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

PAW — the Protein Analysis Workshop for 2D Nuclear Magnetic Resonance Spectroscopy

VOLUME II

Spectral Processing and Assignment Guide

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

Wilford Lie

1999

The study described in this thesis has been supported by AgResearch, the New Zealand Pastoral Agriculture Research Institute, a division of the Crown Research Institute of New Zealand.

Errata for Volume II of Wilford Lie's PhD thesis, 1999

Note:
- Each correction is highlighted.
- A negative line number is counted from bottom up.

Page	Line	Correction
56	Fig 0.6	Figure 5.6
75	4	(Figure 6.4)
86	3	with a 5^{th} order polynomial
86	-1	(Figure 6.19)
87	2	(Figure 6.20)
115	1	the above display modes
117	-3	sent
169	-2	the entire region was closely checked
173	-5	Bloch-Siegert
214	-8	a horizontal oblong indicates
220	-6	Trp residue

22/12/99 13:13

Table of Contents

Chapter 1:

Preliminary

1.1 Introduction

PAW, the Protein Analysis Workshop for 2D Nuclear Magnetic Resonance (NMR) Spectroscopy, is an X Window-based software package to process and analyse high-resolution NMR spectra of proteins.

This guide provides operational instructions on using PAW. It is organised in an introductory way with examples, beginning with the basic interface method and moving gradually to the 2D NMR data processing and spectral assignment. However, showing an example often requires a number of preliminary operations related to other topics. Therefore, when reading a section, the reader is recommended to concentrate on the topic of the section, and ignore any unfamiliar operations.

Most of the nomenclature, notation and abbreviations employed in this booklet are commonly seen in many protein NMR articles. In addition, consistent visual-cues and documentation conventions are used to help identify the different types of information and achieve brevity in description.

1.2 Abbreviations and Notation

- nD n-dimensional, where n is a number, e.g., 1D, 2D, 3D
- COSY 2D correlated spectroscopy
- Dn Dimension-n, e.g., D1, D2, D3
- DQF-COSY Double-quantum filtered COSY
- f-domain frequency domain
- FFT fast Fourier Transform
- FID (or fid) free induction decay
- HN amide proton
- HA α-proton
- HB β-proton
- HG γ-proton
- HD δ-proton
- HE ε-proton
- HZ ζ-proton
- HH η-proton
- HFT Hilbert Fourier Transform
- IFT Inverse Fourier Transform

- LHS — the left-hand side of an equation
- MsBtn#1 — the left *mouse button*
- MsBtn#2 — the middle mouse button
- MsBtn#3 — the right mouse button
- NMR — Nuclear magnetic resonance
- NOE — nuclear Overhauser effect
- NOESY — 2D NOE spectroscopy
- PAW — the Protein Analysis Workshop, this package
- ppm — part(s) per million
- S/N ratio — signal-to-noise ratio
- SD — standard deviation
- SE — Schrödinger equation
- t-domain — time domain
- TOCSY — 2D total correlation spectroscopy
- t_1 — evolution period in a 2D NMR experiment
- t_2 — acquisition period in a 2D NMR experiment
- t1,t2,t3,t4 — The time-domains corresponding to D1, D2, D3 and D4
- T_1 — longitudinal relaxation time
- T_2 — transverse relaxation time
- γ — gyro-magnetic ratio
- $\Delta\omega$ — offset of ω_{rotate} from the Larmor frequency ω_0
- δ — chemical shift
- τ_m — mixing time
- ν_0 — Larmor frequency in Hz
- ω_0 — Larmor frequency in rad/sec
- ω_{rotate} — angular frequency of a rotating frame
- Ω — angular frequency offset of a Lamor frequency in a rotating frame

1.3 Textual Conventions

➢ Conventions for formatted text

- Text like *this* signals a special term.
- Text like this is a command, a keyboard entry or a header in a dialog.
- `Text like this is a macro- or program-statement.`
- Text in a macro or program headed with a hash like `#this` starts a note.
- Text enclosed by a pair of square brackets such as [Ernst *et al.* 1987] is a reference.
- Highlighted text enclosed by a pair of square brackets such as [Load] is a button in a menu, toolbox or a dialog.
- Two numbers separated by a comma enclosed by square brackets like [2000, 2100] defines a 1D region.
- Two words separated by a slash (/) within square brackets like [yes/no] is an exclusive-OR option for an entry.
- A highlighted character-string like Ctrl represents a key on the keyboard.
- A compound character-string like DataDir is a variable name used in PAW.
- A boldface character-string containing a dot such as **PAW.Init** is a filename.
- A boldface character-string ended with a slash (/) such as **~/PAW/** is a directory name, where ~/ is the user root directory.

➢ Conventions for mouse operations

- To point or to position means to move the mouse cursor to a specific screen position.
- To click means to position the mouse cursor, then press and release a mouse button without moving the mouse.
- To double-click means to click a mouse button twice in rapid succession without moving the mouse.
- To click-and-drag means to press a mouse button, hold it down while the mouse is moving, and then release the button at a desirable position.
- To choose means to click on a command button with MsBtn#1, e.g., 'choose [Convert]'.
- To pick means to drag the mouse cursor with MsBtn#2, then release the button after pointing to a desirable position, e.g., 'pick a peak'.
- To select an item means to click an item in a name list with MsBtn#1, e.g., 'select **data7noesy150.2Ddata**.
- To block or select a region means to define a rectangle region in a plot by dragging the mouse pointer with the MsBtn#1 after clicking on a position in a window then release the button at a desirable position. During the process, an expanding (or contracting) rubber rectangle is drawn while the mouse is moving. Note that the

edges of a rubber rectangle must not be less than five pixels; otherwise, the operation is cancelled.

- To load a file means to select a filename in the file list of a file-selection dialog, and then choose [Load] in the dialog, e.g., 'load data7noesy150.pdt'. If the filename is not in the current list, you may need to select a sub-directory in the directory list then choose [Filter] in the dialog to change the directory. (See Chapter 4.)

- To turn on (or off) a button means clicking on the button with MsBtn#1 so that it looks as if it has been (or not been) pressed down after the mouse button is released. For example, 'turn off [DspBufs] in the *1D-display Toolbox*'.

In addition, if no button number is specified for a mouse operation, the task is by default performed by MsBtn#1.

➢ Conventions for keyboard entries

- The key names used in this booklet match the standard keyboard labels, e.g., F1, Enter, Ctrl, Alt, #, Space.

- A comma between two key-names indicates that two keys are to be pressed sequentially, e.g., F3,F7.

- A plus sign between two key-names indicates that the first key must be pressed and held down while pressing the second key, e.g., Ctrl+F1.

- To *enter* a command means to type the command then press the Enter key while the mouse pointer (or cursor) is in a draw-window. (See Chapter 2.)

1.4 Installation

PAW is distributed on a compact disc that contains an executable file, various macros[1] and sample data. This section describes the installation procedure required.

➢ Installing PAW

1) Open an SGI UNIX shell as shown in Figure 1.1.

[1] A macro is a program containing PAW's macro statements.

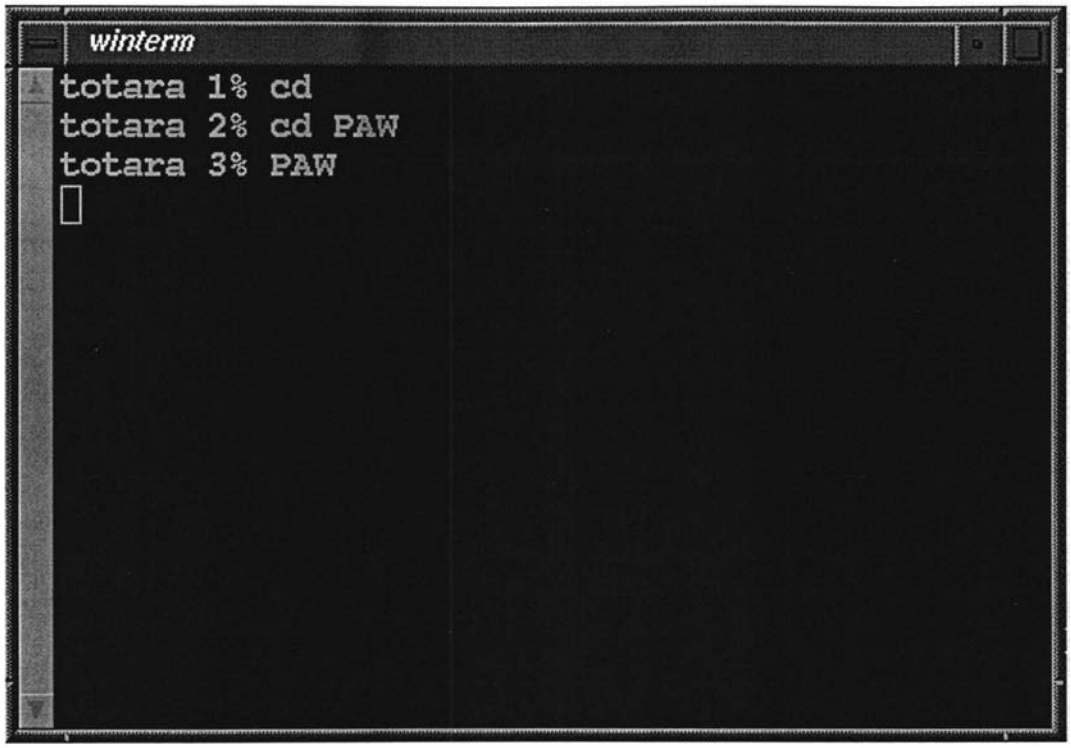


Figure 1.1: An SGI UNIX shell

2) Check to see if there is enough disc space by typing

```
df -k
```

The available disc space will be shown in bytes. If it is less than 180 mega-bytes, ask for more.

3) Make a directory for PAW under the user's root directory as follows:

```
cd
mkdir PAW
```

If permission is denied, ask for permission.

4) Copy all files in the CD into the PAW directory by entering

```
cd PAW
tar -xf /CDROM/
```

5) Read the final release notes by entering

```
more readme.first
```

7) Allow **PAW.setup** to become an executable file by entering

```
chmod +x PAW.setup
```

6) Finally, complete the installation by entering

```
PAW.setup
```

➢ Initialising PAW

Although several directories are made for initial installation, PAW, however, is not restricted to working with the directory names given in the installation instructions.

The way PAW understands details of the directory names is by reading a text file called **PAW.Init** stored in *PAW's system directory* called **~/PAW/**. Three of the statements given in the distributed file are as follows:

```
SystemDir = "/usr/people/liew/PAW/"
DataDir   = "/usr/people/liew/PAW/data/"
MacroDir  = "/usr/people/liew/PAW/macros/"
```

Here, the directory names are given as examples only. The content of the **SystemDir**, **DataDir** and **MacroDir** must be changed to match those in the computer in which PAW is installed. For example, if the root directory is **/disk2/usr/maria/**, then, change the statements as follows:

```
SystDir   = "/disk2/usr/maria/PAW/"
DataDir   = "/disk2/usr/maria/PAW/data/"
MacroDir  = "/disk2/usr/maria/PAW/macros/"
```

1.5 Starting PAW

Assuming that PAW has been properly installed, it can be run as follows:

- Open a Unix shell.
- Move the Unix shell to the bottom-left corner of the computer screen. It will be used as a runtime console.
- Type

```
cd
cd PAW
PAW
```

These operations will open PAW's main window[2]. If not, try to figure out the problem(s) in installation or contact the author for technical support.

1.6 Leaving PAW

➢ A normal method

- Either double-click on the window-control button of the Main Window or press Ctrl+c while the mouse cursor is pointing to the Unix shell from which PAW is run.

> Note: PAW does not check if the modified data have been saved when it is closed.

➢ An unusual method

As with all Unix applications, PAW can also be stopped unconditionally from other Unix Shell in the following way:

[2] See Chapter 2.

- Type

```
ps -ef | grep PAW
```

  This command shows the ID number assigned by the operating system to PAW as the third item in the second line.

- Type

```
kill -9 <ID for PAW>
```

Chapter 2:

User Interface

2.1 Introduction

This chapter describes different aspects of the PAW user interface. These include the menu bar, various windows, toolboxes, workbenches, dialogs, hotkeys, macros, commands, and help messages.

The standard window-control features are illustrated in Section 2.3. These features are common to all framed objects, including the overview window, draw-windows, dialog boxes and toolboxes.

2.2 The Main Window

The first window to appear when running PAW is the main window (Figure 2.1), where the *title bar* displays the window's name and the *main canvas* serves as a shield to cover PAW from other applications.

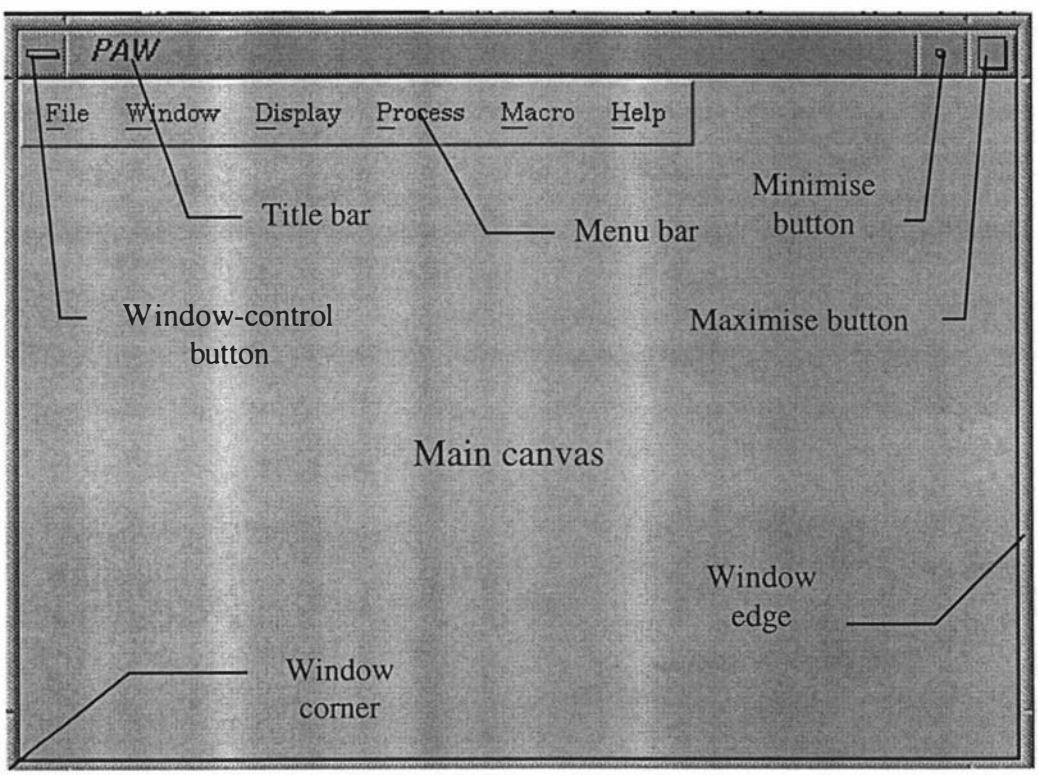


Figure 2.1: The main window.

The following window-control operations are standard to all window-like objects.

➢ **To reposition the main window**

- Point to the Title Bar
- Drag the mouse while holding down MsBtn#1, then release the button.

➢ **To move the entire application to the top**

- Either click on the Title Bar or any of the window edges and window corners.

➢ **To change the size of the window**

- Point to any of the window edges or the window corners.
- Drag the mouse while holding down MsBtn#1, then release the button

➢ **To shrink the whole application into a small icon**

- Click on the minimise button.

➢ **To expand the window into a full screen window**

- Click on the maximise button or type Alt-F10.

Alternatively, these tasks can be performed by the following operations:

- Click on the *window-control button* to open the *window-control menu* that contains a list of standard window-control options (Figure 2.2).
- Click on any button therein.

Pop	(# same as clicking the title bar)
Push	(# send the application to the background)
Move	
Resize	
Stow	(# same as minimise)
Open	(# only active after being stowed)
Full Size	(# same as maximise)
Close	(# close the application)
Quit	(# close the application)

Figure 2.2: The window-control menu

2.3 The Menu Interface

The *menu bar*, as shown in Figure 2.1, displays a list of *menu headers* for PAW's pull-down menus. To open a pull-down menu

- Click on a menu header with MsBtn#1.

The content in the menu bar and the pull-down menus described below are set up in a user-definable macro called **PAW.MenuBar**. (See Chapter 3.)

Six pull-down menus have been set up for the distribution package. These are the *File Menu*, *Window Menu*, *Display Menu*, *Process Menu*, *Macro Menu*, and *Help Menu*. Each of the menus consists of a number of options, as shown in the next figures. (Note

that the menus are defined in a macro, the contents in them are variable. The final release of this package may differ slightly.)

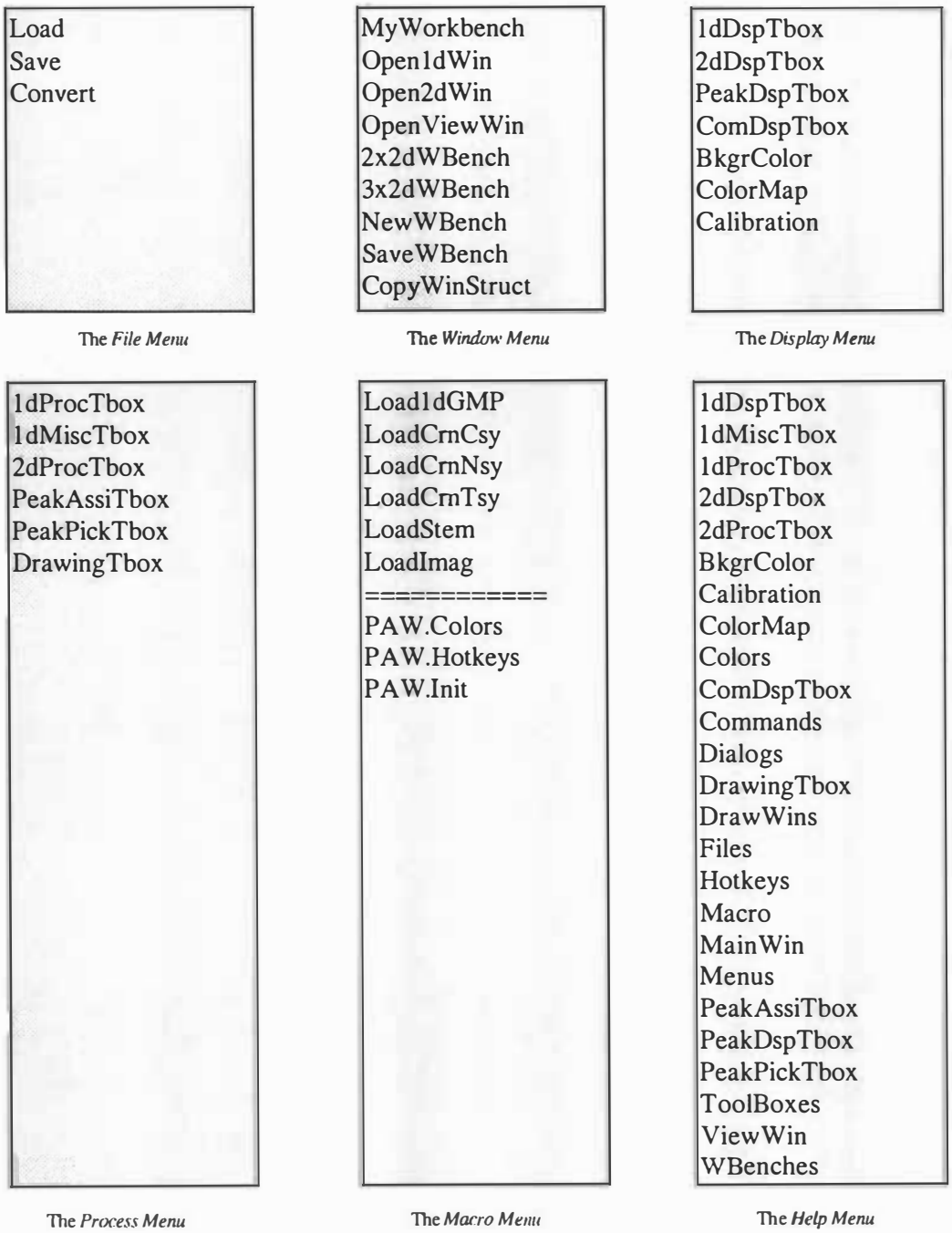


Figure 2.3: The six menus set up for the distribution package.

As with any X-Windows application, a *menu option* can be selected by keyboard or mouse operation.

➢ To execute a menu option

- Choose a menu header to open a pull-down menu.
- Choose a menu option in the menu.

2.4 The Draw-windows

A draw-window is a window for plotting data and drawing other objects such as lines, rectangles and text. The title bar of a draw-window displays the filename of the data displayed on the window, if any. The size, shape and location of a newly opened draw-window depend upon the pre-set geometric values for the window.

➢ To open a 1D draw-window

- Type o1w or choose [Open1dWin] from the *Windows pull-down menu* (Figure 2.4).

