i][\ell] &:= \max\{\max_{\ell \leq k \leq n} \{M[i][k] + \\ &\qquad \delta_{(k-\ell)} + \sigma(-, A[j])\}, M[i][\ell-1] + \sigma(-, A[j]), M[i][\ell]\}. \\ &:= \max\{\Omega[i, \ell] + \sigma(-, A[\ell])\}, M[i][\ell-1] + \sigma(-, A[\ell]), M[i][\ell]\}. \end{aligned}$$

  The pseudo-code of the algorithm is shown in Algorithm 2.

7.3.3 Correctness of the algorithm

By induction we prove that the matrix M holds the optimal alignment score of aligning D against a string $W \in A_d$.

Suppose that the sub-matrix $M_{i-1,n}$ has been correctly computed for some $i \geq 1$. That is, assume that $M[i-1][j]$ is the optimal alignment score of any alignment of the substring $D[1, i-1]$ against a string $A' \in A_d$ ending in a_j.

Consider an alignment $(\bar{W}, \bar{D})$ of $D[1, i]$ against a string $W \in A_d$. We consider three cases according to the final characters of $\bar{D}$ and $\bar{W}$.

1. If $\bar{D}$ ends in $D[i]$ and $\bar{W}$ in $A[j]$, then deleting these characters gives an alignment of $D[1, i-1]$ against a string $W' \in A_d$ ending in $A[j-1]$ or $A[k]$, where $k \geq j$.

Data: Strings A, D, a scoring matrix σ and a duplication vector cost $\vec{\delta}$
Result: Matrix M containing optimal alignment scores with respect to σ and $\vec{\delta}$ of alignments of D against substrings of strings in A_d.

for $i = 0$ **to** m=$|D|$ **do**
  $M[i][0] := -\infty$
end
for $j = 0$ **to** n=$|A|$ **do**
  $\Omega[0][j] := -\infty$
end
for $i = 1$ **to** m **do**
  for $j = 1$ **to** n **do**
    $M[i][j] := \max\{M[i-1][j-1] + \sigma(D[i], A[j]),$
    $M[i-1][j] + \sigma(D[i], -), M[i][j-1] + \sigma(-, A[j]), \Omega[i-1][j] + \sigma(D[i], A[j])\}$
  end
  for $j = 1$ **to** n **do**
    $\Omega[i][j] := \max_{j \leq k \leq n} \{M[i][k] + \delta_{(k-j)}\}$
  end
  for $j = 1$ **to** n **do**
    $M[i][j] := \max\{\Omega[i][j] + \sigma(-, A[j]), M[i][j-1] + \sigma(-, A[j]), M[i][j]\}$
  end
end

Algorithm 2: Pseudo-code for our algorithm to solve the ancestor-descendant alignment problem.

It follows that

$$\sigma\left(\bar{D}, \bar{W}\right) \leq \max(M[i-1][j-1] + \sigma(D[i], A[j]), \max_{j \leq k \leq n} \{M[i-1][k] + \delta_{(k-j)}\} + \sigma(D[i], A[j]),$$

with equality when $\bar{W}$ and $\bar{D}$ are obtained by appending $A[j]$ and $D[i]$ to an optimal alignment at $M[i-1][k]$, where $k \geq j-1$.

2. If $\bar{D}$ ends in $D[i]$ and $\bar{W}$ in a gap, then deleting these characters gives an alignment of $D[1, i-1]$ against W'. Then

$$\sigma\left(\bar{D}, \bar{W}\right) \leq M[i-1][j] + \sigma(-, D[i]),$$

with equality when $\bar{W}$ and $\bar{D}$ are obtained by appending "$-$" and $D[i]$ to an optimal alignment at $M[i-1][j]$.

3. If $\bar{D}$ ends in a gap and $\bar{W}$ ends in $A[j]$ then we may write

$$\bar{D} = \bar{D}'(-)^s,$$

where s is as large as possible. We will prove by induction on s that

$$\sigma(\bar{D}, \bar{W}) \leq M[i][j].$$

In the base case $s = 0$, we have

$$\sigma(\bar{D}, \bar{W}) = \sigma(\bar{D}'(-)^0, \bar{W}) = \sigma(\bar{D}', \bar{W}) \leq M[i][j],$$

which is already established in cases (1) and (2) above.

Let $\bar{W} = \bar{W}'A[j]$. If $\bar{W}'$ ends in $A[j-1]$ then

$$\begin{aligned}\sigma(\bar{D}, \bar{W}) &= \sigma(\bar{D}'(-)^{s-1}(-), \bar{W}'A[j]) \\ &= \sigma(\bar{D}'(-)^{s-1}, \bar{W}') + \sigma((-), A[j]).\end{aligned}$$

By our inductive assumption

$$\sigma(\bar{D}'(-)^{s-1}, \bar{W}') \leq M[i][j-1],$$

and so

$$\sigma(\bar{D}, \bar{W}) \leq M[i][j-1] + \sigma((-), A[j]) \tag{7.1}$$

$$\leq M[i][j], \tag{7.2}$$

as required. Moreover, equality occurs in (7.2) if $\bar{D}$ and $\bar{W}$ are obtained by appending $(-)$ and $A[j]$ to an optimal alignment entry at $M[i][j-1]$.

Now suppose that $\bar{W}'$ ends in $A[\ell]$ for some $\ell \geq j$. Then

$$\sigma(\bar{D}, \bar{W}) = \sigma(\bar{D}(-)^{s-1}, \bar{W}') + \delta_{l-j} + \sigma(-, A[j]).$$

By our inductive hypothesis we have

$$\sigma(\bar{D}(-)^{s-1}, \bar{W}') \leq M[i][\ell],$$

and so

$$\begin{aligned}\sigma(\bar{D}, \bar{W}) &\leq M[i][\ell] + \delta_{l-j} + \sigma(-, A[j]). \\ &\leq \max_{j \leq k \leq n} \{M[i][k] + \delta_{k-j}\} + \sigma(-, A[j]).\end{aligned}$$

Since

$$\Omega[i][j] + \sigma(-, A[j]) \leq M[i][j],$$

we will have

$$\sigma(\bar{D}, \bar{W}) \leq M[i][\ell]$$

provided we can show that

$$\Omega[i][j] = \max_{j \leq k \leq n} \{M[i][k] + \delta_{k-j}\}.$$

Since $\Omega[i][j]$ is defined by this formula at the end of the first update round, we must show that equality still holds at the end of the second round. In other words, we must show that the value of

$$\max_{j \leq k \leq n} \{M[i][k] + \delta_{k-j}\}$$

is unchanged during the second update.