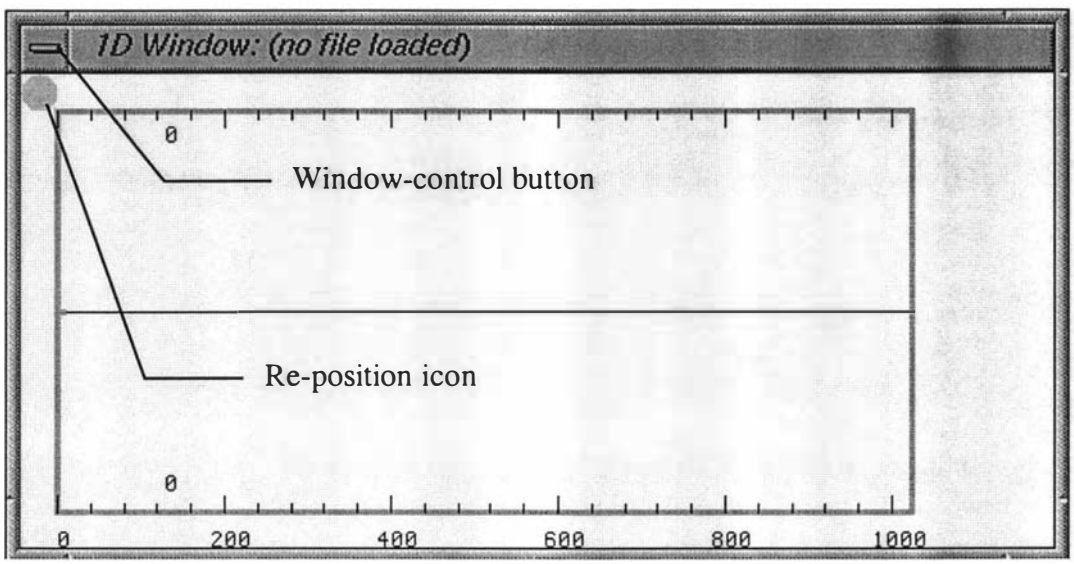


Figure 2.4: A 1D draw-window

➢ To close a 1D draw-window

- Type c1w or double-click on the window-control button.

➢ To display the geometric values of a 1D draw-window

- Type g1w or click in the window with MsBtn#3. The values will be shown in the Unix shell from which PAW is run. The values are useful for designing a workbench, as described in the next chapter.

➢ To open a 2D draw-window

- Type o2w or choose [Open2dWin] from the *Windows pull-down menu* (Figure 2.5).

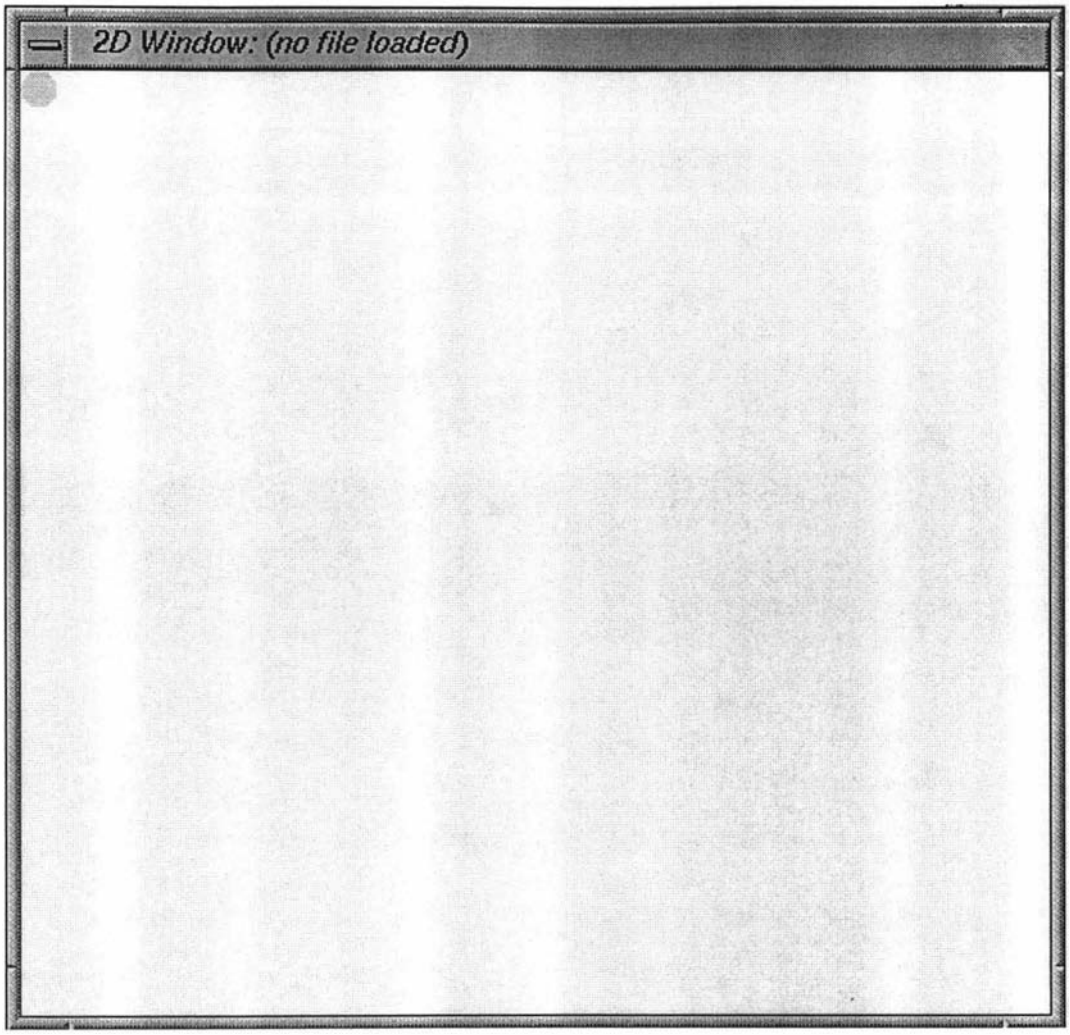


Figure 2.5: A 2D draw-window

➢ To close a 2D draw-window

- Type c2w or double-click on the window-control button.

➢ To display the geometric values of a 2D draw-window

- Type g2w or click in the window with MsBtn#3. The values will be shown in the Unix shell from which PAW is run. (This also clears the keyboard buffer.)

PAW can open up to nine draw-windows (excluding toolboxes) to display different spectra. Among them, only one can be currently active for each type of draw-windows, which are called respectively the *active 1D draw-window* and *active 2D draw-window*.

➢ To activate a draw-window

- Click in the window with MsBtn#1.

➢ To resume the initial size and position of a draw-window

- Click on the round grey circle on the top-left corner (i.e., the reposition icon).

➢ To change the background colour of a draw-window

- Choose [Display] in the menu bar to open the *Display Menu.*
- Choose [BkgrColor] in the menu to open the *SetBkgrColor Dialog* (Figure 2.6).

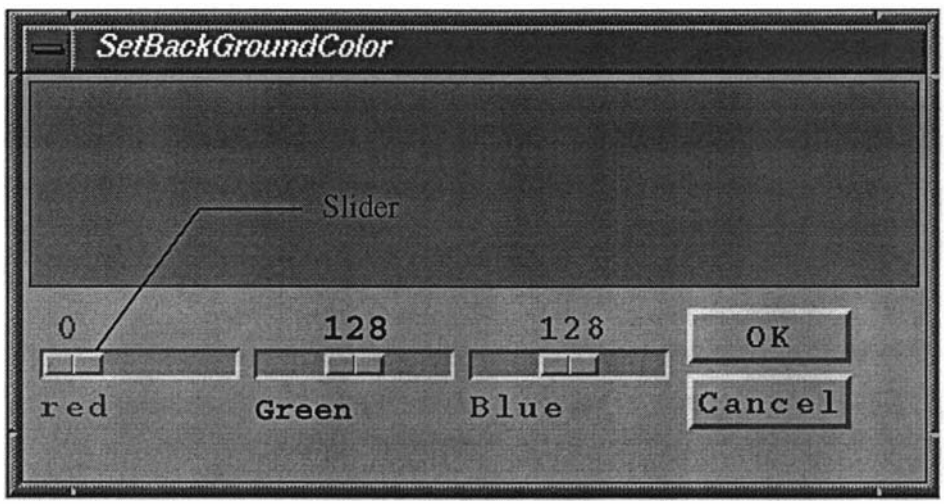


Figure 2.6: The *SetBkgrColor Dialog*.

- Adjust the colour values with MsBtn#1.
- Click in the draw-window for which the colour is to be set.
- Choose [OK] in the dialog to change the background colour of the draw-window.

Note that only the background colour of one draw-window is changed at a time.

➢ The draw-windows and buffers

In PAW, a data set is stored in a defined memory space called a *buffer*. Every 2D draw-window is linked to a file, which in turn is associated with a *2D buffer*. Every 1D draw-window, however, is commonly associated with nine *1D buffers*, and is not permanently attached to any file. Therefore, a 2D draw-window displays the content of the 2D buffer associated with a file that is linked to it. A 1D draw-window (selectively) displays the contents of nine 1D buffers that can be loaded from any file or files. The nine 1D buffers are named 1DBuf #0, 1DBuf #2, ... 1DBuf #8, respectively. A 2D buffer is dynamically created and associated to a file, and is not named. (See also section 2.12 on keyboard commands and buffers.)

2.5 The 2D View-window

The 2D view-window shows the global locations of 2D region(s) that are currently displayed on a 2D draw-window. The locations are indicated by a number of rectangles on a full-view 2D plot.

➢ To open a 2D view window

- Either type o2v or choose [Window] to open a menu then choose [Open2dViewWin] in the menu.

Figure 2.7 shows a view-window associated with the 2D multi-region plot.

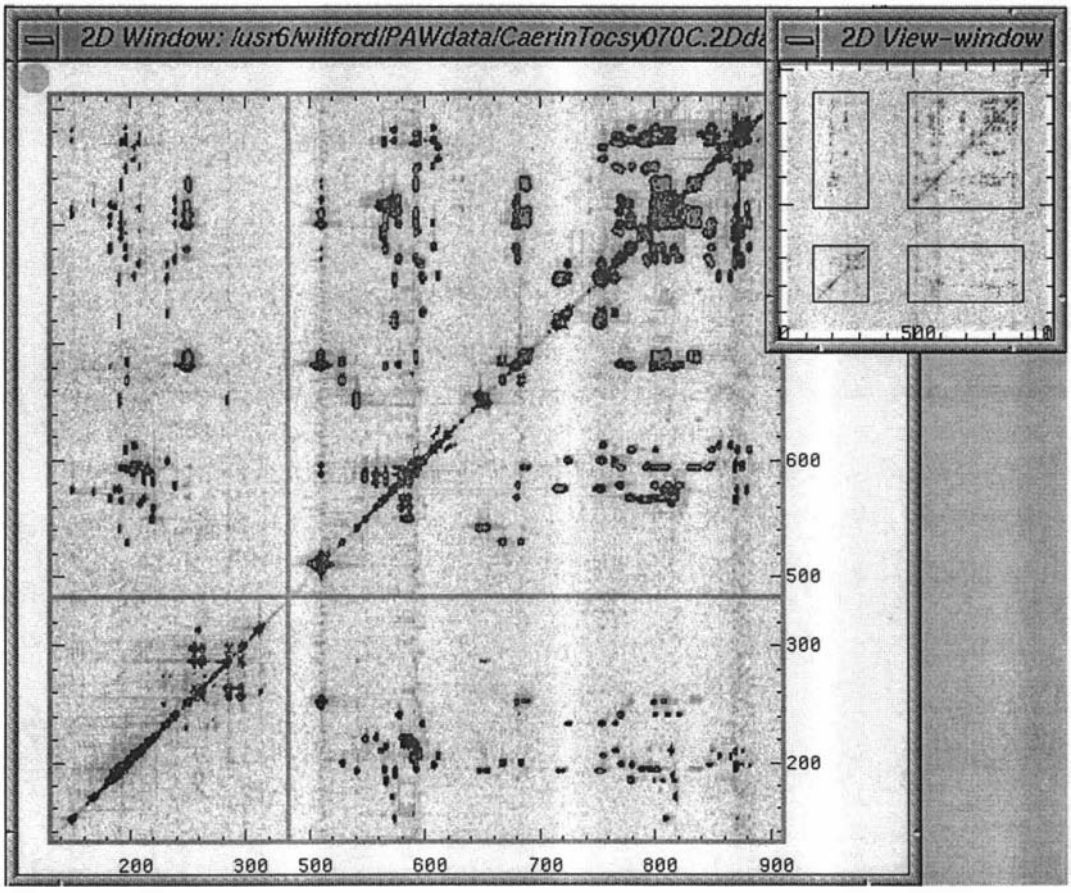


Figure 2.7: A 2D plot and a view window.

In fact, if a view-window is not opened, PAW always opens it when drawing a 2D plot. The initial location and size of the view-window are defined in **PAW.Init**. These can be re-defined in any other macros, as illustrated in the next chapter.

To speed up the display, PAW redraws only the rectangles on the view-window for every new 2D display, and does not upgrade its background spectrum.

➢ To redraw the plot in the view window

- Choose [DrawViewWin] in the *2D-display Toolbox* (see Chapter 6).

2.6 The Toolbox Interface

A toolbox is a window containing a number of command buttons. Different from the pull-down menu, a toolbox does not close by itself. In addition, they can be moved and resized.

Some toolboxes contain checkboxes and/or radio-boxes with a number of switching buttons. For example, the *Common-display Toolbox* (Figure 2.8) for setting both 1D and 2D common display parameters contains a radio-box, a checkbox and three command buttons.

The last two command buttons are common for all toolboxes.

Figure 2.8: The *Common-display Toolbox*

➢ To re-initialise parameters related to a toolbox

- Choosing [ResetParms].

➢ To reset the size and location of a toolbox

- Choose [ResetBox].

➢ To display the geometric values of a toolbox

- Click on the toolbox with MsBtn#3. The values will be shown in the Unix shell from which PAW is run. (This also clears the keyboard buffer.)

➢ To open a toolbox

- Either type an open-toolbox command or choose a name that contains the character string Tbox from the [Display] or [Process] menu.

For example, To open the *Common-display Toolbox*, either type ocd or choose [ComDspTbox] from the [Display] menu.

➢ To close a toolbox

- Either double-click on the window-control button of the toolbox, or type a close-toolbox command[1].

For example, to close the *Common-display Toolbox*, simply type xcd.

[1] See also Chapter 12 for a list of close-toolbox commands.

2.7 The Workbenches

A group of draw-windows and toolboxes that are opened for a task is called a *workbench* in PAW.

➢ To open a workbench

- Choose [Window] on the menu bar to open a menu.
- In the menu, choose a name that contains a character string WBench.

Figure 2.9 show the workbench opened by choosing [2 2dWBench] in the menu.

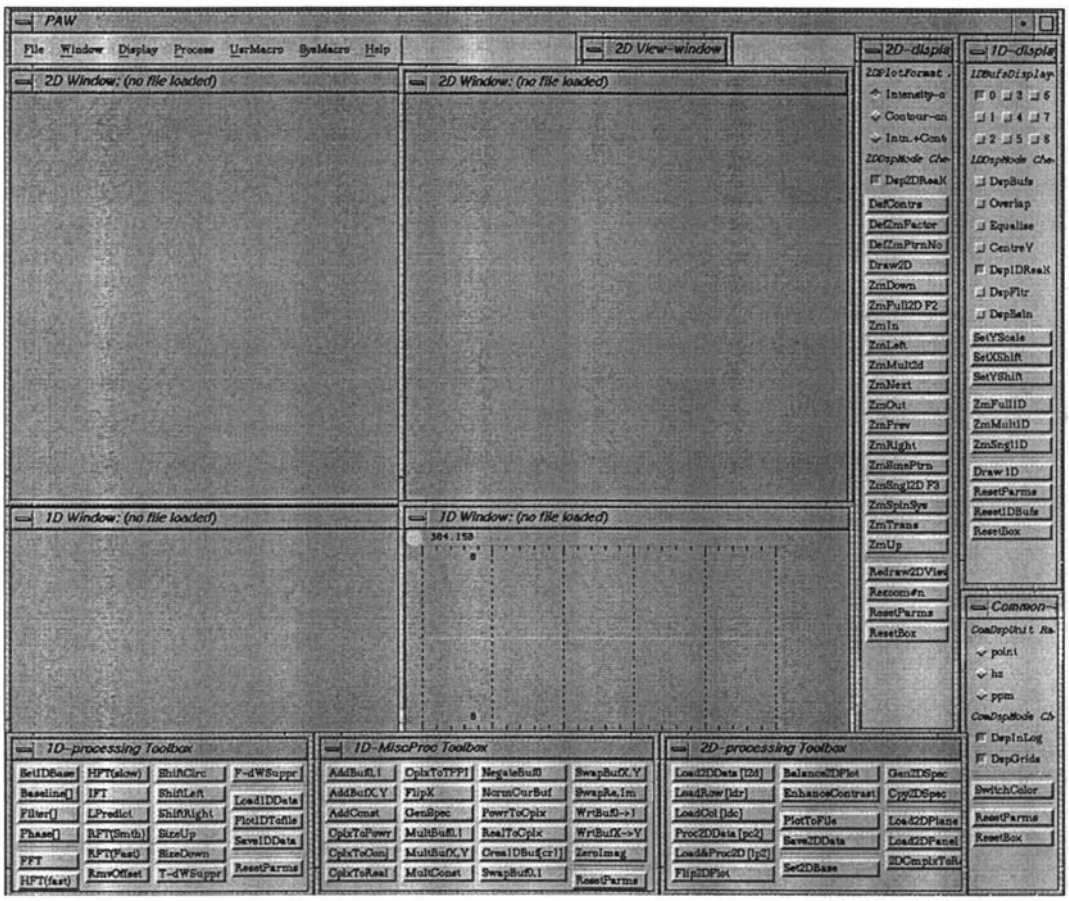


Figure 2.9: A complex of 2×2dWorkBench

Note that PAW does not close any existing window(s) before opening a workbench. It is recommended that all existing windows be closed before opening a workbench, unless they are to be kept. Otherwise, they may be covered without being noticed.

The content and layout of a workbench are user-definable, as described in Chapter 3.

2.8 The Dialog Interface

PAW frequently uses *dialog boxes* (or simply dialogs) as an important means of user interface. There are three categories of dialogs in PAW. These are the keyboard-entry, dual-entry, and file-selection dialogs.

➢ The keyboard entry dialog

An entry item in a keyboard-entry dialog is called a *field*, which can either be numeric or alphabetic. A box in which a value (or an *entry*) is to be entered for a field is called an *entry box*. Entries containing a mixture of both are considered alphabetic. For a numeric field, only numerical characters are allowed. For an alphabetic field, proper format must be strictly followed. Some entries expect a single character or a string of non-separated characters; others may expect a macro statement or a string of separated characters.

Note that a field that expects only a character is called a character field. In a dialog, its label contains a number of words separated by a slash (/) and enclosed in a pair of square brackets. An entry must be the first character of any enclosed words.

For example, the 2D-processing dialog on the right has a combination of different fields. The entries for the matrix dimensions must be numeric. An entry for any of the linear prediction, apodisation, phasing, and baseline correction must either be a macro statement or a single character n for 'no operation'. The entries for the last field must either be s for the States method or t for the TPPI method.

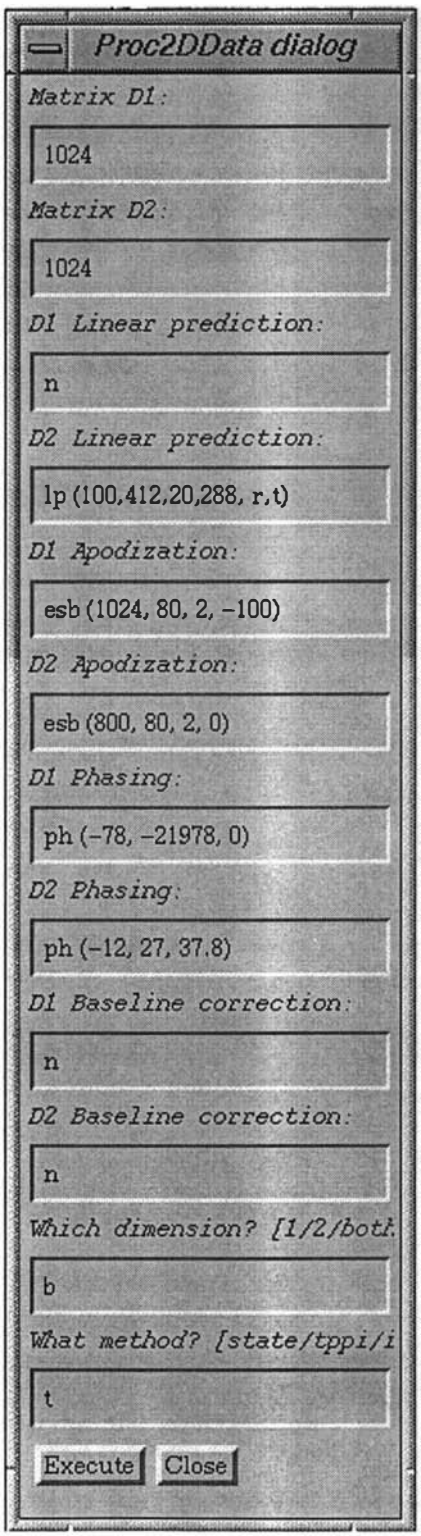


Figure 2.10: The *2D-processing Dialog*.

➢ The dual-entry dialog

A dual-entry dialog allows values to be set using either sliders or keyboard. To enter values with the keyboard, choose [Entry] in the dual-entry dialog.

For example, the *SetPh0 Dual-entry Dialog* (Figure 2.11) contains three sliders to set *Ph0*.

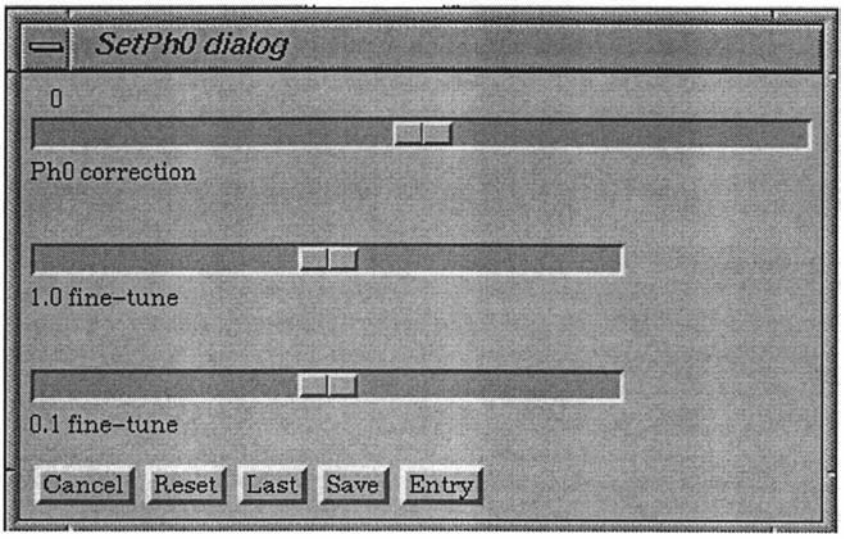


Figure 2.11: The SetPh dual- entry dialog

Here, the scales for the first two sliders is 1.0 and, for the third, 0.1. The total value is a summation of all three values.

If [Entry] in the dialog is chosen, it will be replaced by a keyboard-entry dialog (Figure 2.12).

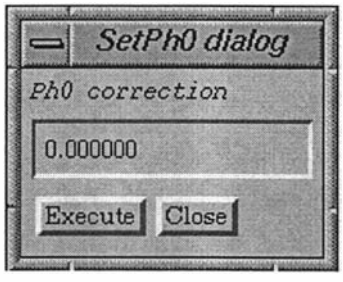


Figure 2.12: The SetPh keyboard-entry dialog

➢ The file-selection dialog

A file-selection dialog (Figure 2.13) contains two entry boxes with two separate lists of directory names and filenames for selection. Choosing a name in the file list automatically updates the content in the selection entry-box. Other details on the file-selection dialog will be discussed in the next chapter.

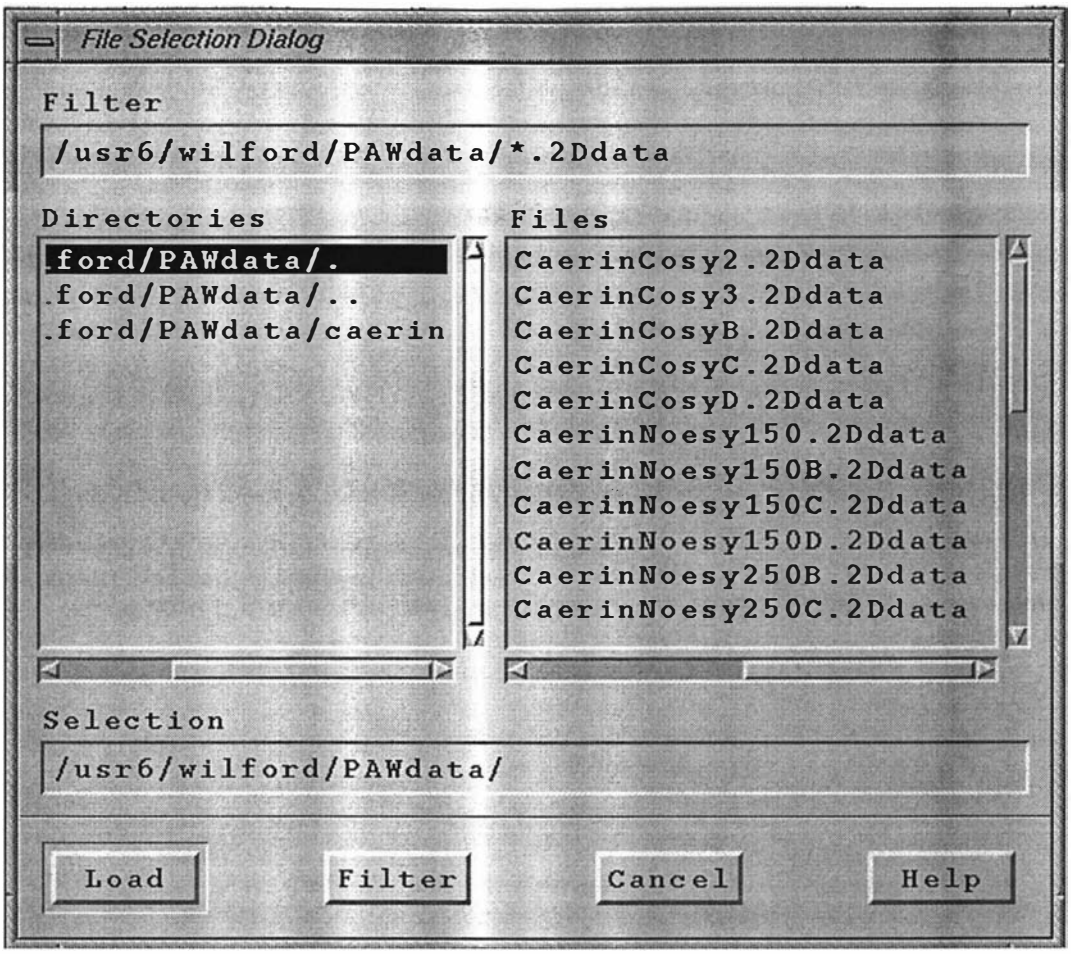


Figure 2.13: The file-selection dialog

➢ The exposure of a dialog box

Due to limited space on a screen, text in the dialogs are often not fully shown, because they are less important once they are known. They can be viewed by extending the window using the click-and-drag technique described previously.

A dialog's geometric parameters (including the size and location) can be re-defined, as described in the next chapter.

2.9 The Macro Interface

The macro interface is an efficient way of performing a sequence of operations. This section describes PAW's macro language, components and programming technique.

➢ The macro language

PAW's macro language has been designed to be as simple as possible. It mainly consists of three-character commands. A complete list of the available commands with explanation can be found in the last chapter of this booklet.

Note that the macro commands in PAW are case sensitive.

➢ The macro components

As with all other macro languages, the language is composed of six major components. These are the constants, variables, operators, commands, expressions, and statements. In brief, they can be described separately as follows:

1. There are only a few predefined constants. For example, the value of PI is defined as 3.141592654 in the system. In addition, any single character headed with an '@' is treated as a constant that equals the ASCII value of the character, e.g., @1 is the same as 49.
2. There are a number of pre-defined variable names, e.g., DataDir, ProjNo. In addition, there are 36 general purpose variables, including
   - six integer type loop-variables: %I, %J, %K, %L, %M, %N;
   - 10 integer type variables: I0, I1, ..., I9;
   - 10 float type variables: V0, V1, ..., V9; and
   - 10 string type variables: S0, S1, ..., S9.
3. The valid operators are:
   - the arithmetic operators: +, -, *, / and ^
   - the equal-operator: =
   - the negate-operator: -
   - the logical and-operato: &
   - the logical or-operator: |
   - the not-operator: !
   - *the relational operators*: >, <, >=, <=, == and !=
4. Every command in a macro consists only of either two or three lowercase characters.
5. An expression in a macro statement is any valid combination of operators, constants, variables, and round brackets that follows the general rules of algebra.
6. A macro statement is a line of ordered strings that follows strictly the macro statement conventions as described below.

➢ The macro statement conventions

- A statement must end with a linefeed without punctuation. Statement items must be separated by a Space or Tab, e.g.,

```
ScrnWidth = 1280
```

- A hash '#' in a line signals a comment, e.g.,

```
ScrnHeight = 938   # screen height in pixels
```

- An integer-type variable must be assigned a number, whereas a float type variable must be assigned a float value as shown below.

```
SpecFreq = 500.0   # spectrometer frequency 500 Mega Hz
```

- A logical-type variable must be assigned either T for true or F for false:

```
DebugOn = F   # logical variable can be False or True
```

- A numeric or an alphabetic character headed by an @ is treated as a number equivalent to its ASCII value. (The @ can be omitted for an alphabetic character.) For example, the following two statements print the same value on the Unix shell:

```
prn (1024*(z-96) +32*(@1-96))
prn (1024*(122-96) +32*(49-96))
```

- A string item must either be enclosed by a pair double-quotes, or headed by a '$' sign, e.g.,

```
DataDir   = "/usr6/wilford/data/PAWdata/"
DataDir   = $/usr6/wilford/data/PAWdata/
```

- The parameters for a command must be separated by comma and enclosed by round brackets, e.g.,

```
g1d (0, 93, 100, 630)        # Set 1dDsplBox
```

- An if-else-condition statement block must end with an end command, e.g.,

```
if (SiteNo=1)
     DataDir   = "/usr6/wilford/data/PAWdata/"
     MacroDir  = "/usr/people/liew/PAW/macros/"
else
     DataDir   = "/disk2/people/wilford/PAWdata/"
     MacroDir  = "/disk2/people/wilford/PAW/macros/"
end
```

- A for-loop statement block must end with a next command and the loop-variable headed by a '%', e.g.,

```
for (%I=1, 2, 1)
     prs ("--- Test 1 ---")
     for (%J=1, 12, 3)
          prs ("--- Test 2 ---")
     next
next
```

- A build-menu block (either bmm for 'build main menu' or bsm for 'build submenu') must end with a bme command, e.g.,

```
bmm  # build main menu
   bmb ("File",          F, none,  "smn: FileMenu")
   bmb ("Window",        W, none,  "smn: WindMenu")
   bmb ("Display",       D, none,  "smn: DsplMenu")
   bmb ("Process",       T, none,  "smn: ToolMenu")
   bmb ("Macro",    U, none,  "smn: UserMenu")
   bmb ("Help",          H, none,  "smn: HelpMenu")
bme
```

➢ Writing a macro

A macro is a collection of macro statements for defining variables and performing operations. Because macros are often run during varying stages of data processing and/or display, certain conditions must be met before running them. Run-time errors may occur if a macro is run with inadequate conditions. To avoid making mistakes in writing a macro, it is easier to start by copying an existing macro, and then modify it. Many macros are provided in the distribution package. They can be found in the system-macro directory and data directory.

Any text editor can be used for writing a macro. A macro written on a PC or a Macintosh must be saved in text format. These files may contain hard-return (ASCII #13) characters that are not recognised by an SGI word processor. PAW's package contains a program called **cv13to10** to convert all #13 characters into #10. To convert a file, type

```
cv13to10  <OrgFileName>  <NewFileName>
```

For example,

```
cv13to10 MyMacro NewMacro
```

2.10 The Hotkey Interface

➢ The hotkeys

PAW allows users to set up a number of hotkeys to perform different tasks. These are the F1 to F12 on the keyboard, and combinations of them with MsBtn#1, MsBtn#2 and MsBtn#3. Hence, a total of 48 keys can be set.

Since hotkeys can be set up to run a macro, they are capable of performing quite complicated tasks. The method used to define hot keys is described Chapter 3.

➢ To perform an operation with a combined hotkey

- Press a function key while holding a mouse button.

2.11 The Keyboard-command Interface

The keyboard-command interface allows the execution of commands without having to search for the buttons in the menus or toolboxes. A keyboard command is executed when three keys are typed sequentially. Almost all macro commands can be performed by keyboard entry. The available commands are listed in the last chapter of this booklet.

➢ To enter a two-character command

- Type the command followed with either the Space or Enter key.

➢ To enter a three-character command

- Type the three characters in sequence

➢ To repeat the last command

- Press the '.' key.

➢ To clear the keyboard buffer

- Click in a draw-window with the MsBtn#3. (This also lists the window's geometric parameters.)

➢ Keyboard commands and target draw-windows

There are a few general points to remember regarding how commands affect draw-windows:

- A keyboard command that changes the data displayed on a draw-window must be entered after the window is clicked and while the cursor is positioned in it.
- If a series of commands are entered for the same draw-window, only one click at the beginning is required.
- A common zooming command (such as zf, zs or zm for zooming into a full-single- or multi-region view) can change the display on either the 1D or 2D active draw-window, depending on which one is clicked last.

➢ Keyboard commands that required more mouse operation(s)

Some commands require that other mouse operation(s) be performed after entering them. In this case, the operation(s) required will be shown on top of the draw-window or the Unix shell.

For example, the steps to zoom the display on a draw-window (say, Window #2) with the same pattern as that shown on another draw-window (say, Window #1) are as follows:

- Click in Window #1.
- Type zsp (for zooming with the same pattern).
- Click in Window #2.

➢ Keyboard commands and changes to buffers

There are two simple rules regarding which buffer a command acts on:

- A 1D-processing command always changes data in 1DBuf#0 and displays the result in the active 1D draw-window.
- A 2D-processing command always changes the buffer associated with the active 2D draw-window.

In addition, 1DBuf #6 to #8 are also changed during some complicated operations such as curve fitting and baseline correction. Therefore, to backup a result in 1DBuf #0, it must be saved into either 1DBuf #2, 3, 4 or 5 before a new operation is performed, as described in Chapter 5.

➢ Keyboard commands and sizes of buffers

A buffer is dynamically created before an NMR data file[2] is loaded or created. The sizes of all 1D buffers are identical. The maximum 1D buffer size is set to 32768 bytes each by a macro-variable called Max1DDim in **PAW.Init**. The need to reduce

[2] In PAW. there is no difference in the file formats between time-domain and frequency-domain NMR data files. The term 'NMR data files' is used for data in both domains. The time- and frequency domain data (sets) are also called respectively the raw data (sets) and spectra.

Max1DDim is rare unless memory space is very limited. It must not be less than twice the size of any 1D data set loaded, because some 1D processes require doubled size buffers.

2.12 The Help Messages and Instructions

Depending on the content of the message, text can appear in one of the following three panels.

1. Runtime messages are displayed in the Unix shell from which PAW is run, which provides a history of operations.
2. The cursor and any immediate action required are displayed in the active draw-window.
3. An important message is displayed in a pop-up shell, as shown in the next diagram.

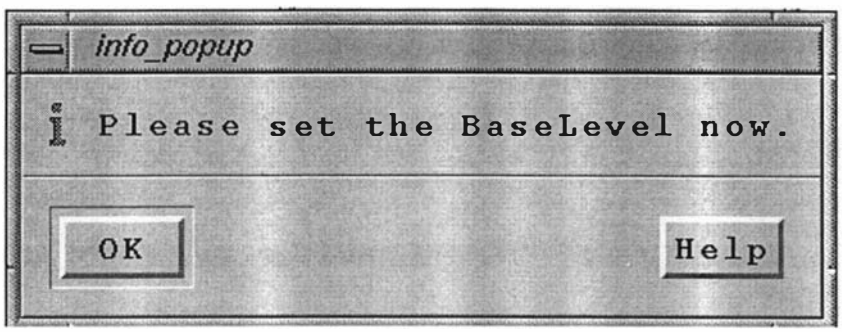


Figure 2.14: A pop-up information shell

If [Help] is chosen, another pop-up shell will appear, such as the one shown below.

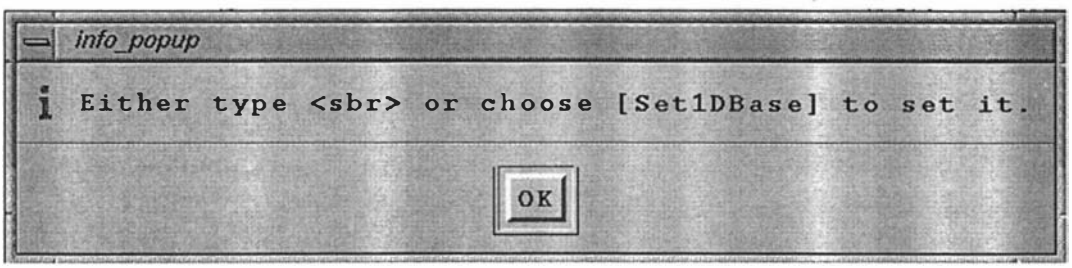


Figure 2.15: A help shell

Some help messages are kept in the help macros that can be viewed by choosing a menu button in the *Help Menu*. More details on the help macros can be found in Chapter 4.

Chapter 3:

Building a Working Environment

3.1 Introduction

This chapter describes the methods used to build a working environment. These include changing system parameters, modifying the menu-bar, defining hotkeys, and creating customised workbenches.

3.2 Initialising System Parameters

System parameters are initialised in the macro file **PAW.Init**, which is run at start-up. The macro distributed with this booklet is as follows:

```
# Program begins
# --------------------
# File name: PAW.Init
# --------------------

# ----------------------
# Print welcome message:
# ----------------------
prs ("*****************************************************")
prs ("*                                                   *")
prs ("*                  WELCOME TO PAW                   *")
prs ("* --- An XWindows based Protein Analysis Workshop  *")
prs ("*             for Processing and Assigning          *")
prs ("*       High Resolution NMR Spectra of Proteins     *")
prs ("*                                                   *")
prs ("*****************************************************")

# --------------------
# Assign variables:
# --------------------
SiteNo         = 2
ProjNo         = 1
ScrnWidth      = 1280
ScrnHeight     = 938
Max1DDim       = 32768  # Maximum is 32768.
DebugOn        = F

run PAW.Colors      # run another macro for colour assignment

if (SiteNo=1)
   # At Massey University:
   SystDir  = $/disk2/people/wilford/PAW/
   MacroDir = $/disk2/people/wilford/PAW/macros/
   DataDir  = $/disk2/people/wilford/PAWdata/
   prs ("At Massey University")
else
   # At Wallaceville:
   SystDir  = $/usr/people/liew/PAW/
   MacroDir = $/usr/people/liew/PAW/macros/
   DataDir  = $/usr6/wilford/PAWdata/
   prs ("At Wallaceville")
end

if (ProjNo=1)
   # Caerin 4.1
   Sequence = $GLWQKIKSAAGDLASGIVEGIKSX
end
if (ProjNo=2)
```

```
    # Caerin 4.0
    Sequence = $GLWQKIKSAAGDLASGIVEGIKSX
end

# --------------------
# Print the values of two variables:
# --------------------
prn (ProjNo)
prn (SiteNo)
prs (DataDir)

# --------------------
# Set the geometry parameters (ie, XYWH) of 1dDtProcShell, etc.:
# --------------------
g1d (0, 93, 100, 630)   # Set_1dDsplBox
g1m (0, 93, 100, 600)   # Set_1dMnplBox
g1p (0, 93, 374, 123)   # Set_1dProcBox
g1w (0, 93, 640, 280)   # Set_1dDrawWin

g2d (8, 32, 116, 880)   # Set_2dDsplBox
g2p (0, 93, 107, 635)   # Set_2dProcBox
g2v (643, 58, 160, 160) # Set_2dViewWin
g2w (0, 93, 640, 640)   # Set_2dDrawWin

gbc (0, 93, 100, 200)   # SetBsLineBox
gft (0, 93, 100, 100)   # SetFltBox
gph (0, 93, 100, 100)   # SetPhaseBox

gcd (8, 30, 118, 228)   # SetComDspBox
gdr (0, 93, 100, 800)   # SetDrawBox
gpa (0, 93, 100, 800)   # SetPeakAssiBox
gpd (0, 93, 100, 800)   # SetPeakDspBox
gfs (0, 93, 600, 500)   # SetFileSelBox
# Program ends
```

The meaning of each command in the file is briefly commented. Note that every variable must be assigned a valid value; otherwise, the program may be suspended if severe errors occur.

The ProjNo is useful when PAW is used to assign more than one protein.

➢ To add a new project

- Add an if-block for the project, which must contain at least a statement that assigns the variable named Sequence. The block may also contain statements for setting the data and macro directories.

➢ To switch from one project to another

- Change the ProjNo before starting PAW.

The SiteNo is only useful for running PAW on different computers to perform the same task. In this case, changing the SiteNo specifies different directories for PAW.

➢ To add a new site

- Add an if-block for the site, which must contain statements for setting the data and macro directories.

➢ To switch from one site to another

- Change the SiteNo before starting PAW.

➢ To set colours

Colours are assigned in a separate macro run by **PAW.Init** called **PAW.Colors**. The macro distributed with this version of PAW is as follows:

```
# Program begins

BkgrColor      = $Grey
TextColor      = $black
DrawColor      = $navy
CPeakColor     = $blue
RPeakColor     = $black
DPeakColor     = $yellow
PstvCtourColor= $red
NgtvCtourColor= $ForestGreen

# Curve colours:
Color01        = $navy
Color02        = $Red
Color03        = $Orange
Color04        = $Magenta
Color05        = $black
Color06        = $Blue
Color07        = $Brown
Color08        = $Violet
Color09        = $Grey
Color10        = $Cyan         # also used for 1D x-axes
Color11        = $Purple       # also used for filters and baselines
Color12        = $Yellow
Color13        = $Pink
Color14        = $Green
Color15        = $LimeGreen
Color16        = $Wheat
Color17        = $Turquoise
Color18        = $Wpresse

rsc                            # update colours

# Program ends
```

Note that the colours set in this way does not affect the colour map. In addition, only *valid colour names* can be assigned to the colour variables.