If $M[i][j]$ increased during the second update, it must have taken its value

- from $M[i][j-1] + \sigma((-), \tau)$, where $\tau \in \Sigma_v$, in which case $M[i][j-1]$ must have increased too.
- or from $\max\{M[i][k] + \delta_{k-j} + \sigma((-), \tau)\}$ with $k \geq j$.

Suppose it takes its value from $M[i][k] + \delta_{k-j} + \sigma((-), \tau)$ and that $p \leq j$. To show that $\Omega[i][p]$ does not increase due to the increase in $M[i][j]$, it is enough to show

that

$$M[i][j] + \delta_{j-p} \leq M[i][k] + \delta_{k-p}.$$

Substituting $M[i][j] = M[i][k] + \delta_{k-j} + \sigma((-), \tau)$ this is equivalent to

$$M[i][k] + \delta_{k-j} + \delta_{j-p} + \sigma((-), \tau) \leq M[i][k] + \delta_{k-p},$$

which is equivalent to

$$\delta_{k-j} + \delta_{j-p} + \sigma((-), \tau) \leq \delta_{k-p}.$$

Setting $a = k - j$ and $b = j - p$, this is

$$\delta_a + \delta_b + \sigma((-), \tau) \leq \delta_{a+b},$$

and since $\sigma((-), \tau) < 0$ this holds by our assumption $\delta_a + \delta_b \leq \delta_{a+b}$.

Now suppose $M[i][j]$ takes its value from $M[i][j-1]+\sigma((-), \tau)$. Since $M[i][j-1]$ must also have increased, we may assume as an inductive hypothesis that for $p \leq j - 1$ we have

$$M[i][j - 1] + \delta_{j-1-p} \leq \Omega[i][p],$$

that is, the increase in $M[i][j - 1]$ did not increase $\Omega[i][p]$ for $p \leq j - 1$.
Suppose $q \leq j$. If $q \leq j - 1$ then

$$\begin{aligned} M[i][j] + \delta_{j-q} &= M[i][j - 1] + \delta_{j-q} + \sigma((-), \tau) \\ &= M[i][j - 1] + \delta_{j-q} + \delta_{j-q-1} - \delta_{j-q-1} + \sigma((-), \tau). \end{aligned}$$

Then we will have

$$M[i][j] + \delta_{j-q} \leq M[i][j - 1] + \delta_{j-q-1} \leq \Omega[i][q]$$

provided

$$\delta_{j-q} - \delta_{j-q-1} + \sigma((-), \tau) \leq 0.$$

Letting $a = j - q - 1$ this is equivalent to

$$\delta_{a+1} + \sigma((-), \tau) \leq \delta_a,$$

and since $\sigma((-), \tau) < 0$ this holds by our assumption $\delta_{a+1} \leq \delta_a$.

Now consider the case $q = j$. Then choose k_* such that

$$\Omega[i][j-1] = \tilde{M}[i][k_*] + \delta_{k_*-j+1},$$

where $\tilde{M}$ is the matrix M after the first update round. Then from our inductive hypothesis we have

$$M[i][j-1] + \delta_0 \leq \tilde{M}[i][k_*] + \delta_{k_*-j+1}.$$

Then

$$\begin{aligned}
M[i][j] + \delta_0 &= M[i][j-1] + \delta_0 + \sigma((-), \tau) \\
&\leq \tilde{M}[i][k_*] + \delta_{k_*-j+1} + \sigma((-), \tau) \\
&= \tilde{M}[i][k_*] + \delta_{k_*-j} - \delta_{k_*-j} + \delta_{k_*-j+1} + \sigma((-), \tau) \\
&\leq \tilde{M}[i][k_*] + \delta_{k_*-j} \\
&\leq \Omega[i][j],
\end{aligned}$$

where we have used $\sigma((-), \tau) < 0$ and $\delta_{k_*-j+1} \leq \delta_{k_*-j}$.

This completes the proof that $\Omega[i][j]$ does not increase after the second update, which completes the proof of correctness of the ancestor-descendant alignment algorithm.

7.4 A Longest Common Subsequence approach to estimating the most recent common ancestor

In this section, we introduce a modification of the Longest Common Subsequence (LCS) algorithm, as one approach to the problem in step 1, of estimating the most recent common ancestor A of two closely related tandem repeats. We construct the ancestral sequence $A_v = LCS((S_1)_v, (S_2)_v)$ using the known dynamic programming technique (Gusfield, 1997). The dynamic programming matrix L has size $|(S_1)_v| \times |(S_2)_v|$. The scoring matrix σ specifies the reward or penalty for aligning any two characters of $\bar{\Sigma}_v$ against each other. The matrix L is filled using the following formula:

if $\sigma((S_1)_v[i], (S_2)_v[j]) \leq 2$ then

$$L[i][j] := L[i-1][j-1] + 1,$$

else

$$L[i][j] := \max \left\{ \begin{array}{c} L[i-1][j], \\ L[i][j-1] \end{array} \right\}.$$

The backtracking procedure is modified to accommodate mismatches of edit distance at most 2. If two characters at position i and j are mismatched and the distance between them is less than or equal 2 then the LCS will contain their lowest common ancestor. The lowest common ancestor problem of two variants x and y in a tree is the problem of finding the vertex closest to x and y that appears in both shortest paths of x and y to the "ancestral" vertex. The lowest common ancestor can be pre-calculated from a minimum spanning tree of the variants graph in Section 3.3. There are a number of algorithms constructed to solve the lowest common ancestor problem (Aho et al., 1976; Harel and Tarjan, 1984).

7.5 An application to real DNA sequences

We have applied our algorithm on JP1 and NZ1. The ancestral sequence of both JP1 and NZ1 is constructed using a modification of the longest common subsequence algorithm,

as described in Section 7.4.

The duplication penalty function that is used in this test is $\delta_i = d + (i \times \epsilon)$, where $d = -10,\ \epsilon = -1$. The match function between two variants is calculated using the edit distance between them. If $a \neq b$ then σ(a,b) = edit distance(a,b)$\times(-10)$, and if $a = b$ then σ(a,b) = 40. The gap penalty is $\sigma(\alpha, -) = \sigma(-, \alpha) = -40$. The output of our program is shown in Figure 7.2.

7.6 Conclusion

This chapter addresses the problem of aligning two closely related tandem repeat sequences. The model we have considered accommodates block duplications, deletions and substitutions. The algorithm we have introduced has $O(m \times n)$ time and space complexity, where m and n are the lengths of the two compared sequences. There are limitations in using our approach, such as the fact that the duplications are considered before any substitutions. For example, when aligning aa against $abab$ our algorithm will give the score of two substitutions and a 2-duplication, where the maximum parsimony history is $aa \rightarrow (ab) \rightarrow (ab)(ab)$. However, this problem can be overcome by aligning the ancestor sequence against its descendant twice. The ancestor can be corrected from the first alignment and the corrected ancestor can then be aligned against the descendant sequence.

```
JP1  = AacaAaAabkfddAabcakfdAjebecbcbBcafaAababecababBgaaaaag
       qaddaAdbiabbbmabBahaaaaaacaaafohaAarababbbapjvlBggaa
NZ1  = AacaAaAabkfddAabeebecbcbBacafaAababecababBgaa
       qaddaAdbiiaiabbbmabBahaaaacaaafohaAaaabaabbapjvlBggaa

LCAS = AacaAaAabkfddAabebecbcbBcafaAababecababBgaa
       qaddaAdbiabbbmabBahaaaacaaafohaAaababbapjvlBggaa
```

```
LCAS= AacaAa(Aabkfdd)                AabebecbcbBcafaAababecababBg(aa)
An  = AacaAa(Aabkfdd)(Aa(b-)(bk)fdd)AabebecbcbBcafaAababecababBg(aa)(aa)(aa)
      |||||| |||||||  || |   *| |   |* |||||||||||||||||||||||| ||  ||  |*
JP1 = AacaAa Aabkfdd  Aa bc  ak fd- Aj-ebecbcbBcafaAababecababBg aa  aa  ag

LCAS= qaddaAdbiabbbmabBahaa(aa)    caaafohaAa(a)   bab(b)   apjvlBggaa
An  = qaddaAdbiabbbmabBahaa(aa)(aa)caaafohaAa(a)(a)bab(b)(b)apjvlBggaa
      |||||||||||||||||||| ||  || |||||||||| *  | ||| |  | |||||||||
JP1 = qaddaAdbiabbbmabBahaa aa  aa caaafohaAa r  a bab b  b apjvlBggaa

e(LCAS,JP1)=14764
```

```
LCAS= AacaAaAabkfddAab(e)   becbcbB(ca)    faAababecababBgaa
An  = AacaAaAabkfddAab(e)(e)becbcbB(ca)(ca)faAababecababBgaa
      ||||||||||||||| |  | ||||||| ||  || ||||||||||||||||
NZ1 = AacaAaAabkfddAab e  e becbcbB -a  ca faAababecababBgaa

LCAS= qaddaAdb   ((i)a)    bbbmabBahaaaacaaafohaAa(a)   b(a)   bbapjvlBggaa
An  = qaddaAdb((i)(i)a)(ia)bbbmabBahaaaacaaafohaAa(a)(a)b(a)(a)bbapjvlBggaa
      ||||||||  |  | |  || |||||||||||||||||||||||| |  | | |  | ||||||||||||
NZ1 = qaddaAdb  i  i a  ia bbbmabBahaaaacaaafohaAa a  a b a  a bbapjvlBggaa
e(LCAS,JP1)=14532
```

Figure 7.2: The comparison of NZ1 and JP1. Characters that have been duplicated are enclosed in round brackets. The character "|" indicates a match, the character "*" indicates a mismatch, and the character "−" represents a gap. The number $e(LCAS, JP1) = 14764$ is the score of aligning the longest common approximate subsequence of JP1 and NZ1 against JP1, and the number $e(LCAS, NZ1) = 14532$ is the score of aligning LCAS against NZ1. Thus the estimated distance $d(JP1, NZ1)$ is $14764 + 14532 = 29296$. Note that the two scores $e(LCAS, JP1)$ and $e(LCAS, NZ1)$ are approximately equal to each other, as we would typically expect of the distances from two sequences to their most recent common ancestor.

Chapter 8

Conclusion

In this project, nested tandem repeats have been investigated. A close look at the taro nested tandem repeat structure leads to a number of open problems that are of interest to the computational biology community. Some of the questions that were addressed in this thesis are not specific to nested tandem repeat structures but can also be raised in other repeated structures. A summary of the key outcomes of this project is listed below:

- The main goal of Chapter 3 is to analyse the nested tandem repeats found in taro and illustrate some observations. A number of issues have been addressed in this chapter, such as the number of expected substitutions in a DHT and the variants distribution.

- Once the nested tandem repeat structure in the IGS region of the rDNA of taro was observed, the question of the ubiquitousness of such structures in DNA arises. To address this issue, a software tool (NTRFinder) was constructed to search for nested tandem repeats in DNA. This tool was presented in Chapter 4. NTRfinder consists of two major components (detecting the signal of an NTR and aligning an NTR against its two motifs). As a result, nested tandem repeat have been found in many locations across species and across chromosomes. One interesting observation is that a number of nested tandem repeat structures were found in the human Y chromosome, and they are all found in the pseudoautosomal region. On the other hand, NTRFinder did not find any nested tandem repeats in other parts of the Y chromosome. It was found that some plant rDNA IGS regions contain nested tandem repeat structures. Nested tandem repeats in different plants do not appear to have significant similarity.

- The problem of aligning two motifs against a nested tandem repeat was addressed in Chapter 5. A solution using a dynamic programming technique that guarantees optimality was introduced. This algorithm was implemented as part of the software NTRFinder, as the verification component.

- In Chapter 6, the question of determining the boundaries of the motifs in a tandem or nested tandem repeat is addressed. This problem has not been given much attention in the literature to date. We have presented three heuristic criteria to help in determining the boundaries in both tandem repeats and nested tandem repeats. When there are a small number of copies and/or the copies are identical these criteria are not of a great help in determining the parsing points.

- Chapter 7 addresses a specific segment alignment problem relating to comparing two tandem or nested tandem repeats. An algorithm that guarantees optimality is introduced. This algorithm aligns a hypothesized ancestor sequence against its descendant sequence (the observed sequence). It is an asymmetric alignment. This algorithm can be used to estimate the distance between two closely related repeated sequences.