➢ To view the defined names for different colours

- On a Unix, shell, type

```
more /usr/lib/X11/rgb.txt
```

- Repeatedly press the SpaceBar to view the list, as shown in Figure 3.1

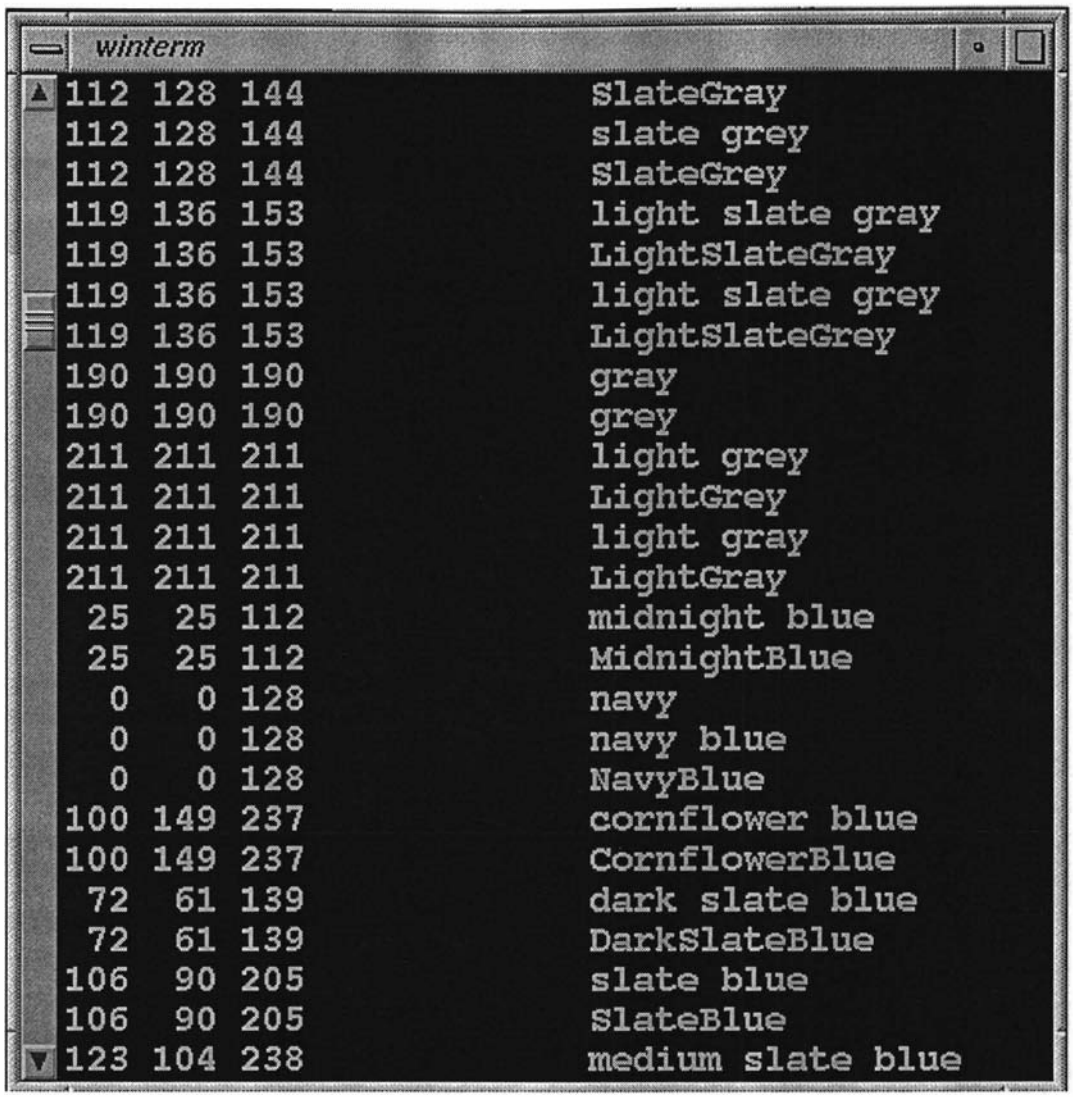


Figure 3.1: The system colour definitions.

To search for all colours of the same kind say, blue, type

- Type

```
more /usr/lib/X11/rgb.txt | grep lue
```

Here, the string 'lue' enable the display of all defined colour-names containing 'Blue' or 'blue'.

➢ To change a colour

- Replace the one in the set BkgrColor statement with a defined colour name.

Note that all of the parameters and variables set in **PAW.Init** and **PAW.Colors** can be reset in any other macros. PAW only keeps the settings defined in the most recent macro it runs. A simple way to have a few sets of parameters is to have a few macros built into the pull-down menus, as shown in the next section.

3.3 Modifying the Menu-bar

This section describes the method used to modify the menu bar. An example to add a menu to the menu bar is given. It involves the addition of a statement and a build-menu block to a system macro.

➢ The macro that defines the menu bar

The menu bar is built by the macro **PAW.MenuBar**, which is run immediately after PAW starts. The macro distributed with this booklet is as follows:

```
# Program begins
# ----------------------
# File name: PAW.Menubar
# ----------------------
#
# The parameters for each menu and window must be defined
# with the formats shown in the examples below.
# Each item in a line MUST be separated by either a space,
# a comma or a colon.
# The macro commands involved in building the menu-bar are:
#    bmm ---   BldMainMenu
#    bmb ---   BldMenuBtn
#    bme ---   BldMenuEnd
#    bsm ---   BldSubmenu
# The callback indicators in the bmb statement are:
#    com ---   macro command indicator
#    smn ---   submenu indicator
#    mcr ---   macro indicator


#************************************************
#    Block one: Building MenuBar                *
#************************************************
bmm # Menu bar
  bmb ("File",      F, Alt+F, "smn: FileMenu")
  bmb ("Window",    W, Alt+W, "smn: WindMenu")
  bmb ("Display",   D, Alt+D, "smn: DsplMenu")
  bmb ("Process",   P, Alt+P, "smn: ProcMenu")
  bmb ("Macro",     U, Alt+U, "smn: MacroMenu")
  bmb ("Help",      H, Alt+H, "smn: HelpMenu")
bme

#************************************************
#    Block two: Building Menus                  *
# Each bmb in the Menu MUST be built in order.  *
#************************************************
bsm # FileMenu
  bmb ("Load",      L, none, "smn: LoadMenu")
  bmb ("Save",      S, none, "smn: SaveMenu")
  bmb ("Convert",   C, Alt+C, "com: ocv")
  bmb ("Show notes",    o, none, "mcr: notes.mcr")
bme

bsm # WindMenu
  bmb ("Open1dWin",      1, none, "com: o1w")
  bmb ("Open2dWin",      2, none, "com: o2w")
  bmb ("Open2dViewWin", T, none, "com: o2v")
  bmb ("2 2d WBench",    M, none, "mcr: WBench_2x2d.mcr")
  bmb ("3 2d WBench",    M, none, "mcr: WBench_3x2d.mcr")
  bmb ("New  WBench",    M, none, "mcr: WBench_New.mcr")
  bmb ("SaveWBench",     P, none, "com: swb")
  bmb ("CopyWinStruct", T, none, "com: cws")
```

```
bme

bsm # DsplMenu
  bmb ("1dDspTbox",     D, none, "com: o1d")
  bmb ("2dDspTbox",     D, none, "com: o2d")
  bmb ("PeakDspTbox",   D, none, "com: opd")
  bmb ("GenDspTbox",    D, none, "com: ogd")
  bmb ("BkgrColor",     B, none, "com: sbg")
  bmb ("ColorMap", M, none, "com: scm")
  bmb ("Calibration",   C, none, "com: ocb")
bme

bsm # ProcMenu
  bmb ("1d ProcTbox",   1, none, "com: o1p")
  bmb ("1d MiscTbox",   M, none, "com: o1m")
  bmb ("2d ProcTbox",   2, none, "com: o2p")
  bmb ("PeakAssiTbox",  P, none, "com: opa")
  bmb ("PeakPickTbox",  P, none, "com: opp")
  bmb ("DrawingTbox",   D, none, "com: odr")
bme

bsm # MacroMenu
  bmb ("RefineCosy5",   I, none, "mcr: RefineCrinCosy5.mcr")
  bmb ("RefineCosy4",   I, none, "mcr: RefineCrinCosy4.mcr")
  bmb ("RefineCosy3",   I, none, "mcr: RefineCrinCosy3.mcr")
  bmb ("RefineCosy2",   I, none, "mcr: RefineCrinCosy2.mcr")
  bmb ("RefineCosy",    I, none, "mcr: RefineCrinCosy.mcr")
  bmb ("RefineCosyD1",  I, none, "mcr: RefineCrinCosyD1.mcr")
  bmb ("RefineCosyD2",  I, none, "mcr: RefineCrinCosyD2.mcr")
  bmb ("ProcCrinCosy",  7, none, "mcr: ProcCrinCosy.mcr")
  bmb ("ProcCrinTsy070",7, none, "mcr: ProcCrinTsy070.mcr")
  bmb ("ProcCrinNsy080",8, none, "mcr: ProcCrinNsy080.mcr")
  bmb ("ProcCrinNsy150",5, none, "mcr: ProcCrinNsy150.mcr")
  bmb ("ProcCrinNsy250",2, none, "mcr: ProcCrinNsy250.mcr")
  bmb ("ProcLSZcosyb",  I, none, "mcr: ProcLSZcosyb.mcr")
  bmb ("Load 1d GMP",   G, none, "mcr: Load1dGMP.mcr")
  bmb ("Load Stem",     S, none, "mcr: LoadStem.mcr")
  bmb ("Load Imag",     I, none, "mcr: LoadImag.mcr")
  bmb ("==========", I, none, "mcr: PAW.Separator")
  bmb ("PAW.Colors",    P, none, "mcr: PAW.Colors")
  bmb ("PAW.Hotkeys",   P, none, "mcr: PAW.Hotkeys")
  bmb ("PAW.Init", P, none, "mcr: PAW.Init")
bme

bsm # HelpMenu
  bmb ("Contents", C, none, "mcr: HelpCont.mcr")
  bmb ("Index",         I, none, "mcr: HelpIndx.mcr")
bme

#*************************************************************
#    Macro block three: Building sub-Menus                   *
# Each sub-Menu must also be built in order.                 *
#*************************************************************
bsm # LoadMenu
  bmb ("Load1dData",    1, Alt+1, "com: ld1")
  bmb ("Load2dData",    2, Alt+2, "com: ld2")
  bmb ("LoadPkLists",   P, Alt+P, "com: lvp")
bme

bsm # SaveMenu
  bmb ("Save1dData",    1, none, "com: sv1")
  bmb ("Save2dData",    2, none, "com: sv2")
  bmb ("SavePkLists",   P, none, "com: svp")
bme
# Program ends
```

Only four commands are involved in this macro; each of them is briefly commented. The macro starts with the bmm-block to build the menu-bar main items, followed by a series of bsm-blocks to build submenus and sub-submenus in the order of appearance.

Each of the build-menu blocks contains a number of bmb statements to build menu items as follows:

```
bmb (<label>, <underlined character>, <hotkey>,
     <button command>)
```

The second and third parameters are reserved for future development. The value for the fourth parameter must be a string, which can either be

- a *submenu indicator* smn that is followed by a colon and a submenu name,
- a *command indicator* com that is followed by a colon and a macro command, or
- a *macro indicator* mcr that is followed by a colon and a macro filename.

Note that when modifying the macro, the order of the build-menu blocks and the formats of the statements must be correct to avoid run time errors.

➢ To add a menu button

- Add a bmb statement to a submenu building block.

➢ To remove a menu button

- Remove a bmb statement from a submenu building block.

➢ To add a submenu button

- Insert a bmb statement to the bmm-block.
- Insert a submenu block with a number of bmb statements at an appropriate location.

➢ To remove a submenu button

- Remove the related bmb statement from the bmm block.
- Remove the bmm-block that builds the submenu.

Example

The following operations add a submenu called *MyMacro Menu* after the *Macro Menu* and build two buttons that run respectively **MyMacro1.mcr** and **MyMacro2.mcr**:

1. In the bmm-block, insert a bmb statement after that for the *Macro Menu* as follows:

```
bmb ("MyMacros ",     M, none, "smn: MyMacros ")
```

2. Add a build-submenu block after the *MacroMenu* block with two bmb statements as follows:

```
bsm # MyMacros
   bmb ("MyMacro1",   1, none, "run: MyMacro1.mcr")
   bmb ("MyMacro2",   2, none, "run: MyMacro2.mcr")
```

```
bme
```

3.4 Defining Hotkeys

This section describes the method used to define a hotkey. It involves the addition of a statement that assigns a string containing a macro statement to a predefined key name.

➢ The key-name conventions

- All *simple key-names in PAW's macro* starts with the word 'Key', followed by a function-key symbol as shown on the keyboard, e.g., KeyF1, KeyF2;
- All *complex key-names in PAW's macro* starts with the word 'Key', followed by a simplified word B1, B2 or B3, then a function key symbol as shown on the keyboard, e.g., KeyB1F1.

Only a few hotkeys are set up in the distribution package. They are defined in a system macro called **PAW.Hotkeys**, in which the first two lines are:

```
# --------------------
# Define hotkeys:
# --------------------
KeyF2 = $zf
KeyF3 = $dr
```

➢ To define a hotkey

- Open the **PAW.Hotkeys** using a text editor.
- Add an assign statement to assign a key name with a string starting with a dollar sign ($).

➢ To define a hotkey that runs a macro

The statement for setting up a hotkey to run a macro has a special format as follows: (Note that the macro names are not enclosed in double quotes.)

```
<key name> = $run ("", <macro name>)
```

For example, to associate F5 with the macro called **PAW.zm1D**:

```
KeyF5 = $run ("", /usr/people/liew/PAW/PAW.zm1D)
```

The content of **PAW.zm1D** is as follows:

```
sw1         # Set 1D active window to be the current window
zf          # Zoom full
dr          # Re-draw 1D plot
zm          # Set up for multi-region zooming operations
```

The first statement is equivalent to clicking in the 1D active window.

➢ To define a combined key

To define a combined key for changing the data directory, the statement can be as follows:

```
KeyB3F5 = $DataDir = /usr6/wilford/data/PAWdata/caerin
```

Note that combined hotkeys are not suitable for performing tasks that require mouse operations.

3.5 Designing a Workbench

A group of draw-windows and toolboxes that are opened for a task or a project is called a *workbench* in PAW. The content of a workbench can be defined by a macro. This section describes the method used to design a workbench.

➢ The workbench macros

A macro that generates a workbench contains statements for setting the geometric parameters of the toolboxes and draw-windows, as well as statements for opening them. In addition, it may also contain statements for closing toolboxes and setting background colours for the draw-windows. For example, the macro below closes all toolboxes, and then generates a workbench containing a number of toolboxes with one 1D and three 2D draw-windows in a light-grey colour defined by the sbg statements.

```
# WBench_2x2d.mcr
xab                        # Close all toolboxes
g2v (643 58 160 160)       # Set 2D view-window
g2d (828 58 100 648)       # Set 2D-display toolbox
gpa (943 58 100 284)       # Set peak-assignment toolbox
gpp (943 332 100 375)      # Set peak-picking toolbox
gdr (1057 58 100 648)      # Set drawing toolbox
g2p (828 604 285 98)       # Set 2D-processing toolbox
g1d (1172 58 100 648)      # Set 1D-display toolbox
g1p (828 729 317 143)      # Set 1D-processing Toolbox
gft (860 754 350 75)       # Set filtering toolbox
gph (888 779 349 80)       # Set phasing toolbox
gbc (918 804 350 73)       # Set baseline-correction toolbox
g1m (828 900 430 116)      # Set 1D miscellaneous-functions toolbox
gcd (1172 622 100 268)     # Set common-display toolbox
gpd (1172  99 100 772)     # Set peak-display toolbox

sbg (228,228,228)          # Set background color
o2v                        # open 2D view-window

g1w (8 725 520 236)        # Set 1D draw-window
o1w                        # Open 1D draw-window
sbg (228,228,228)          # Set background color

g2w (8  93 520 470)        # Set 2D draw-window
o2w                        # open 2D draw-window
sbg (228,228,228)          # Set background color

g2w (28  133 520 470)      # Set 2D draw-window
o2w                        # open 2D draw-window
sbg (228,228,228)          # Set background color

g2w (48  173 520 470)      # Set 2D draw-window
o2w                        # open 2D draw-window
sbg (228,228,228)          # Set background color

o2d       # Open 2D-display toolbox
```

```
opa         # Open peak-assignment toolbox
opp         # Open peak-picking toolbox
odr         # Open drawing toolbox
o2p         # Open 2D-processing toolbox

o1d         # Open 1D-display toolbox
o1p         # Open 1D-processing toolbox
opd         # Open peak-display toolbox
ocd         # Set common-display toolbox
o1m         # Set 1D miscellaneous-functions toolbox
```

Note that an object opened will cover (fully or partially) any opened objects if there are overlapped regions defined in the values of the geometric parameters. Also, if an object is opened immediately after its geometric parameters are defined, the SGI system manager may not respond correctly — a bug beyond control. That is why nearly all of the objects are opened at the end.

➢ To design a workbench

- Choose [Window] in the menu-bar.
- Choose any of the pre-set workbenches. (Optional)
- Open the draw-windows and toolboxes for the new workbench, and close those unwanted windows and toolboxes, if any.
- Rearrange the sizes and the locations of the toolboxes and draw-windows with the mouse operations described in Section 2.3.
- Either type swb or choose [SaveWBench] in the Window menu to save the workbench layout onto the macro called **WBench_New.mcr** in the macro directly.
- Rename the macro.
- Rearrange the opening order of the windows and re-align the objects by making similar values in the brackets identical.

For example, an automatic created workbench macro may look like this:

```
# WBench_new.mcr
g2v (643 58 160 160)
o2v
g1w (8 572 490 232)    # Set 1D draw-window (first statement)
o1w
g2w (8 93 486 445)
o2w
g2w (445 94 492 445)
o2w
g1w (445 572 494 236) # Set 1D draw-window (second statement)
o1w

x1d          # Close the 1D-display Toolbox.
g1d (1068 59 100 571)
o1d

x1m
g1m (350 825 372 143)
o1m
```

```
xlp
glp (9 825 326 143)
olp

x2d
g2d (953 58 100 730)
o2d

x2p
g2p (737 825 320 142)
o2p

xbc
gbc (154 882 377 82)
obc

xft
gft (91 827 366 80)
oft

xcd
gcd (1069 669 100 297)
ocd

xph
gph (123 854 371 88)
oph
```

Here, a statement starting with an 'x' closes a toolbox. The reader may wonder why it closes every toolbox individually. This again is due to problems in the SGI system management. (Note that the macro created in this way may not open the draw-windows and the toolboxes in a desirable order. Also, the alignment and the widths of objects in the workbench may not be perfect. For example, the four underscored values for the widths of the draw-windows can be made identical using a text editor.)

➢ To modify an existing workbench

Often, a small modification to an existing workbench is required. In this case, it is easier to obtain the geometric values of the objects to be changed, then modify the workbench macro using a text editor as follows:

- Follow the first four steps for designing a workbench to open an existing workbench with a desired layout of a new workbench.
- Display the geometric values of a modified draw-window or toolbox by clicking in it with MsBtn#3.
- Repeat the last step until the geometric values of all modified objects are printed.
- Copy the values to the workbench macro using a text editor and save it.

Chapter 4:

Macros and NMR Data Files

4.1 Introduction

PAW has ten types of macros that are user-definable. Except for the system macros, each of the macros has a specific format and filename extension. This chapter describes different types of files (including macros) and the file-management operations.

Note that, in PAW, there is no difference in the file formats between time-domain and frequency-domain NMR data files. The term 'NMR data' is used for data in both the time and frequency domains. Unprocessed NMR time-domain data (sets) collected from NMR spectrometers are also called the *NMR raw data* (sets), whereas processed frequency-domain data sets are often called the *NMR spectra*.

4.2 The Macros

➢ The system macros

PAW has four system macros that are stored in the system directory (specified by a statement in **PAW.Init**). Each of them has been discussed in Chapter 2, and they are named by **'PAW.*'**, as listed below.

- **PAW.MenuBar** — a macro to produce the menu bar
- **PAW.Init** — a macro to initialise basic variables
- **PAW.Colours** — a macro to define colours for various objects
- **PAW.Hotkeys** — a macro to define hotkeys

The system macros are run by PAW's compiled routine, and hence their filenames must not be changed.

➢ The basic-parameter macros

These are named with an extension[1] of the form **mDparm**, where **m** indicates the spectral dimension, e.g., **noesy150.2Dparm**. The initial part of the filename must be consistent with that of the corresponding NMR data file, which in this case is **noesy150.2Ddata**.

The **mDparm** macros are programs created in the file conversion process. They contain statements that define the basic parameters required for displaying and processing the corresponding NMR data. A typical example is as follows:

```
# ****************************************
# *      Parameter macro for PAW data      *
# ****************************************
# This file can be edited using a word processor.
# Note: The [Tab] key can be used in this file.
#
```

[1] A filename extension is a string of characters attached to the end of a filename after a period.

```
# SpecFreq is in MHz.
# SweepWidth is sw in VARIAN's system.
# NofFIDs = NofRealFIDs + NofImagFIDs.
# NofCmpPnts = np/2 for VARIAN's data.
# D1Struct,D2Struct can either be complex or real.
# DataType can either be float or long.
# FileType can either be binary or ASCII.
# FileFormat can be PAW, VARIAN, FELIX, BRUKER etc.
# FileStruct can either be series or block-matrix.
# BlockSize is used only if FileStruct is block-matrix.
#
SpecFreq          = 600.000
SweepWidth        = 6613.750
NofFIDs           = 512
NofCmpPnts        = 1024
D1Struct          = c
D2Struct          = r
DataType          = f
FileType          = b
FileFormat        = p
FileStruct        = s
BlockSize         = -1
XPPMofCalibPnt    = 4.700
YPPMofCalibPnt    = 4.700
XofCalibPnt       = 510.418
YofCalibPnt       = 511.050
```

Here, the last four lines contain values set from a calibration process, as described in Chapter 5. Initially, they were 0.0.

➢ The processing-parameter macros

These are named with an extension of the form **mDproc**. The naming convention for them is the same as that used by the **mDparm** macros.

The **mDproc** macros are user-created programs. They do not process data but define the parameters and commands required for processing the NMR data. Typically, they are similar to the following:

```
# 2d processing parameters for Caerin41noesy150
MD1          = 1024;
MD2          = 1024;
ApodCom1     = "nsb (512, 90, 2, -200)"
ApodCom2     = "nsb (512, 90, 2, -200)"
PhaseCom1    = "ph (-73, -21976, 1023)"
PhaseCom2    = "ph (12, 29, 883)"
BsLnCorrCom1 = "n"
BsLnCorrCom2 = "n"
DimNoToTrans = b
ExprmtType   = t
sbs (4, 340,480, 44,140, 900,936, 960,1020)
# SetBslineSegments(NofSegms, Segm1L,Segm1R, ...)
```

Although the order of the statements is changeable, an **mDproc** macro must contain all the statements in the example. The values for the processing parameters in an **mDproc** macro are obtained from a series of interactive NMR data-processing operations, as will be described later in Chapter 8. The simplest way to create one is

to copy an existing macro and then change a few values therein. Details on how to refine the values will be discussed later because of the complexity involved.

➢ The peak-list macros

PAW has three types of peak-list macros:

- the *raw-peak-list macros* that have a common extension **RPeaks**,
- the *diagonal-peak-list macros* that have a common extension **DPeaks**, and
- the *cross-peak-list macros* that have a common extension **CPeaks**.

They are produced and modified by the four *SavePkList* commands in the *Peak-Assignment Toolbox*, as explained in Chapter 7. They contain the lpi-statements to load one peak item at a time.

Note that the lpi-statement have two formats, as shown in the two examples below:

```
lpi (7.164,7.179);
lpi (2.913,0.932, "K",22,"HE", "V",18,"HG", 0,20,"","", 0.0);
```

The short format is used for the raw-peak-list and diagonal-peak-list macros, and the long format, for the cross-peak-list macros.

The parameters, from left to right, are

- the X and Y co-ordinates of a peak in ppm,
- the *AminoAcidCode1*, *ResNo1*, *AtomCode1*,
- the *AminoAcidCode2*, *ResNo2*, *AtomCode2*,
- the *LabelDistX* and *LabelDistY* in ppb (part per billion),
- the *SpecFlag*, *Notes*, and *Intensity*.

➢ The drawing-object macros

There are three types of drawing object macros:

- the line-list macros that have a common extension **Lines**,
- the rectangle-list macros that have a common extension **Rects**, and
- the text-list macros that have a common extension **Text**.

A line-list macro is produced and modified by the *SaveLines* commands in the *Drawing Toolbox*, as explained in Chapter 7. They contain a number of lli-statements to load one line item at a time, e.g.,

```
lli ( 7.164, 7.179, 7.864, 7.879)
```

Here, the values are, from left to right, the co-ordinates (x1,y1) and (x2,y2) in ppm for two end points of a line.

A rectangle-list macro is produced and modified by the *SaveRects* commands in the *Drawing Toolbox*, as explained in Chapter 7. They contain a number of lri-statements to load one rectangle item at a time, e.g.,

```
lri ( 7.164, 7.179, 7.864, 7.879)
```

Here, the values are, from left to right, the co-ordinates (x1,y1) and (x2,y2) in ppm for two opposite corners of a rectangle.

A text-list macro is produced and modified by the *SaveTexts* commands in the *Drawing Toolbox*, as explained in Chapter 7. They contain a number of lti-statements to load one text item at a time, e.g.,

```
lti ( 7.164, 7.179, "This is the text.")
```

Here, the first two values are the x and y co-ordinates of a text item in ppm.

➢ The help macros

These belong to a type of macros named with the extension **help**. They contain only the prs statements that display help messages as follows:

```
prs ("<message>")
```

The name of a help macro is self-explanatory. For example, **DrawWin.help** refers to the help macro for draw-windows. Typically, the statements in a help macro are similar to the ones below:

```
prs ("1) A draw-window can be resumed to its initial shape,")
prs ("   size and location by clicking on the grey circle at")
prs ("   the top-left corner of the window.")
prs ("2) An operation that requires multiple mouse actions")
prs ("   such as peak-picking can be cancelled by clicking")
prs ("   on the third button of the mouse while the cursor")
prs ("   is on the window.")
```

The help macros are stored in the help subdirectory under the system directory. For example, if the system directory is **/usr/people/liew/PAW/**, then the help subdirectory is **/usr/people/liew/PAW/help/**. Each of the help macros is associated with a menu button defined in **PAW.Menubar** and hence even their names can be changed.

➢ Other macros

Because the macro language can be used to perform any task, they are not restricted to the ten types described above. Other macros may contain a collection of any macro commands. The macros for building the workbenches (see Chapter 3) and processing NMR data (see Chapter 8), for example, have no restrictions in terms of their contents, formats and filenames. However, the maximum number of macro statements is 4096.

4.3 The NMR Data Files

➢ General format of the NMR data file

An *NMR data file* is, in general, composed of a *file header*, if any, followed by a number of *records*. The file header contains information such as the number of points per FID, number of records, etc. Each record may contain a *record header*, a sequence of NMR data, and a *record footer*. In addition, a complex FID containing both real and imaginary part is conventionally stored in one record.

Most NMR data are stored in a series that contains sequential signals from FID#1 to FID#2, ... to FID#*n*. Some NMR data are stored in block-matrix format that contains blocks of data in a 2D or 3D data matrix.

➢ The NMR data-file format of PAW

The PAW distribution package provides a number of NMR data files with the extension '**mDdata**' in their filenames, where **m** indicates the spectral dimension, e.g., **noesy150.2Ddata**.

A PAW NMR data file contains only 32-bit float-type data in binary format. This allows the file size to be calculated easily. For example, the size of a 512×1024 complex 2D NMR data file must be 512×1024×4×2 = 4,194,304 bytes (or four megabytes) exactly. This format is exactly the same as that of Bruker NMR data files.

The filename extension and the file format are used for both the raw (i.e. unprocessed) and processed NMR data files.

➢ Other data formats recognisable by PAW

PAW currently recognises two other NMR data formats by their extensions. These are

- Bruker NMR data files with extension 'ser',
- Varian NMR data files that have an extension 'fid',

These NMR data files can be converted into PAW's format using a file-conversion option. These include also the ser-files, because the conversion routine also creates a parameter macro for PAW, as explained next. (Any other NMR data file must be converted into one of the above formats before it is readable by PAW.).

Note that Varian and Bruker NMR data files are stored in different data directories, and are indistinguishably named **fid** and **ser**, respectively. They must be changed to names that are recognisable by PAW, e.g., **noesy150.fid**, or **noesy150.ser**.

➢ To convert NMR data files of other formats

- Choose [Convert] in the *File Menu* to open the *File-conversion Dialog 1* (Figure 4.1).

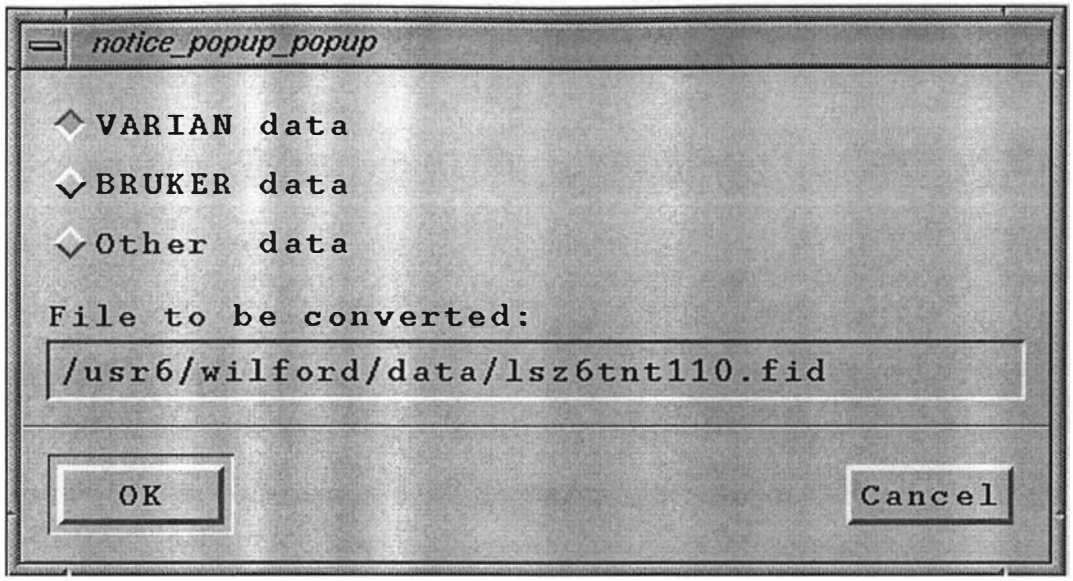


Figure 4.1: The *File-conversion Dialog 1*.

- Choose a file format from the radio-box on top of the dialog.
- Enter the name (including the path name) of the file to be converted.
- Choose [OK] to open the *File-conversion Dialog 2* (Figure 4.2):

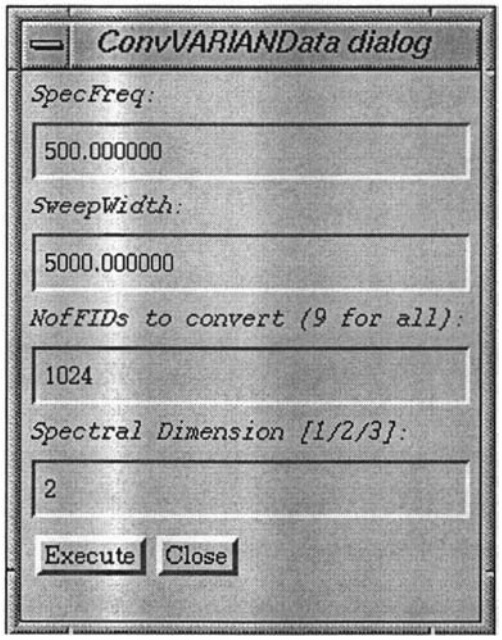


Figure 4.2: The *File-conversion Dialog 2*.

- Correct the entry items if necessary, then choose [Execute].

(For Varian data, the default values in the second dialog are set by PAW according to the information in the file header.)

Note that the value entered for the *NofFIDs to convert* is not necessary the total number of FIDs collected. For example, damaged records at the end of a 2D data set can be removed by specifying a smaller number for the entry. For a Varian NMR data

file, entering 0 for the *NofFIDs to convert* shows only the header information, and entering –1 converts the whole file.

➢ **Files created during data conversion**

The conversion process creates an NMR data file with extension m**Ddata** and a data-parameter macro with extension m**Dparm**. The original NMR data file will be unchanged. The created files will be placed at the same directory as that of the original NMR data file.

To check this,

- Change the directory to where the original file is, and then
- Type a UNIX command

```
ls -l
```

For example, if the original file is **noesy150.fid**, then, after the conversion, two other files named **noesy150.2Ddata** and **noesy150.2Dparm** will be found.

Note that any existing files with exactly the same filename as those to be created will be overwritten without warning. Different filenames must be used in the conversion if the existing files are to be kept.

4.4 Loading and Saving Files

Different types of files are loaded and saved by typing different commands or choosing different buttons. This section describes the standard method for loading and saving files. The keyboard-command methods are described individually in the related section because they are specific to different macro commands.

➢ **To load NMR data files and run macros interactively**

There are different commands for loading different types of files and macros, such as ld2 for loading 2D NMR data, ldp for saving peak lists, and so on. However, the method for loading NMR data files and running macros interactively is the same regardless of the type. For running any macro that is not associated with any menu button, the corresponding load-file command is simply 'run'.

The operations required are as follows:

- Either type a load-file command (e.g., ld2, ldp, run, etc.) or choose an option from the Load Submenu of the File Menu to open a File-selection Dialog.

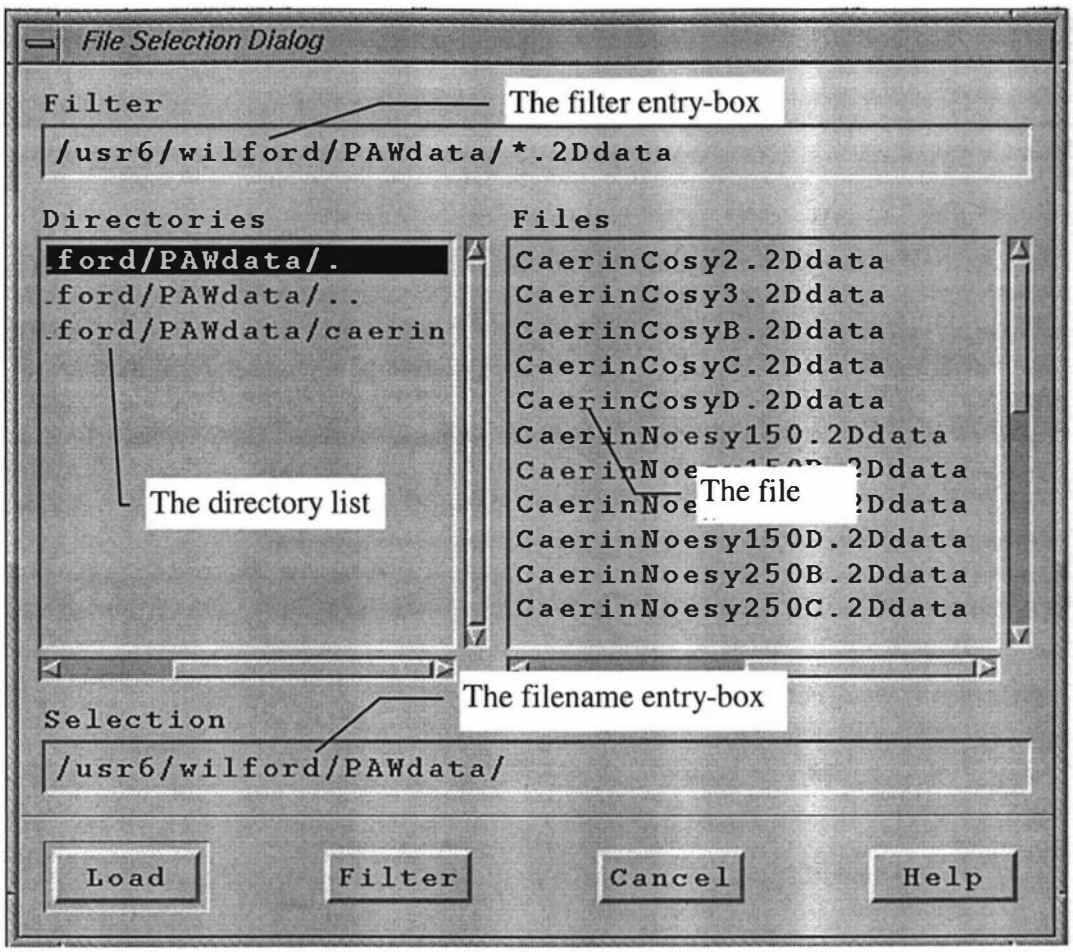


Figure 4.3: A *File-selection Dialog*.

- Double-click on a directory name on the directory list to select a directory if the filename is not in the file list. (Alternatively, change the string in the filter entry-box then choose [Filter].)
- Repeat the last step until the filename is seen.
- Double-click on the filename to load the file. (Alternatively, change the string in the filename entry-box then choose [Load].)

For example, to load a 1D NMR data file

- Type ld1 to open a dialog.
- Select a correct directory from the dialog, if necessary.
- Select the filename and choose [Load] from the dialog.

Note that every NMR data file must have a correct parameter file; otherwise the data cannot be displayed properly. In addition, if the parameters for the data to be loaded are different from those of the current data, the 1D buffers must be reset before the data is loaded. To do this,

- Choose [Reset1DBufs] in the *1D-display Toolbox*.

➢ To run macros using a macro

This requires one statement each for running a macro:

```
run <macro name>
```

Note that PAW automatically attaches a path name to a filename. The rules for the automatic path selection are as follows:

- If the filename or the path name contains '**PAW**', the default path is the system directory.
- If the filename contains '**Dparm**' or '**Ddata**', the default path is the data directory.
- If the filename contains '**.mcr**', the default path is the macro directory.
- If the filename contains '**.help**', the default path is the help directory.
- If none of the above strings appears in the path name and the file name, no default path is given.

For example, the statement below runs **PAW.color** in the system directory:

```
run PAW.color
```

(See also Chapter 3 for setting up menu buttons to run macros.)

➢ To load NMR data files using a macro

This requires two statements for loading each NMR data file:

```
run <mDparm filename>
ld1 ("<path name>", "<mDdata filename>")
```

Here, the first statement runs the parameter macro associated with the NMR data file, and the second, loads the data set. The rules for the automatic path section described in the last subsection also apply to any load-file statement such as the ld1 statement in the macro.

For example, running the macro below loads and displays the **CrnNsyRow0b.1Ddata** (Figure 4.4) found in the data directory.

```
# LoadCrnNsyRow0b.mcr
run CrnNsyRow0b.1Dparm
ld1 ("", "CrnNsyRow0b.1Ddata")
zf1
```

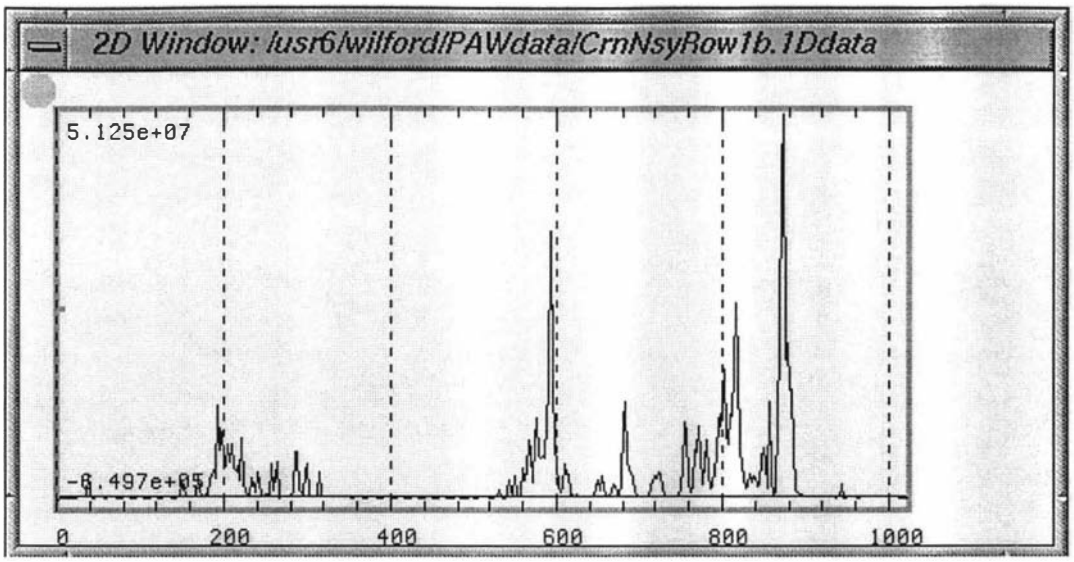


Figure 4.4: A plot of 1D NMR data loaded using a macro.

Similarly, running the macro below loads and displays the **CaerinNoesy150C.2Ddata** (see Figure 9.2 in Chapter 9) in the data directory.

```
# LoadCaerinNoesy150C.mcr
run CaerinNoesy150C.2Dparm          # Run a parameter macro
ld2 ("", "CaerinNoesy150C.2Ddata") # Load 2D data
zf2
```

➢ To save any data

The steps required to save a data set are as follows:

1. Click in the draw-window that displays the data to be saved. This operation resets the current file parameter values. Otherwise, they may be incorrect.
2. Choose [Save] in the File-menu to open the *Save Submenu.*
3. Choose an option from the menu to open a file-selection dialog that is similar to Figure 4.3 but contains a [Save] button instead of the [Load] button.
4. Double-click on a directory name in the directory list to select a directory, if necessary. (Alternatively, change the string in the filter entry-box then choose [Filter].)
5. Enter a filename in the selection box with a proper filename extension, or choose a filename in the file list with a single click to copy the filename into the filename entry box.
6. Change the filename if necessary.
7. Choose [Save].

Note that:

- Saving a file with an existing filename overwrites the existing file.

- If necessary, an **mDparm** file can be modified using a text editor.
- To avoid mistakes, NMR data file will only be created if there is data in the buffer.

Chapter 5:

Handling 1D NMR Data

5.1 Introduction

This chapter describes the management and display methods for 1D data sets and 1D buffers, including loading and displaying 1D NMR data, as well as calibrating and plotting 1D NMR spectra.

Some examples given in this chapter requires both 1D and 2D draw-windows. Also, the **CaerinNoesy150C.2Ddata** must have been loaded, as described in Chapter 4.

5.2 Loading 1D NMR Data

A 1D NMR data set can be loaded from either a 2D buffer or a 1D NMR data file. The latter can be loaded either interactively or by running a macro, as have been described in Chapter 4 of this volume and will not repeated.

This section discusses the manual method used to load 1D NMR data sets from a 2D buffer that is associated with the active 2D draw-window.

➢ To load one row (or column) from a 2D buffer using a macro

This is often required in 2D data processing. The statements are:

```
ldr (<row number>, <data type>, <buffer number>)
```

and

```
ldc (<column number>, <data type>, <buffer number>)
```

➢ To interactively load one row (or column) from a 2D buffer

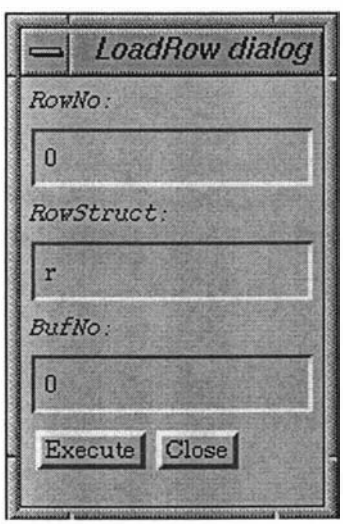


Figure 5.1: The *LoadRow Dialog*.