8.1 Future work

The early migration of people in the Pacific has been studied using genomic data from several different organisms (Storey et al., 2013). The work in this thesis can be continued on to study the population genetics of the taro plant. With the advances in sequencing technology, it has become feasible to sequence a large number of taro IGS sequences from the Pacific region. Studying the distribution of taro in the Pacific is an interesting goal that will further enable us to understand the migration of people in this region.

Another important direction that this project can progress is the study of the concerted evolution that is believed to occur in the rDNA of some organisms. rDNA genes exist in arrays of thousands of copies in some genomes. The relationship between these copies can provide hints about the mechanism of evolution of the rDNA copies in the one genome. Nested tandem repeat structures that exist in the IGS of rDNA of some organism can be used as a marker to infer the phylogeny of rDNA genes. Once third generation sequencing

technology (single molecule sequencing) is ready to be used, it will be interesting to proceed with a project that addresses the phylogeny of rDNA genes.

Appendix A

Published chapters

A list of published papers are listed along with the front page of each paper as it appears in the journal/proceeding.

For the purpose of consistency, background and definitions that appear in more than one chapter were moved to Chapter 1.

- The text in Chapter 4 is a modification of the paper titled "NTRFinder: a software tool to find nested tandem repeats". The section "Sequences, edit operations and the edit distance" and section "Classification of tandem repeats", were moved to Chapter 1.

- Chapter 5 contains the material of the paper "An algorithm to solve the motif alignment problem for approximate nested tandem repeats in biological sequences".

- Chapter 6 contains an extended version of the text of the paper "A comparison of three heuristic methods for solving the parsing problem for tandem repeats". This extended version was submitted to the journal of IEEE/ACM Transactions on Computational Biology and Bioinformatics, and I have also incorporated a number of changes suggested by the referees of that journal.

The completed DRC16 forms "Statement of contribution to doctoral thesis containing publications", together with the front pages of the published papers are shown in the next pages.

Nucleic Acids Research Advance Access published November 25, 2011

Nucleic Acids Research, 2011, 1–6
doi:10.1093/nar/gkr1070

NTRFinder: a software tool to find nested tandem repeats

Atheer A. Matroud[1,2,*], M. D. Hendy[3] and C. P. Tuffley[1]

[1]Institute of Fundamental Sciences, [2]Allan Wilson Centre for Molecular Ecology and Evolution, Massey University, Private Bag 11 222, Palmerston North 4442 and [3]Department of Mathematics and Statistics, University of Otago, PO Box 56, Dunedin 9054, New Zealand

Received April 12, 2011; Revised October 3, 2011; Accepted October 28, 2011

ABSTRACT

We introduce the software tool `NTRFinder` to search for a complex repetitive structure in DNA we call a nested tandem repeat (NTR). An NTR is a recurrence of two or more distinct tandem motifs interspersed with each other. We propose that NTRs can be used as phylogenetic and population markers. We have tested our algorithm on both real and simulated data, and present some real NTRs of interest. `NTRFinder` can be downloaded from http://www.maths.otago.ac.nz/~aamatroud/.

INTRODUCTION

Genomic DNA has long been known to contain 'tandem repeats': repetitive structures in which many approximate copies of a common segment (the 'motif') appear consecutively. Several studies have proposed different mechanisms for the occurrence of tandem repeats (1,2), but their biological role is not well understood.

Recently, we have observed a more complex repetitive structure in the ribosomal DNA of *Colocasia esculenta* (taro), consisting of multiple approximate copies of two distinct motifs interspersed with one another. We call such structures nested tandem repeats (NTRs), and the problem of finding them in sequence data is the focus of this article. Our motivation is their potential use for studying populations: for example, a preliminary analysis suggests that changes in the NTR in taro have been occurring on a 1000 year time scale, so a greater understanding of this NTR offers the potential to date the early agriculture of this ancient staple food crop.

The problem of locating tandem repeats is well known, as their implication for neurological disorders (3,4), and their use to infer evolutionary histories has urged some researchers to develop tools to find them. This has resulted in a number of software tools, each of which has its own strengths and limitations. However, the existing tools were not designed to find NTRs, and consequently do not generally find them. In this article, we present a new software tool, `NTRFinder`, which is designed to find these more complex repetitive structures.

We report here the algorithm on which `NTRFinder` is based and report some of the NTRs it has identified, including an even more complex structure where copies of four distinct motifs are interspersed.

Sequences, edit operations and the edit distance

A DNA sequence is a sequence of symbols from the nucleotide alphabet $\Sigma = \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$. We define a DNA segment to be a string of contiguous DNA nucleotides and define a site to be a component in a segment. For a DNA segment

$$\mathbf{X} = x_1 x_2 \cdots x_n,$$

$x_i \in \Sigma$ is the nucleotide at the i-th site and $|\mathbf{X}| = n$ is the length of $\mathbf{X}$.

Copying errors happen in DNA replication due to different external and internal factors. These changes include substitution, insertion, deletion, duplication and contraction. We refer to these as 'edit operations'. By giving each type of edit operation some specific weight, we can in principle find a series of edit operations which transforms segment x to segment y, whose sum of weights is minimal. We will refer to this sum as the 'edit distance', and denote it by $d(x, y)$. For the purposes of this article, the edit operations allowed in calculating the edit distance are restricted to single nucleotide substitutions, and single nucleotide insertions or deletions (indels).

Classification of tandem repeats

Many classifications of tandem repeat schemas have been introduced in the computational biology literature. We list some which are commonly used:

- (Exact) tandem repeats: an 'exact tandem repeat' (TR) is a sequence comprising two or more contiguous copies $\mathbf{XX}\ldots\mathbf{X}$ of identical segments $\mathbf{X}$ (referred to as the *motif*).
- k–Approximate tandem repeats: a *k–approximate tandem repeat* (k–TR) is a sequence comprising two

*To whom correspondence should be addressed. Tel: +6434797989; Fax: +6434798427; Email: a.a.matroud@massey.ac.nz

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION TO DOCTORAL THESIS CONTAINING PUBLICATIONS

(To appear at the end of each thesis chapter/section/appendix submitted as an article/paper or collected as an appendix at the end of the thesis)

We, the candidate and the candidate's Principal 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: Atheer Matroud

Name/Title of Principal Supervisor: Dr. Christopher Tuffley

Name of Published Research Output and full reference:

NTRFinder: a software tool to find nested tandem repeats.
Nucleic Acids Research, 2011, 1–6, doi:10.1093/nar/gkr1070.

In which Chapter is the Published Work: 4

Please indicate either:

- The percentage of the Published Work that was contributed by the candidate:

  and / or

- Describe the contribution that the candidate has made to the Published Work:

  Developed and implemented the software; carried out all tests on simulated and real data; wrote the first draft of the paper.

Atheer — Digitally signed by Atheer, DN: cn=Atheer, o=Massey University, ou=IFS, email=a.a.matroud@massey.ac.nz, c=NZ, Date: 2013.06.19 10:50:12 +12'00'

Candidate's Signature — 19/6/2013 — Date

Christopher Tuffley — Digitally signed by Christopher Tuffley, DN: cn=Christopher Tuffley, o=Massey University, ou=Institute of Fundamental Sciences, email=c.tuffley@massey.ac.nz, c=NZ, Date: 2013.06.19 12:01:26 +12'00'

Principal Supervisor's signature — 19/6/2013 — Date

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 18, Number 9, 2011

Pp. 1211–1218
DOI: 10.1089/cmb.2011.0101

An Algorithm to Solve the Motif Alignment Problem for Approximate Nested Tandem Repeats in Biological Sequences

ATHEER A. MATROUD,[1,2] CHRISTOPHER P. TUFFLEY,[2] and MICHAEL D. HENDY[3]

ABSTRACT

An *approximate nested tandem repeat* (NTR) in a string T is a complex repetitive structure consisting of many approximate copies of two substrings x and X ("motifs") interspersed with one another. NTRs fall into a class of repetitive structures broadly known as *subrepeats*. NTRs have been found in real DNA sequences and are expected to be important in evolutionary biology, both in understanding evolution of the ribosomal DNA (where NTRs can occur), and as a potential marker in population genetic and phylogenetic studies. This article describes an alignment algorithm for the verification phase of the software tool NTRFinder developed for database searches for NTRs. When the search algorithm has located a subsequence containing a possible NTR, with motifs X and x, a verification step aligns this subsequence against an exact NTR built from the templates X and x, to determine whether the subsequence contains an approximate NTR and its extent. This article describes an algorithm to solve this alignment problem in $O(|\mathbf{T}|(|\mathbf{X}| + |\mathbf{x}|))$ space and time. The algorithm is based on Fischetti et al.'s wrap-around dynamic programming.

Key words: algorithms, alignment, molecular evolution, satellites, simple sequence repeats.

1. INTRODUCTION

An *approximate nested tandem repeat* (NTR) in a string **T** is a complex repetitive structure consisting of many approximate copies of two substrings **x** and **X** (''motifs'') interspersed with one another. The name derives from the fact that an NTR may be thought of as two tandem repeats nested within one another.

Approximate nested tandem repeats have been found in real DNA sequences, such as that of *Colocasia esculenta*, the ancient staple food crop taro (Matroud et al., 2011). The intergenic spacer (IGS) region in the taro ribosomal DNA contains an NTR consisting of eleven approximate copies of a 48-bp motif, interspersed within a tandem repeat consisting of 96 approximate copies of an 11-bp motif. The NTR found in taro, used as a genetic marker, offers the potential to elucidate the prehistory of the early agriculture of this

[1]Allan Wilson Centre for Molecular Ecology and Evolution, New Zealand.
[2]Institute of Fundamental Sciences, Massey University, Palmerston North, New Zealand.
[3]Department of Mathematics and Statistics, University of Otago, Dunedin, New Zealand.

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION TO DOCTORAL THESIS CONTAINING PUBLICATIONS

(To appear at the end of each thesis chapter/section/appendix submitted as an article/paper or collected as an appendix at the end of the thesis)

We, the candidate and the candidate's Principal 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: Atheer Matroud

Name/Title of Principal Supervisor: Dr. Christopher Tuffley

Name of Published Research Output and full reference:

An Algorithm to Solve the Motif Alignment Problem for Approximate Nested Tandem Repeats in Biological Sequences.
JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 18, Number 9, 2011, pp. 1211–1218,
DOI: 10.1089/cmb.2011.0101

In which Chapter is the Published Work: 5

Please indicate either:

- The percentage of the Published Work that was contributed by the candidate:

  and / or

- Describe the contribution that the candidate has made to the Published Work:

  Developed and implemented the algorithm; carried out all tests on simulated and real data; wrote the first draft of the paper.

Atheer
Digitally signed by Atheer
DN: cn=Atheer, o=Massey University, ou=IFS, email=a.a.matroud@massey.ac.nz, c=NZ
Date: 2013.06.19 10:50:12 +12'00'

Candidate's Signature

19/6/2013
Date

Christopher Tuffley
Digitally signed by Christopher Tuffley
DN: cn=Christopher Tuffley, o=Massey University, ou=Institute of Fundamental Sciences, email=c.tuffley@massey.ac.nz, c=NZ
Date: 2013.06.19 12:03:28 +12'00'

Principal Supervisor's signature

19/6/2013
Date

A Comparison of Three Heuristic Methods for Solving the Parsing Problem for Tandem Repeats

A.A. Matroud[1,3], C.P. Tuffley[1], D. Bryant[2,3], and M.D. Hendy[2]

[1] Institute of Fundamental Sciences, Massey University, Private Bag 11222, Palmerston North, New Zealand
[2] Department of Mathematics and Statistics, University of Otago, Dunedin, New Zealand
[3] Allan Wilson Centre for Molecular Ecology and Evolution

Abstract. In many applications of tandem repeats the outcome depends critically on the choice of boundaries (beginning and end) of the repeated motif: for example, different choices of pattern boundaries can lead to different duplication history trees. However, the best choice of boundaries or *parsing* of the tandem repeat is often ambiguous, as the flanking regions before and after the tandem repeat often contain partial approximate copies of the motif, making it difficult to determine where the tandem repeat (and hence the motif) begins and ends. We define the *parsing problem* for tandem repeats to be the problem of discriminating among the possible choices of parsing.