- Type ldr or choose [LoadRow] in the *2D-processing Toolbox* to open the *LoadRow Dialog* (Figure 5.1).
- Pick a row with MsBtn#2 then choose [Close] in the dialog. (Or, enter a row number in the dialog then choose [Execute]).

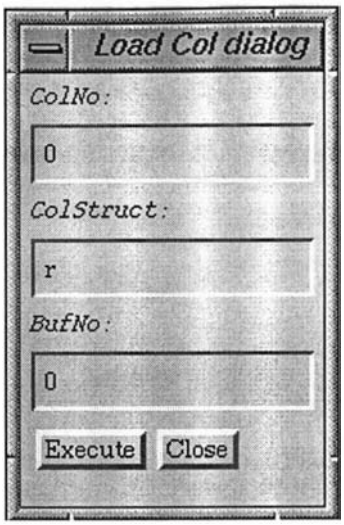


Figure 5.2: The *LoadCol Dialog*.

To load a column, the first step above should be:

- Type ldc or choose [LoadRow] (or [LoadCol] for column) in the *2D-processing Toolbox* to open the *LoadCol Dialog* (Figure 5.2).

➢ To load more than one row (or column) into different buffers using a macro

This can be similar to a macro below:

```
# Load3Rows.mcr
ldr (574, r, 2)     # Load Row 574 from a real 2D buffer
ldr (587, r, 1)
ldr (594, r, 0)
dr1                 # Display 1D data
```

Running the macro loads rows 574, 587 and 594 from a 2D buffer into 1DBuf #2, #1 and #0. However, it can only result in one buffer being displayed (Figure 5.3), because it requires a few other operations to display all three, as will be described in the next section.

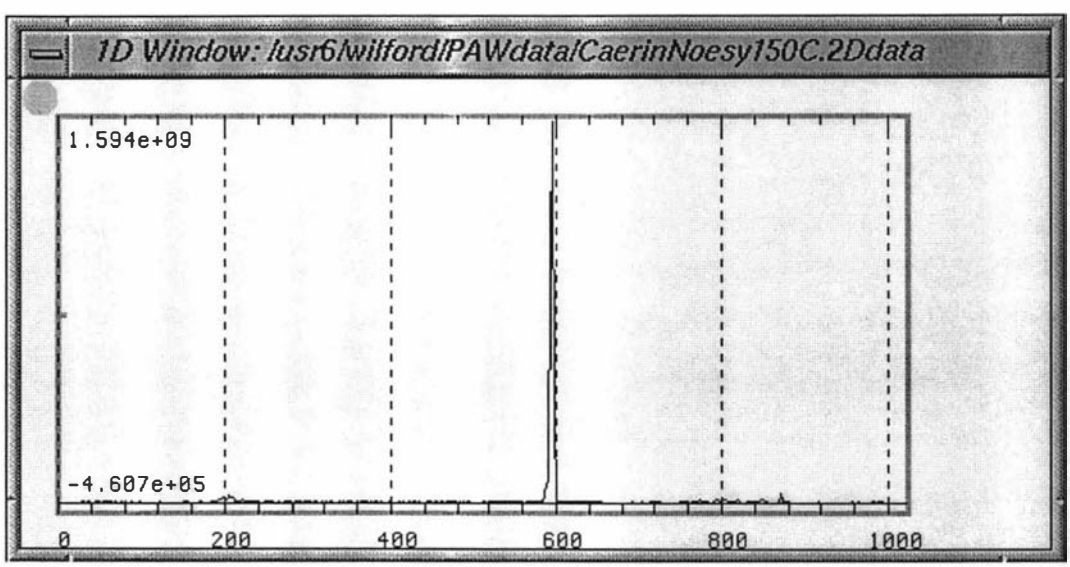


Figure 5.3: The plot for the data in 1DBuf #0 loaded by a macro.

To load columns of data, simply change the ldr statements into ldc statements.

➢ To interactively load more than one row (or column) into different buffers

The following operations load two rows from a 2D buffer:

- Type ldr or choose [LoadRow] in the *2D-processing Toolbox*.
- Click in the draw-window with which the 2D buffer is associated.
- Type lb1 to set the buffer number to 1.
- Load a row. (See the first two subsections.)
- Type lb0 to set the buffer number to 0.
- Load another row.

To load columns of data, simply change the ldr command into ldc.

➢ To load a row of 1D data from an mD data file using a macro

This requires two statements:

```
fsk (<file number>, <number of bytes to skip>)
frd (<file number>, <number of bytes to read>)
```

Here, the fsk statement skips to a position of the file specified by the number of bytes to skip, and frd reads a number of bytes from the file specified by the number of bytes to read.

Since a file must at least be opened before it can be read, a macro must contain more statements, as shown in an example below.

```
run caerin/data7noesy.2Dparm
fcl (2)
fop ("", "caerin/data7noesy.2Ddata")
fsk (2, D1*3)
frd (2, D1)
fcl (2)
zf1
dr1
```

The macro performs the following operations:

- Run the **2Dparm** file that is associated with **data7noesy.2Ddata**.
- Close File #2 before opening it for a new file.
- Open the NMR data file for reading. (The default destination is File #2.)
- Skip D1*3 bytes (namely, rows #0, #1 and #2) from the beginning of the file.
- Read D1 bytes (i.e., Row #3) from the position after skipping.
- Close File #2 again.
- Set the zooming pattern as a fully zoomed view.
- Display the full 1D data set, as shown in the next diagram.

Note that, in PAW, File #1 is reserved for the program when loading data.

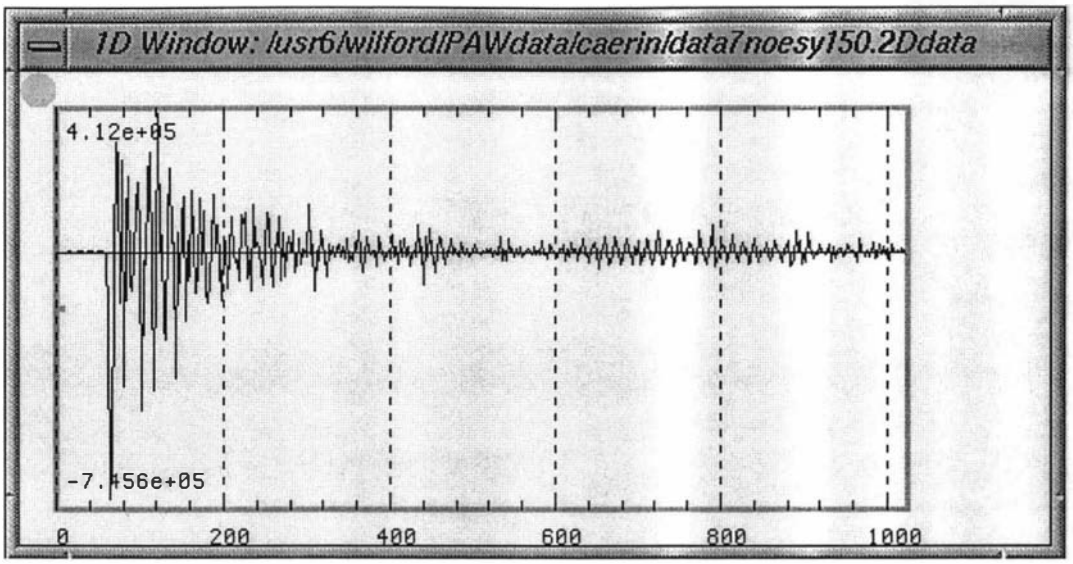


Figure 5.4: A row of NMR data drawn from a 2Ddata set.

5.3 Displaying 1D NMR data

5.3.1 The toolboxes for 1D display

There are two toolboxes that are related to 1D display: the *1D-display Toolbox* and *Common-display Toolbox*.

The *1D-display Toolbox* (Figure 5.5) contains two checkboxes and a number of command buttons. The *1DBufsDisplayed Checkbox* contains switches for setting the buffers to be displayed if the [DspBufs] switch is on. The *1DDisplayMode Checkbox* contains switches for setting various 1D-display modes. The command buttons provides tools to view different regions of 1D data sets, and options to reset 1D buffers and various parameters. (Information on displaying the 1D buffers can be found in Section 5.3.3.)

Figure 5.5: The *2D-display Toolbox*.

The *1D-display Toolbox* (Figure 5.5) contains two checkboxes and a number of command buttons. The *1DBufsDisplayed Checkbox* contains switches for setting the buffers to be displayed if the [DspBufs] switch is on. The *1DDisplayMode Checkbox* contains switches for setting various 1D-display modes. The command buttons provides tools to view different regions of 1D data sets, and options to reset 1D buffers and various parameters. (Information on displaying the 1D buffers can be found in Section 5.3.3.)

The *Common-display Toolbox* (Figure 5.6) contains switches for both 1D and 2D displays.

Figure 5.6: The *Common-display Toolbox*.

➢ To open the *1D-display Toolbox* (o1d)

- Type o1d or choose [1DDspTbox] from the [Display] menu.

The entries in the dialog are as follows:

- Entry 1 specifies the name of the output postscript file.
- Entry 2 specifies the plot header shown on top of the plot.
- Entry 3 specifies the width of the output plot in mm. (The height will be calculated from the shape of the plot on the screen.
- Entry 4 specifies if the output is to be sent to the printer. (Note that this option also produce a postscript file.)

➢ **To close the *1D-display Toolbox* (x1d)**

- Type x1d or double-click on the window-control button of the toolbox.

The method used to open and close the *Common-display Toolbox* have been described in Chapter 2 of this volume and will not be repeated. Currently, the [DspInLog] and [SwitchColor] buttons only affect 2D displays. They are there also for the future development of 3D modules.

5.3.2 Viewing different regions of 1D data sets

In the *1D-display Toolbox*, there are three buttons for viewing different regions of 1D data sets, and one for re-drawing. The following operations require a 1D draw-window be selected first (by clicking on it). If a series of 1D operations are performed for the same 1D draw-window, only one click is enough.

➢ **To display the full extend of the 1D buffers (zf or zf1)**

- Type zf or choose [ZmFull1D] in the *1D-display Toolbox*.

➢ **To display a zoomed single-region of the 1D buffers (zs or zs1)**

- Type zs or choose [ZmSngl1D] in the *1D-display Toolbox*.
- Block a region with MsBtn#1.

➢ **To display multi-regions of the 1D buffers (zm or zm1)**

- Type zm or choose [ZmMult1D] in the *2D-display Toolbox*.
- Block a few regions with MsBtn#1.
- Type dr or choose [Draw1D] in the toolbox.

Note that if any two blocked regions are overlapping, they are combined into one.

For example, the following operations give different views of the three rows loaded in the last section.

- Click in the 1D draw-window.
- Type zs.
- Block the region from points 140 to 250 (approximately) with a rubber rectangle formed by the click-and-drag method using MsBtn#1.

The plot will be re-drawn to show the region (Figure 5.7).

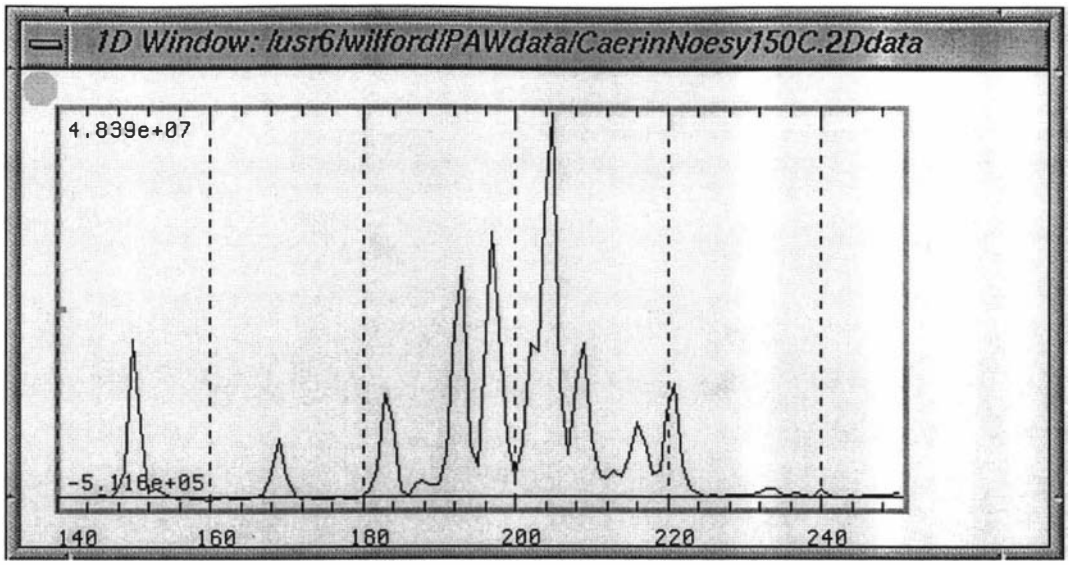


Figure 5.7: A single-region view of a 1D spectrum

Next

- Type zf and then zm.
- Block two regions, say, [120, 320] and [700,900], respectively;
- Type dr.

These operations display a multi-region view of the 1D data (Figure 5.8).

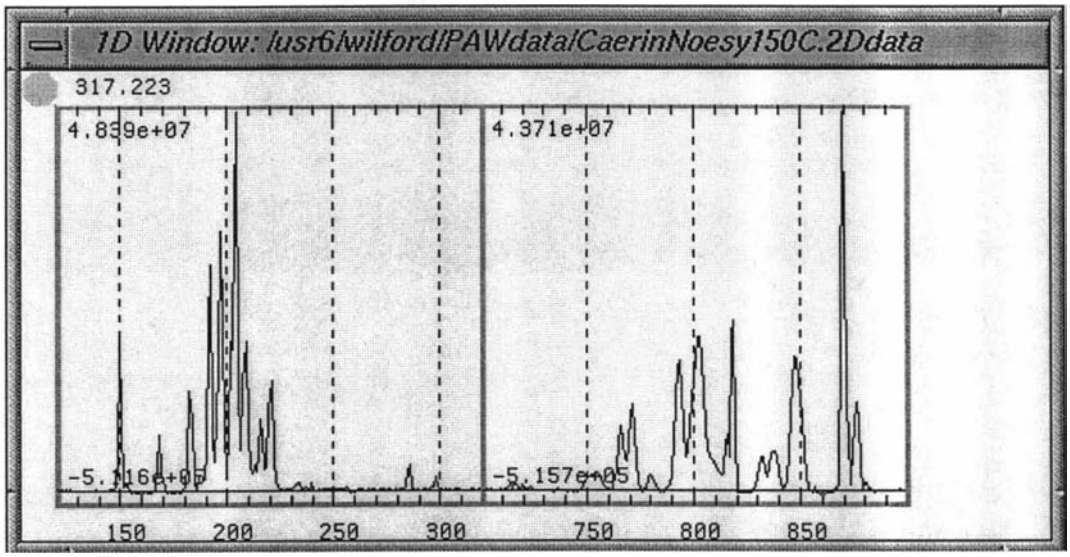


Figure 5.8: A multi-region view of a 1D spectrum

5.3.3 Displaying 1D buffers in different modes

This section describes the use of switches in the *1DDisplayedBuf Checkbox* and *1DDspMode Checkbox* of the *1D-display Toolbox*. For simplicity, it is assumed that three rows of data have been loaded into 1D buffers as described in the last section.

➢ To display more than one 1D buffer

- In the *1DDisplayedBuf Checkbox*, switch on the numbers corresponding to the buffers to be displayed (e.g., 0, 1 and 2).
- Type dbf or choose [DspBufs] in the *1DDspMode Checkbox* to display the buffers (Figure 5.9).

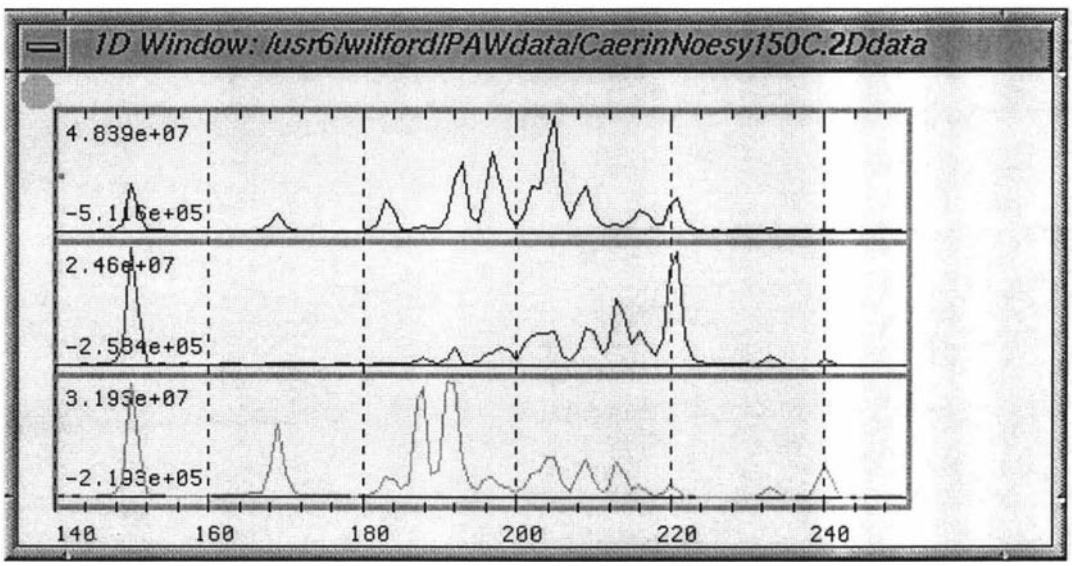


Figure 5.9: Three rows loaded from CrnNoesy150C.2Ddata.

The figures for the following examples are produced with the [DspBufs] switch toggled off.

➢ To set 1D display in the overlay mode

- Type ovl or choose [Overlay] in the *1DDspMode Checkbox* to overlay the plots (Figure 5.10).

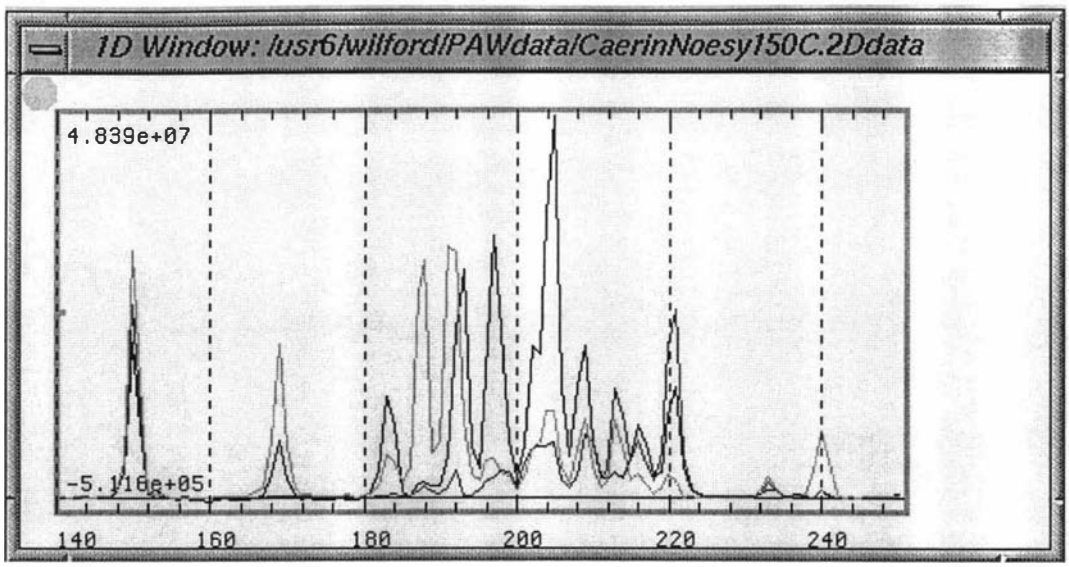


Figure 5.10: An overlaying view of three rows loaded from CrnNoesy150C.2Ddata.

➢ To set 1D display in the Equalise mode

- Type eql or choose [Equalise] in the *1DDspMode Checkbox*. The message "Use MsBtn#2 to set a peak position." will appear on the Unix shell.
- Pick a peak (say, the left most peaks centred at around point 150) with MsBtn#2 to scale the displayed plots so that they have the same height at that point (Figure 5.11).

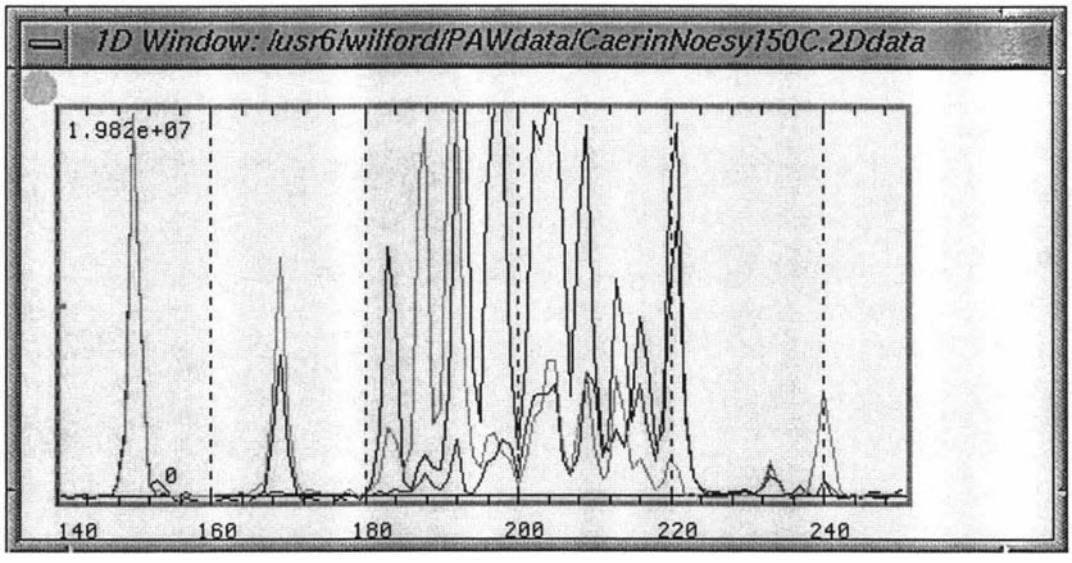


Figure 5.11: An equalised view of three rows loaded from CrnNoesy150C.2Ddata.

Note that, to avoid confusion, the [Equalise] switch will be turned off automatically as soon as the display is performed.

The figures for the following examples are produced with both the [Equalise] and [Overlay] switches toggled off.

➢ To set 1D display in the CentreY mode

- Type cty or choose [CentreY] in the *1DDspMode Checkbox* to move the *y* origin to the vertical centre (Figure 5.12).

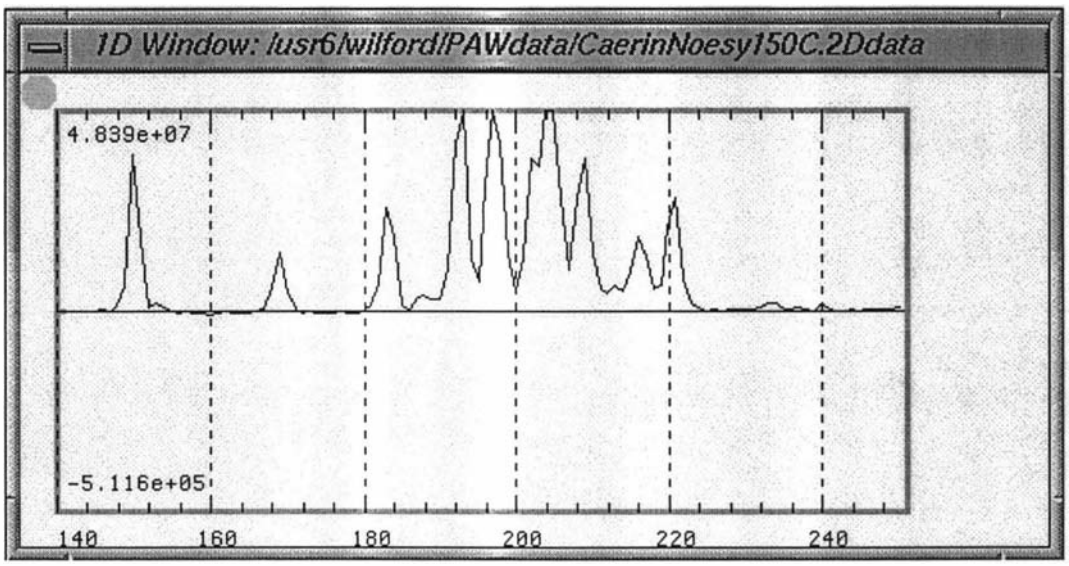


Figure 5.12: A centralised view of three rows loaded from CmNoesy150C.2Ddata.

➢ To display the imaginary data in the 1D buffers

- Switch off the [RealOnly] button in the *1DDspMode Checkbox*.

The imaginary data are all zeros in this case, as shown in Figure 5.13.

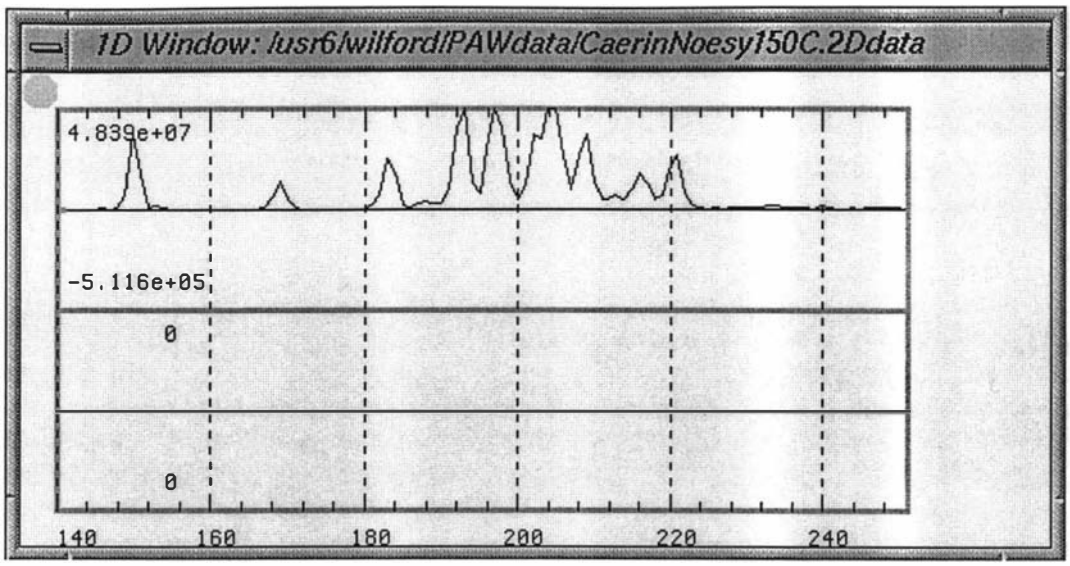


Figure 5.13: A real-only view of three rows loaded from CrnNoesy150C.2Ddata.

The buttons [DspFltr] and [DspBsln] are used for 1D processing, which will be described in Chapter 7.

5.3.4 Scaling and shifting 1D Display

This section describes the methods used to set the scale factor for the 1D display and to position the *x* and *y* origins.

The figures for the following examples are produced with the [RealOnly] switch toggled on.

➢ To set the YScale of the 1D display

- Type ysc or choose [SetYScale] in the *1D-display Toolbox*.

➢ To set the *x*-origin

- Type shx or choose [SetXShift] in the *1D-display Toolbox*.

➢ To set the *y*-origin

- Type shy or choose [SetYShift] in the *1D-display Toolbox*.

For example, the following operations set a scale factor for the display of the data:

- Type ysc to open the *SetYScale Dialog* (Figure 5.14).

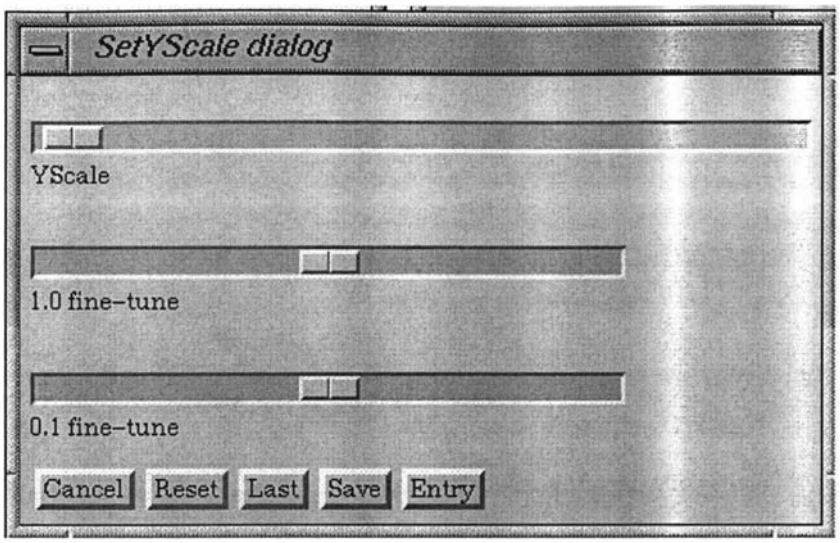


Figure 5.14: The *SetYScale Dialog*.

- Adjust the fine-tuning slider 1 in the dialog to 3. The plot in the 1D draw-window will be re-scaled (Figure 5.15).

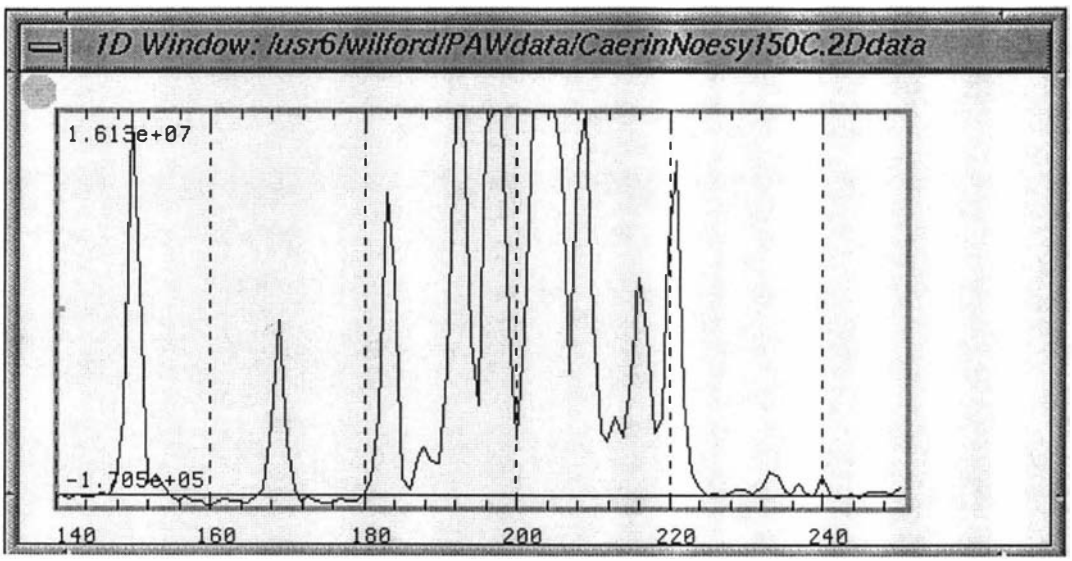


Figure 5.15: A 1D plot with Yscale set to 3.

Now,

- Reset the YScale to 1 by choosing [Reset] in the *SetYScale Dialog*.
- Choose the [SetXShift] option in the *2D-display Toolbox* to open the *SetXShift Dialog* (Figure 5.16).

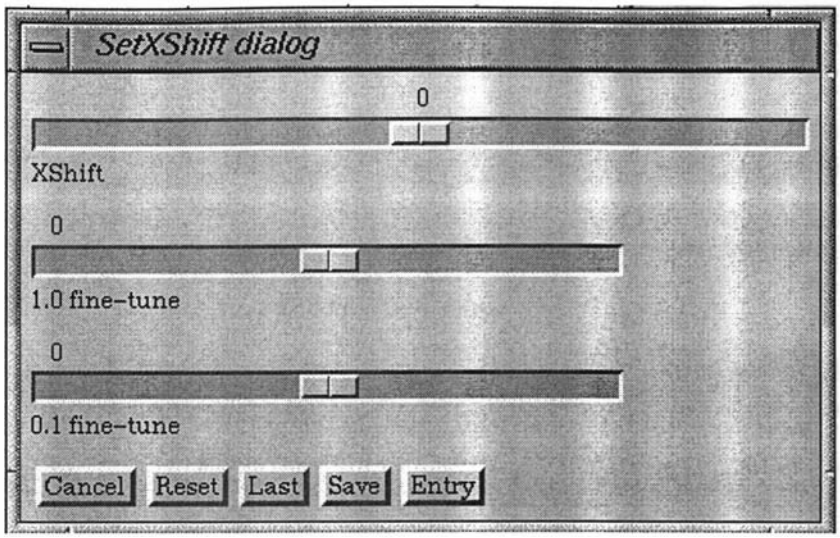


Figure 5.16: The SetXShift dialog.

- Adjust the sliders in the box to set the XShift value.

The plot in the 1D draw-window will be shifted horizontally (Figure 5.17).

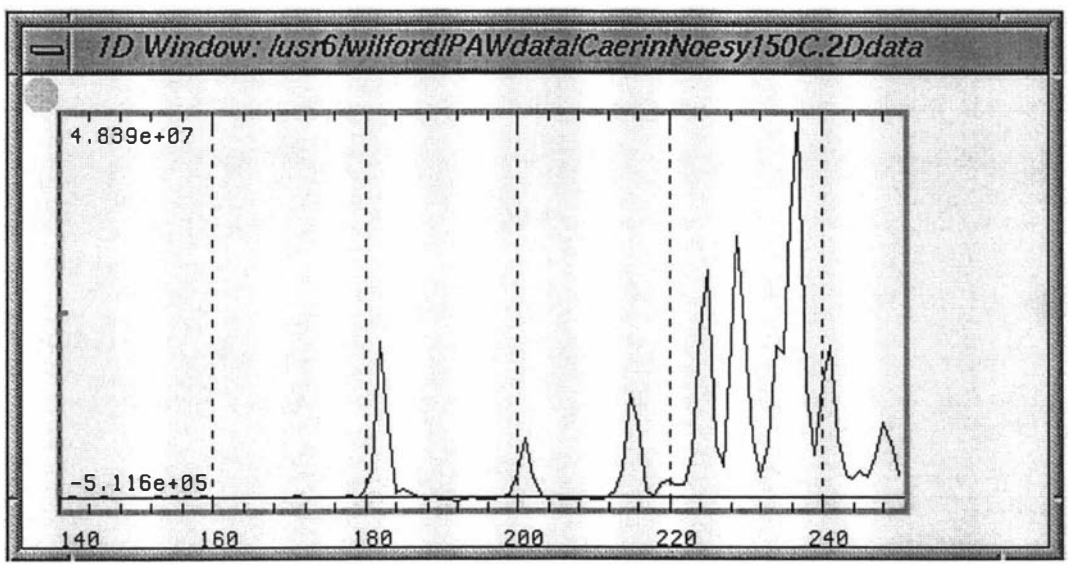


Figure 5.17: A 1D plot with the *x* origin shifted.

Next,

- Reset the XShift to 0 by choosing [Reset] in the *SetXShift Dialog*.
- Choose the [SetYShift] option in the *2D-display Toolbox* to open the *SetYShift Dialog* (Figure 5.18).

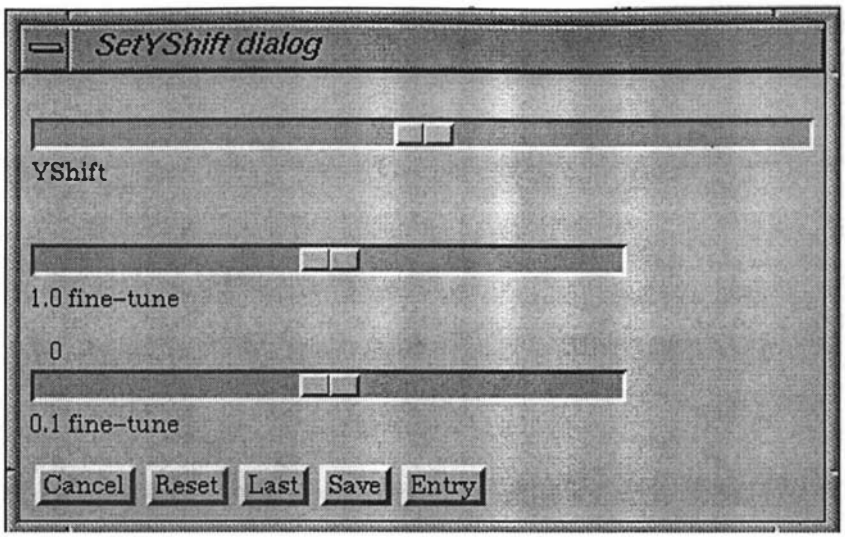


Figure 5.18: The SetYShift dialog.

- Adjust the slider in the box to set the YShift value.

The plot in the 1D draw-window will be shifted vertically (Figure 5.19).

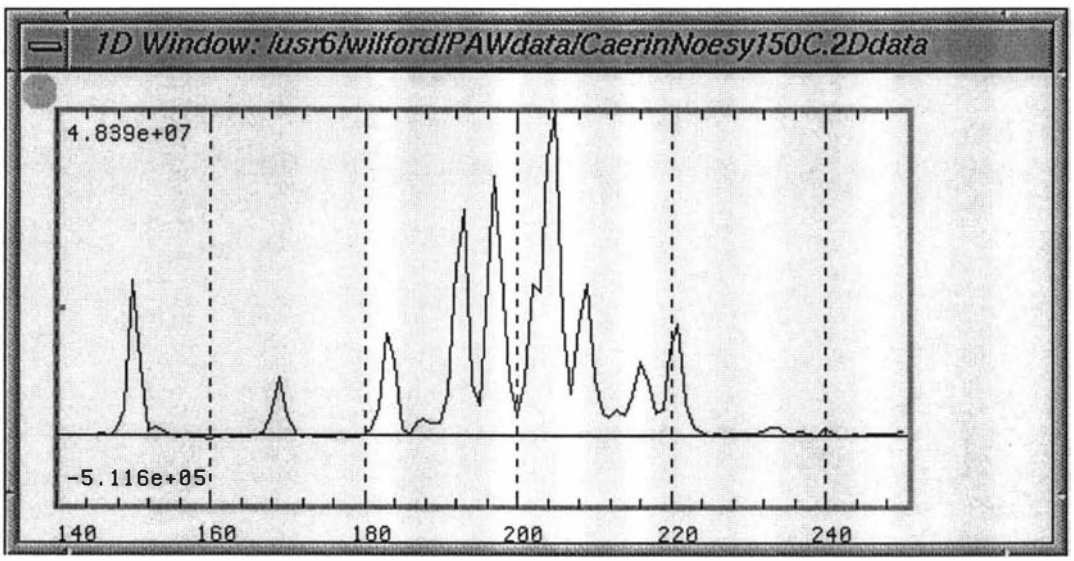


Figure 5.19: A 1D plot with the *y* origin shifted.

For consistency in display, reset the YShift to 0 by choosing [Reset] in the *SetYShift Dialog* for the following operations.

5.3.5 Displaying 1D data in different units

➢ To display NMR data in point units

- Type pnt or choose [point] in the *Common-display Toolbox.*

➢ **To display NMR data in Hz units**

- Type hz or choose [hz] in the *Common-display Toolbox*.

➢ **To display NMR data in ppm units**

- Type ppm or choose [ppm] in the *Common-display Toolbox*.

➢ **To switch on or off the grid display**

- Type gri or choose [DspGrids] in the *Common-display Toolbox*.

5.7 Calibrating 1D Spectra

Although spectra can be calibrated by changing the calibration values in the **mDparm** macros, it is usually perform interactively. This is shown by example as follows:

➢ **To calibrate 1D data — example**

- Choose [Load 1d GMP] in the *Macro Menu* to load **gmp01.1Ddata** (Figure 5.20).

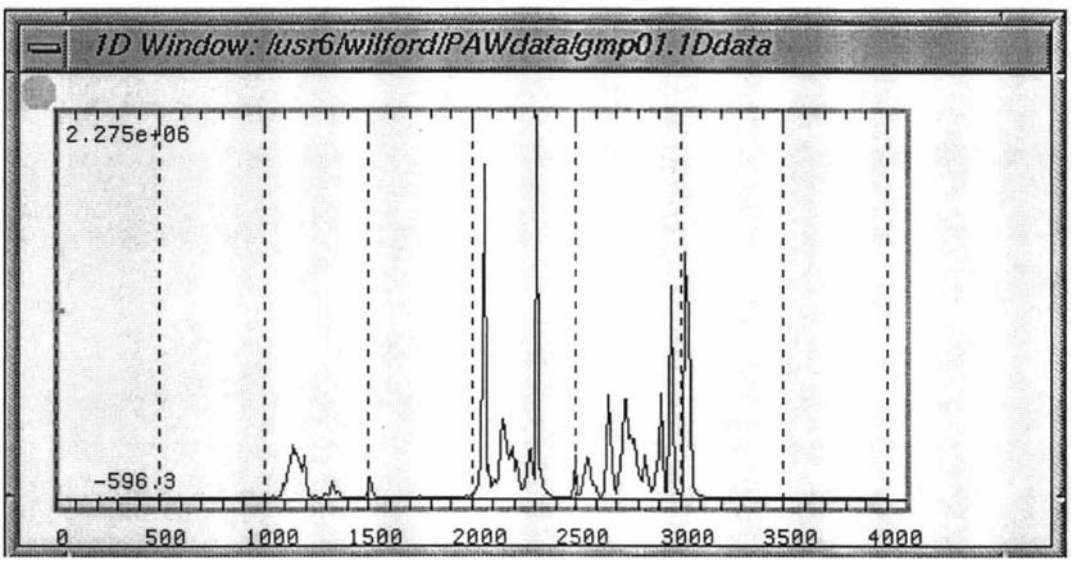


Figure 5.20: A full-view plot of 1D GMP.

- Type ocb or choose [Calibration] in the *Display Menu* to open the *Calibration Dialog* (Figure 5.21).