In this paper we propose and compare three heuristic methods for solving the parsing problem, under the assumption that the parsing is fixed throughout the duplication history of the tandem repeat. The three methods are PAIR, which minimises the number of pairs of common adjacent mutations which span a boundary; VAR, which minimises the total number of variants of the motif; and MST, which minimises the length of the minimum spanning tree connecting the variants, where the weight of each edge is the Hamming distance of the pair of variants. We test the methods on simulated data over a range of motif lengths and relative rates of substitutions to duplications, and show that all three perform better than choosing the parsing arbitrarily. Of the three MST typically performs the best, followed by VAR then PAIR.

1 Introduction

Genomic DNA has long been known to contain *tandem repeats:* repetitive structures in which many approximate copies of a common segment (the *motif*) appear consecutively. The copies of the motif are usually polymorphic, which makes tandem repeats a useful tool for phylogenetics and for inter-population studies (Rivals [15]); in addition, highly polymorphic tandem repeats can be used to discriminate among individuals within a population, and have proved to be useful for DNA fingerprint techniques (Jeffreys et al. [11]). Because of this, many

M.C.P. de Souto and M.G. Kann (Eds.): BSB 2012, LNBI 7409, pp. 37–48, 2012.

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION TO DOCTORAL THESIS CONTAINING PUBLICATIONS

(To appear at the end of each thesis chapter/section/appendix submitted as an article/paper or collected as an appendix at the end of the thesis)

We, the candidate and the candidate's Principal 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: Atheer Matroud

Name/Title of Principal Supervisor: Dr. Christopher Tuffley

Name of Published Research Output and full reference:

A Comparison of Three Heuristic Methods for Solving the Parsing Problem for Tandem Repeats.
Advances in Bioinformatics and Computational Biology, LNBI 7409, 37-48, 2012

In which Chapter is the Published Work: 6

Please indicate either:

- The percentage of the Published Work that was contributed by the candidate:

  and / or

- Describe the contribution that the candidate has made to the Published Work:

  Developed two of the three criteria and implemented all three of them; carried out all tests on simulated and real data; wrote the description of two of the three criteria and wrote first draft of the rest of the paper.

Atheer — Digitally signed by Atheer DN: cn=Atheer, o=Massey University, ou=IFS, email=a.a.matroud@massey.ac.nz, c=NZ Date: 2013.06.19 10:50:12 +12'00'

Candidate's Signature — 19/6/2013 — Date

Christopher Tuffley — Digitally signed by Christopher Tuffley DN: cn=Christopher Tuffley, o=Massey University, ou=Institute of Fundamental Sciences, email=c.tuffley@massey.ac.nz, c=NZ Date: 2013.06.19 12:05:41 +12'00'

Principal Supervisor's signature — 19/6/2013 — Date

Bibliography

The numbers following each reference indicate the pages on which it is cited.

Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. On finding lowest common ancestors in trees. *SIAM Journal on computing*, 5(1):115–132, 1976. 83

A. Apostolico and F. P. Preparata. Optimal off-line detection of repetitions in a string. *Theor. Comput. Sci.*, 22:297–315, 1983. 13

J.A.L. Armour, T. Anttinen, C.A. May, E.E. Vega, A. Sajantila, J.R. Kidd, K.K. Kidd, J. Bertranpetit, S. Paabo, and A.J. Jeffreys. Minisatellite diversity supports a recent African origin for modern humans. *Nature Genetics*, 13(2):154–160, 1996. 2

B. Behzadi and J. Steyaert. An improved algorithm for generalized comparison of minisatellites. In Ricardo Baeza-Yates, Edgar Chvez, and Maxime Crochemore, editors, *Combinatorial Pattern Matching*, volume 2676 of *Lecture Notes in Computer Science*, pages 32–41. Springer Berlin / Heidelberg, 2003. ISBN 978-3-540-40311-1. 16, 52, 72

G. Benson. Tandem repeats finder: a program to analyze DNA sequences. *Nucl. Acids Res.*, 27(2):573–580, 1999. doi: 10.1093/nar/27.2.573. 13, 53

G. Benson and L. Dong. Reconstructing the duplication history of a tandem repeat. In *Proceedings International Conference on Intelligent Systems for Molecular Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology*, page 44, 1999a. 13

Gary Benson. Sequence alignment with tandem duplication. *Journal of Computational Biology*, 4(3):351–367, 1997. 72

Gary Benson and Lan Dong. Reconstructing the duplication history of a tandem repeat. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 44–53. AAAI Press, 1999b. ISBN 1-57735-083-9. URL `http://portal.acm.org/citation.cfm?id=645634.660817`. 15, 53

S. Berard and E. Rivals. Comparison of minisatelites. *Journal of Computational Biology*, 10(3-4):357–372, 2003. 6, 7, 13, 16, 52, 72, 73

D. Bertrand, M. Lajoie, and N. El-Mabrouk. Inferring ancestral gene orders for a family of tandemly arrayed genes. *Journal of Computational Biology*, 15(8):1063–1077, 2008. 52

P Bois and AJ Jeffreys. Minisatellite instability and germline mutation. *Cellular and Molecular Life Sciences CMLS*, 55(12):1636–1648, 1999. 12

C Richard Boland and Ajay Goel. Microsatellite instability in colorectal cancer. *Gastroenterology*, 138(6):2073–2087, 2010. 11

J. Buard and A.J. Jeffreys. Big, bad minisatellites. *Nature Genetics*, 15(4):327–328, 1997. 1, 4, 12

C. Chauve, J. P. Doyon, and N. El-Mabrouk. Gene family evolution by duplication, speciation, and loss. *Journal of Computational Biology*, 15(8):1043–1062, 2008. 52

M. Crochemore. An optimal algorithm for computing the repetitions in a word. *Inf. Process. Lett.*, 12(5):244–250, 1981. 13, 53

O. Delgrange and E. Rivals. Star: an algorithm to search for tandem approximate repeats. *Bioinformatics*, 20(16):2812–2820, 2004. 13

F. Denœud and G. Vergnaud. Identification of polymorphic tandem repeats by direct comparison of genome sequence from different bacterial strains: a web-based resource. *BMC bioinformatics*, 5(1):4, 2004. 54, 65

Colette Dib, Sabine Fauré, Cécile Fizames, Delphine Samson, Nathalie Drouot, Alain Vignal, Philippe Millasseau, Sophie Marc, Jamile Kazan, Eric Seboun, et al. A comprehensive genetic map of the human genome based on 5,264 microsatellites. *Nature*, 380(6570):152–154, 1996. 11

N. O. Domaniç and F. P. Preparata. A novel approach to the detection of genomic approximate tandem repeats in the Levenshtein metric. *Journal of Computational Biology*, 14 (7):873–891, 2007. 13, 14

V.A. Fischetti, G.M. Landau, P.H. Sellers, and J.P. Schmidt. Identifying periodic occurrences of a template with applications to protein structure. *Inform. Process. Lett.*, 45: 11–18, 1993. 15, 40, 41, 44, 45, 54

Walter M. Fitch. Phylogenies constrained by the crossover process as illustrated by human hemoglobins and a thirteen-cycle, eleven-amino-acid repeat in human apolipoprotein a-i. *Genetics*, 86(3):623–644, 1977a. URL `http://www.genetics.org/content/86/3/623.abstract`. 53

W.M. Fitch. Phylogenies constrained by the crossover process as illustrated by human hemoglobins and a thirteen-cycle, eleven-amino-acid repeat in human apolipoprotein ai. *Genetics*, 86(3):623–644, 1977b. 13

L.R. Foulds and R.L. Graham. The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics*, 3(1):43–49, 1982. 23, 53

YH Fu, A Pizzuti, Jr Fenwick, RG, J King, S Rajnarayan, PW Dunne, J Dubel, GA Nasser, T Ashizawa, P de Jong, and et al. An unstable triplet repeat in a gene related to myotonic muscular dystrophy. *Science*, 255(5049):1256–1258, 1992. doi: 10.1126/science.1546326. URL `http://www.sciencemag.org/cgi/content/abstract/255/5049/1256`. 1, 28

Pauline A. Fujita, Brooke Rhead, Ann S. Zweig, Angie S. Hinrichs, Donna Karolchik, Melissa S. Cline, Mary Goldman, Galt P. Barber, Hiram Clawson, Antonio Coelho, Mark Diekhans, Timothy R. Dreszer, Belinda M. Giardine, Rachel A. Harte, Jennifer Hillman-Jackson, Fan Hsu, Vanessa Kirkup, Robert M. Kuhn, Katrina Learned, Chin H. Li, Laurence R. Meyer, Andy Pohl, Brian J. Raney, Kate R. Rosenbloom, Kayla E. Smith, David Haussler, and W. James Kent. The UCSC genome browser database: update 2011. *Nucleic Acids Research*, 2010. doi: 10.1093/nar/gkq963. URL `http://nar.oxfordjournals.org/content/early/2010/10/18/nar.gkq963.abstract`. 32

O. Gascuel, M.D. Hendy, A. Jean-Marie, and R. McLachlan. The combinatorics of tandem duplication trees. *Systematic Biology*, 52(1):110–118, 2003. 8, 53, 57

David B Goldstein and Christian Schlotterer. Microsatellites: evolution and applications. 1999. 12

D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge Univ. Pr., 1997. 15, 83

Dov Harel and Robert Endre Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM Journal on Computing*, 13(2):338–355, 1984. 83

Frederick T Hatch, Anne J Bodner, Joseph A Mazrimas, and Dan H Moore Jr. Satellite dna and cytogenetic evolution. *Chromosoma*, 58(2):155–168, 1976. 1, 11

Amy M. Hauth and Deborah A. Joseph. Beyond tandem repeats: complex pattern structures and distant regions of similarity. *Bioinformatics*, 18(suppl 1):S31–S37, 2002. doi: 10.1093/bioinformatics/18.suppl_1.S31. URL `http://bioinformatics.oxfordjournals.org/content/18/suppl_1/S31.abstract`. 1, 4, 11, 13, 14, 53

MD Hendy, LR Foulds, and D. Penny. Proving phylogenetic trees minimal with l-clustering and set partitioning. *Mathematical Biosciences*, 51(1):71–88, 1980. 22, 23

Philippe Jarne and Pierre JL Lagoda. Microsatellites, from molecules to populations and back. *Trends in Ecology & Evolution*, 11(10):424–429, 1996. 4

A.J. Jeffreys, V. Wilson, and S.L. Thein. Individual-specific fingerprints of human DNA. *Nature.*, 51(2):71–88, 1980. 52

A.J. Jeffreys, A. MacLeod, K. Tamaki, D.L. Neil, D.G. Monckton, et al. Minisatellite repeat coding as a digital approach to dna typing. *Nature*, 354(6350):204–209, 1991. 12

P Keim, LB Price, AM Klevytska, KL Smith, JM Schupp, R Okinaka, PJ Jackson, and ME Hugh-Jones. Multiple-locus variable-number tandem repeat analysis reveals genetic relationships within bacillus anthracis. *Journal of Bacteriology*, 182(10):2928–2936, 2000. 15

M Kimura. Estimation of evolutionary distances between homologous nucleotide sequences. *Proceedings of the National Academy of Sciences*, 78(1):454–458, 1981. 3

Alexandra M Klevytska, Lance B Price, James M Schupp, Patricia L Worsham, Jane Wong, and Paul Keim. Identification and characterization of variable-number tandem repeats in the yersinia pestis genome. *Journal of clinical microbiology*, 39(9):3179–3185, 2001. 15

R. Kolpakov, Gregory Kucherov, and T. G. Logiciel. Finding approximate repetitions under Hamming distance. In *Theoretical Computer Science*, pages 170–181. Springer, 2001. 13

M. Lajoie, D. Bertrand, N. El-Mabrouk, and O. Gascuel. Duplication and inversion history of a tandemly repeated genes family. *Journal of Computational Biology*, 14(4): 462–478, 2007. 52

G. M. Landau, J. P. Schmidt, and D. Sokol. An algorithm for approximate tandem repeats. *Journal of Computational Biology*, 8(1):1–18, 2001. 13

Gene Levinson and George A Gutman. Slipped-strand mispairing: a major mechanism for dna sequence evolution. *Molecular biology and evolution*, 4(3):203–221, 1987. 12

M. G. Main and R. J. Lorentz. An $o(n\ log\ n)$ algorithm for finding all repetitions in a string. *J. Algorithms*, 5(3):422–432, 1984. 13

A. Matroud, C. Tuffley, D. Bryant, and M. Hendy. A comparison of three heuristic methods for solving the parsing problem for tandem repeats. *Advances in Bioinformatics and Computational Biology*, pages 37–48, 2012a. 51

A. A. Matroud, M. D. Hendy, and C. P. Tuffley. An algorithm to solve the motif alignment problem for approximate nested tandem repeats in biological sequences. *Journal of Computational Biology*, 18(9):1211–1218, 2011. 30, 39, 54, 76

A.A. Matroud, M. D. Hendy, and C. P. Tuffley. NTRFinder: a software tool to find nested tandem repeats. *Nucleic Acids Research*, 40(3):e17–e17, 2012b. 27, 40, 41, 50, 53

P. Matthews. A possible tropical wildtype taro: Colocasia esculenta var. aquatilis. *Bulletin of the Indo-Pacific Prehistory Association*, 11:69–81, 1991. 1, 17

P. Matthews, Y. Matsushita, T. Sato, and M. Hirai. Ribosomal and mitochondrial DNA variation in Japanese taro (Colocasia esculenta L. Schott). *Japanese Journal of Breeding*, 42, 1992. 11, 17

G. Navarro. A guided tour to approximate string matching. *ACM. Comp. Surv.*, 33:2001, 1999. 15, 43

S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453, 1970. 15

A. Newman and J. Cooper. Xstream: A practical algorithm for identification and architecture modeling of tandem repeats in protein sequences. *BMC Bioinformatics*, 8(1):382, 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-382. URL `http://www.biomedcentral.com/1471-2105/8/382`. 1, 11

Raisa Nikula, Hamish G Spencer, and Jonathan M Waters. Comparison of population-genetic structuring in congeneric kelp-versus rock-associated snails: a test of a dispersal-by-rafting hypothesis. *Ecology and evolution*, 1(2):169–180, 2011. 11

David C Queller, Joan E Strassmann, and Colin R Hughes. Microsatellites and kinship. *Trends in Ecology & Evolution*, 8(8):285–288, 1993. 11

E. Rivals. A survey on algorithmic aspects of tandem repeats evolution. *Int. J. Found. Comput. Sci.*, 15(2):225–257, 2004a. 52

E. Rivals. A survey on algorithmic aspects of tandem repeats evolution. *International Journal of Foundations of Computer Science*, 15(2):225–257, 2004b. 7

Thomas Rolland, Bernard Dujon, and Guy-Franck Richard. Dynamic evolution of megasatellites in yeasts. *Nucleic Acids Research*, 38(14):4731–4739, 2010. doi: 10.1093/nar/gkq207. URL `http://nar.oxfordjournals.org/content/38/14/4731.abstract`. 1, 4, 11, 12

M. F. Sagot and E. W. Myers. Identifying satellites and periodic repetitions in biological sequences. *Journal of Computational Biology*, 5(3):539–554, 1998. 53

Michael Sammeth and Jens Stoye. Comparing tandem repeats with duplications and excisions of variable degree. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3:395–407, 2006. ISSN 1545-5963. doi: http://doi.ieeecomputersociety.org/10.1109/TCBB.2006.46. 6, 7, 13, 16, 52, 72

T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147(1):195–197, 1981. 15

Alice A Storey, Andrew C Clarke, Thegn Ladefoged, Judith Robins, and Elizabeth Matisoo-Smith. Dna and pacific commensal models: Applications, construction, limitations, and future prospects. *The Journal of Island and Coastal Archaeology*, 8(1): 37–65, 2013. 88

J. Stoye and D. Gusfield. Simple and flexible detection of contiguous repeats using a suffix tree. *Theor. Comput. Sci.*, 270(1-2):843–856, 2002. 13, 53

Richard Truman, Amanda B Fontes, Antonio B de Miranda, Philip Suffys, and Thomas Gillis. Genotypic variation and stability of four variable-number tandem repeats and their suitability for discriminating strains of mycobacterium leprae. *Journal of clinical microbiology*, 42(6):2558–2565, 2004. 15

AJ Verkerk, M. Pieretti, J.S. Sutcliffe, Y.H. Fu, DP Kuhl, A. Pizzuti, O. Reiner, S. Richards, M.F. Victoria, FP Zhang, et al. A novel gene containing a trinucleotide repeat that is expanded and unstable on Huntington's disease chromosomes. The Huntington's Disease Collaborative Research Group. *Cell*, 72(6):971–983, March 1993. ISSN 0092-8674. URL `http://view.ncbi.nlm.nih.gov/pubmed/8458085`. 1, 11, 28

AJ Verkerk, M. Pieretti, J.S. Sutcliffe, Y.H. Fu, DP Kuhl, A. Pizzuti, O. Reiner, S. Richards, M.F. Victoria, FP Zhang, et al. Identification of a Gene (FMR-1) Containing a CGG Repeat Coincident with a Breakpoint Cluster Region Exhibiting Length Variation in Fragile-X Syndrome. *Cell*, 65(5):905–914, May 31 1991. 1, 11

P. Visca, S. D'Arezzo, F. Ramisse, Y. Gelfand, G. Benson, G. Vergnaud, N.K. Fry, and C. Pourcel. Investigation of the $Legionella\ pneumophila$ population structure by analysis of tandem repeat copy number and internal sequence variation. *Microbiology*, 2011. 54

Jonathan M Waters and Graham P Wallis. Across the southern alps by river capture? freshwater fish phylogeography in south island, new zealand. *Molecular Ecology*, 9 (10):1577–1582, 2000. 11

M. N. Weitzmann, K. J. Woodford, and K. Usdin. DNA Secondary Structures and the Evolution of Hypervariable Tandem Arrays. *J. Biol. Chem.*, 272(14):9517–9523, 1997. doi: 10.1074/jbc.272.14.9517. URL `http://www.jbc.org/cgi/content/abstract/272/14/9517`. 28

R. D. Wells. Molecular Basis of Genetic Instability of Triplet Repeats. *J. Biol. Chem.*, 271 (6):2875–2878, 1996. doi: 10.1074/jbc.271.6.2875. URL `http://www.jbc.org`. 28

Y. Wexler, Z. Yakhini, Y. Kashi, and D. Geiger. Finding approximate tandem repeats in genomic sequences. *Journal of Computational Biology*, 12(7):928–942, 2005. doi: 10.1089/cmb.2005.12.928. URL `http://www.liebertonline.com/doi/abs/10.1089/cmb.2005.12.928`. PMID: 16201913. 13, 14, 30, 31