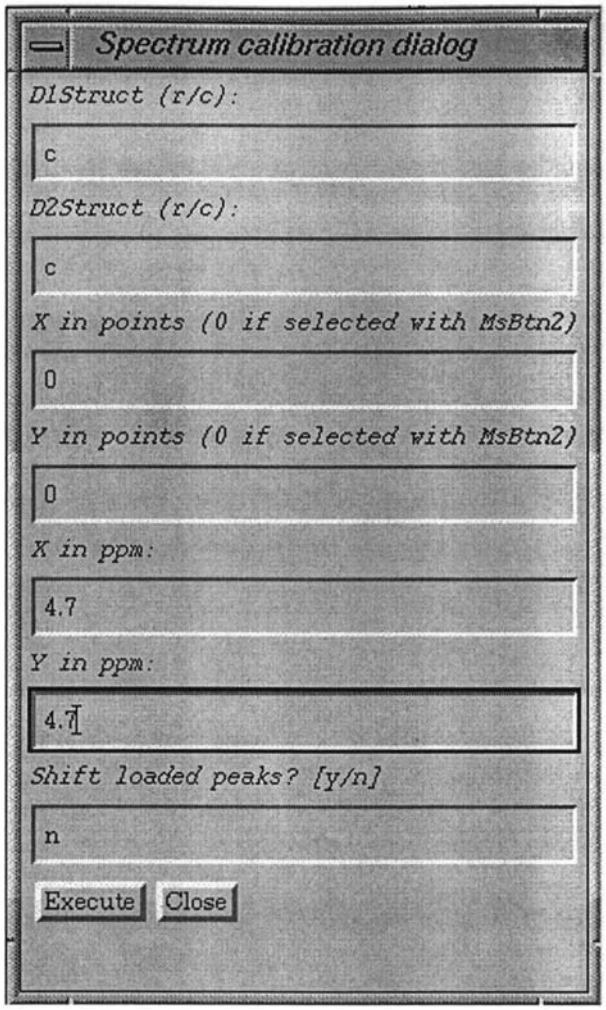


Figure 5.21: The *Calibration Dialog.*

The dialog is designed for 2D calibration, in which the entries are as follows:

- The D1Struct and D2Struct are the data structures in D1 and D2, for which the valid entries are either r for real or c for complex.
- The X in points and Y in points are the X and Y coordinates of the calibration point in units of points. If the values in the dialog are non-zero, they define the locations; otherwise the locations are the *x* and *y* coordinates of a point picked with MsBtn#2 on the 1D draw-window. (Note that picking a point outside the frame at the bottom sets the Y location to zero.)
- The X in ppm and Y in ppm are the X and Y coordinates of the calibration point in ppm.
- The entry for Shift loaded peaks? [y/n] is a flag that is only used for 2D calibration. Being able to shift loaded peaks in the calibration is very useful in spectral assignment, as will be described later.

For 1D calibration, the values for the Y in points and Y in ppm can be ignored. Now,

- Type zs, then block a region from points 2000 to 2100, as shown in the next diagram.

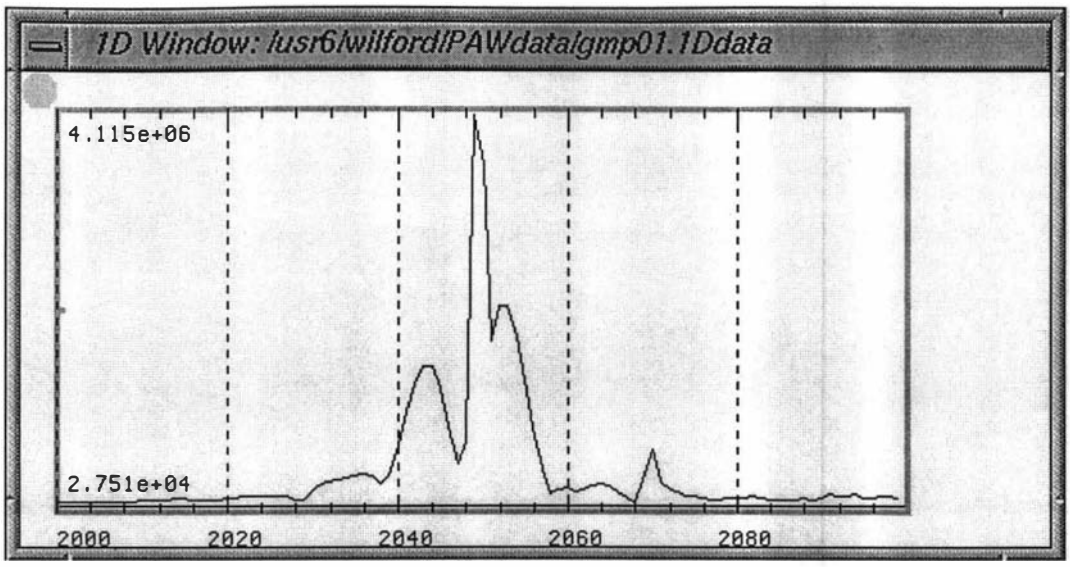


Figure 5.22: An expanded plot for 1D calibration.

- Pick the water peak centred at Point 2048 (approximately) with MsBtn#2.
- Enter 0 for the X in points, and 4.75 for the X in ppm; then choose [Execute] in the dialog.
- Switch on the [ppm] button in the *Common-display Toolbox*.

The scale for the plot will then be in ppm, as shown in Figure 5.23.

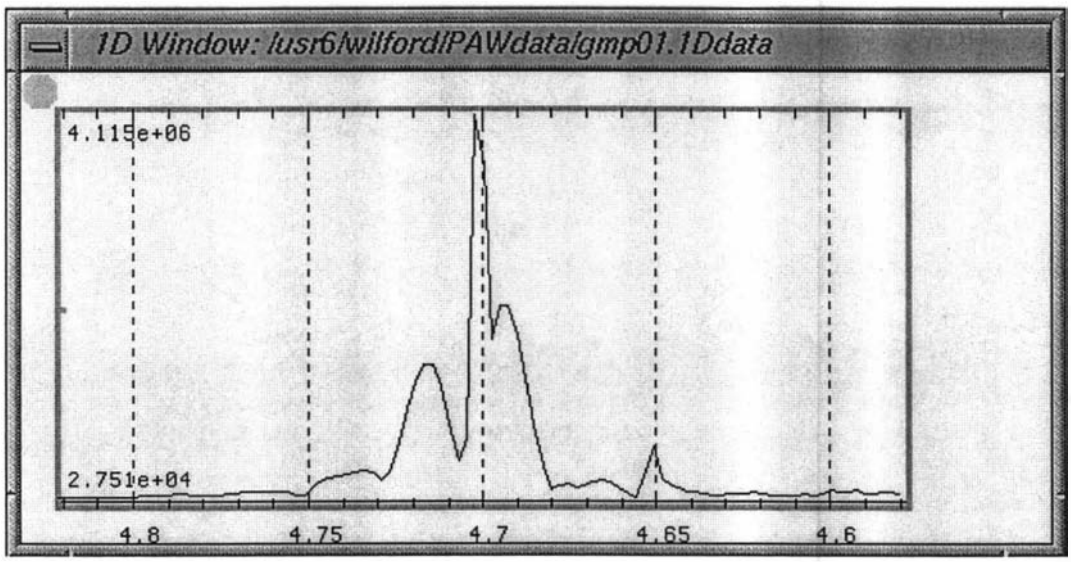


Figure 5.23: A 1D spectrum displayed in ppm.

➢ Auto-upgrade of the calibration values

When manual calibration is finished, the associated **1Dparm** macro is simultaneously upgraded. For example, the contents of **gmp01.1Dparm** after the above operation will become:

```
SpecFreq          =  500.000
SweepWidth        = 5000.000
```

```
NofFIDs            = 1
NofCmpPnts         = 4096
D1Struct           = c
D2Struct           = r
DataType           = f
FileType           = b
FileFormat         = P
FileStruct         = s
BlockSize          = 32
XPPMofCalibPnt     =    4.700
YPPMofCalibPnt     =    0.000
XofCalibPnt        = 2048.139
YofCalibPnt        =    0.000
```

It is recommended that spectral calibration be done interactively to obtain better results with the corresponding **mDparm** macro automatically upgraded. Note that the previous **mDparm** file that contains previous calibration results will be overwritten following the above operation. So if the previous results are to be kept, the file must be backed up first.

5.8 Plotting a 1D Display

In PAW, plotting produces a postscript file corresponding to the display on a current draw-window, along with any annotations. It can also send the result directly to a PostScript printer.

➢ To plot a 1D display

- Type pl1 or choose [Plot1D] in the *1D-display Toolbox* to open the *1D-plotting Dialog* (Figure 5.24).

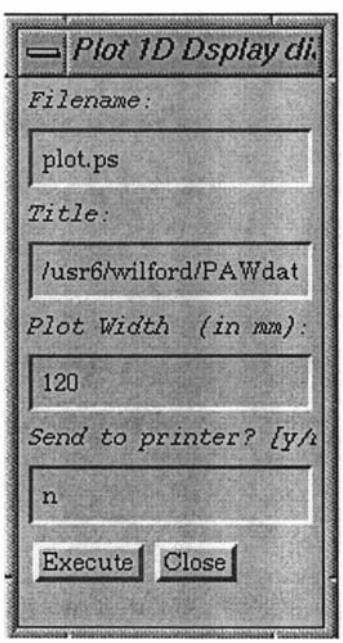


Figure 5.24: The *1D-plotting Dialog*.

The entries in the dialog are as follows:

- Entry 1 specifies the name of the output postscript file.
- Entry 2 specifies the plot header shown on top of the plot.
- Entry 3 specifies the width of the output plot in mm. (The height will be calculated from the shape of the plot on the screen.
- Entry 4 specifies if the output is to be sent to the printer. (Note that this option also produce a postscript file.)

Figure 5.25 shows a printed plot of the **gmp01.1Ddata**.

/disk2/people/wilford/PAWdata/gmp1d01.pdt

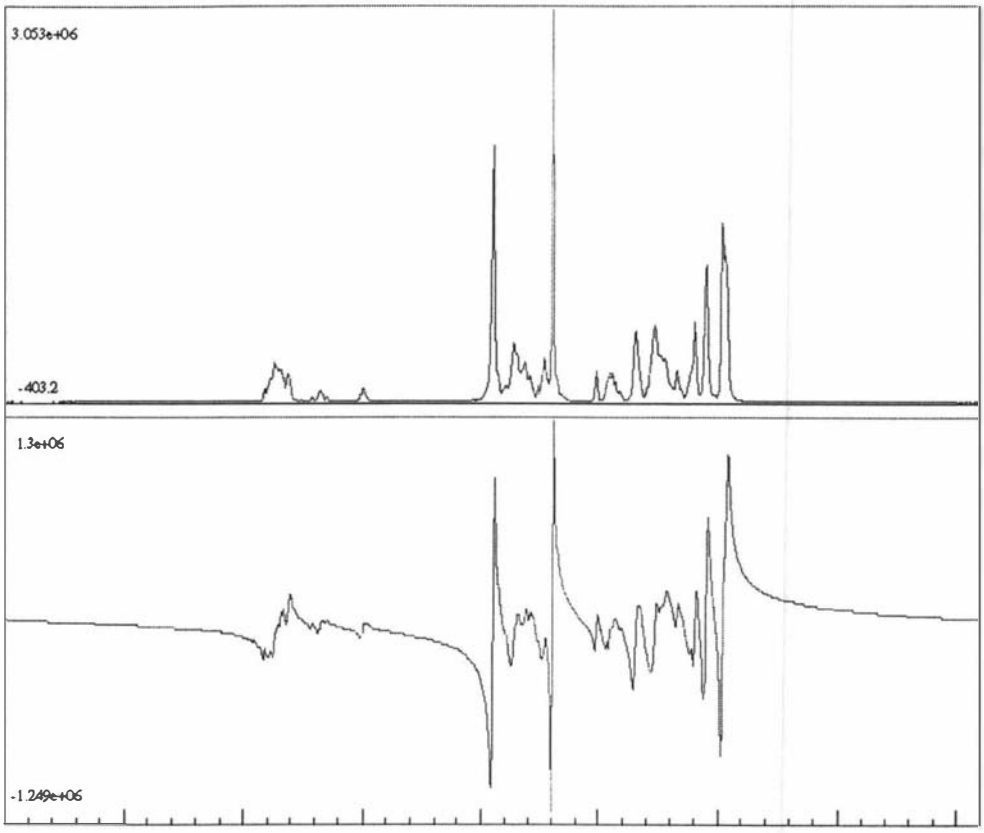


Figure 5.25: A printout of a 1D spectrum

Note that

- PAW produces coloured plots with no colour in the background. With a mono-colour printer, some colours may appear as dotted lines.
- Annotations (such as lines, rectangles and text) must be created and displayed on the screen before plotting.

Chapter 6:

Processing 1D NMR Data

6.1 Introduction

This chapter describes the methods used to process 1D spectra with PAW.

To avoid duplication, examples will only be given if none of the same kind are presented in Chapter 8.

To provide examples, it is recommended that a 1D spectrum stored in the **GMP01b.1Ddata** be loaded as follows:

- Choose [Reset1DBufs] in the *1D-display Toolbox*. (Note: This step is required before loading a 1D data set from a file. See Chapter 4 for the explanation.)
- Choose [Load1DData] in the *1D-processing Toolbox* to load a file-selection dialog.
- Choose **GMP01b.1Ddata** in the file list, and then choose [Load].

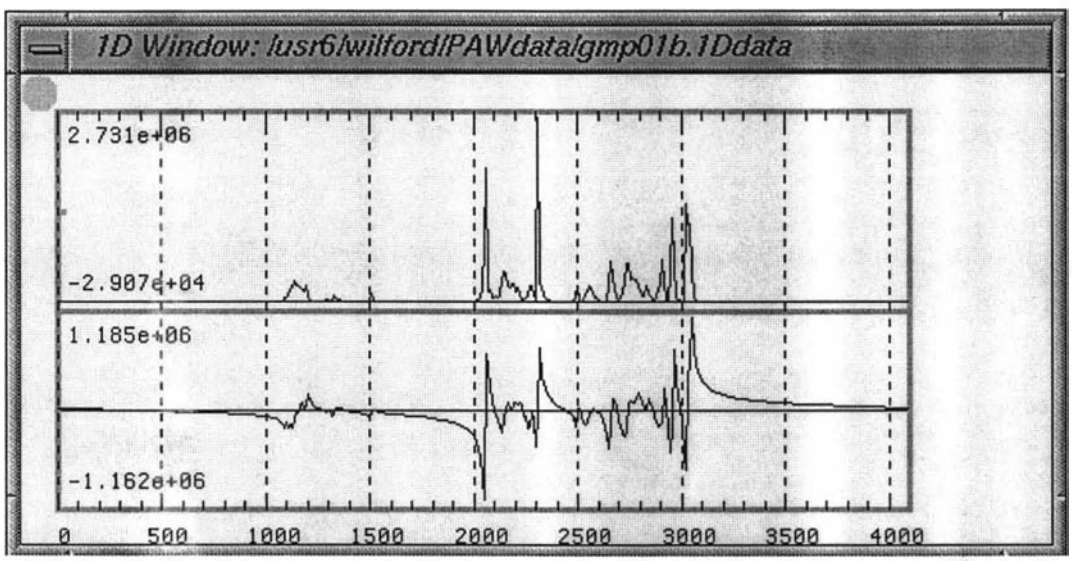


Figure 6.1: The real and imaginary plots of the **GMP01b.1Ddata**.

6.2 The *1D-processing Toolbox*

➢ The *1D-processing Toolbox*

The *1D-processing Toolbox* (Figure 6.1) contains the command buttons for common 1D-processing operations.

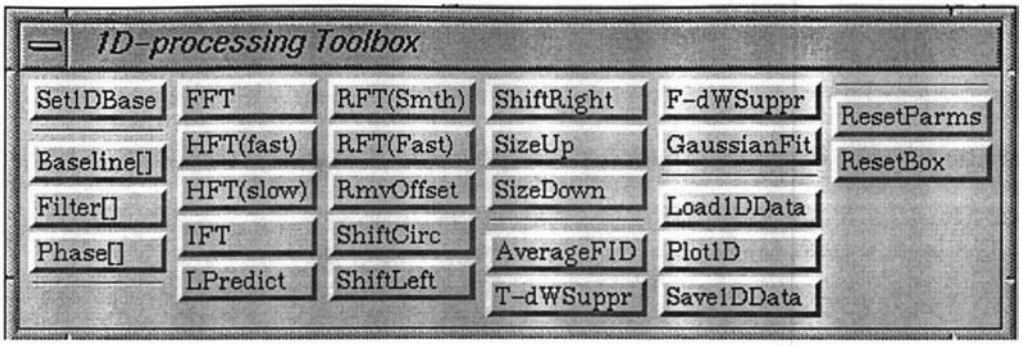


Figure 6.2: The *1D-processing Toolbox*

➢ **To open the *1D-processing Toolbox***

- Type o1p or choose [2DProcTBox] from the [Process] menu.

➢ **To close the *1D-processing Toolbox***

- Type x1p or double-click on the window-control button.

6.3 The *1D-MiscProc Toolbox*

➢ **The *1D-MiscProc Toolbox***

The *1D-MiscProc Toolbox* (Figure 6.2) contains the command buttons for unusual or seldom used 1D-processing operations.

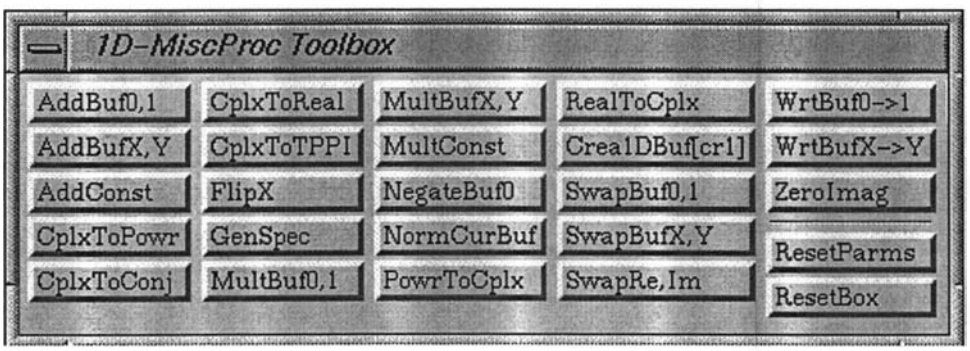


Figure 6.3: The *1D-MiscProc Toolbox*

➢ **To open the *1D-MiscProc Toolbox***

- Type o1m or choose [1DMiscTBox] from the [Process] menu.

➢ **To close the *1D-MiscProc Toolbox***

- Type x1m or double-click on the window-control button.

6.4 Generating 1D Spectra

➢ To generate a non-decaying t-domain data set

- Type gsp or choose [GenSpec] in the *1D-MiscProc Toolbox* to open the *GenSpectrum Dialog* (Figure 6.4).

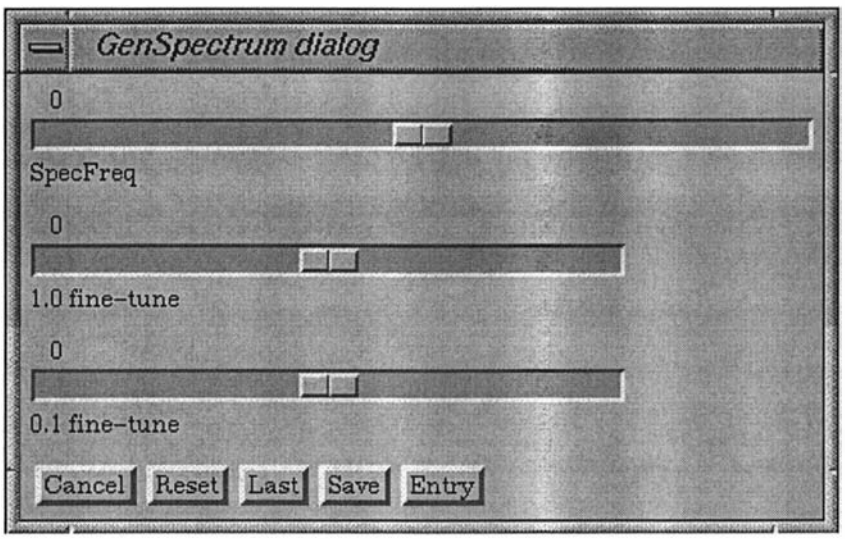


Figure 6.4: The *GenSpectrum Dialog.*

- Adjust the slider to set the desired frequency, such as the one in the next diagram.
- Choose [Save] in the dialog box.

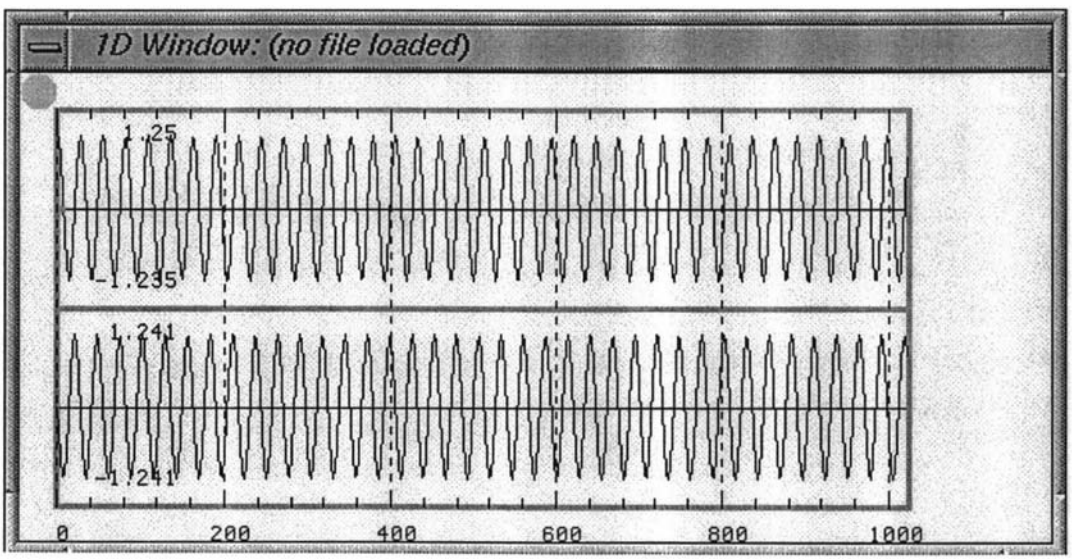


Figure 6.5: A generated 1D data set.

➢ **To produce an exponentially decaying FID**

- Type oft or choose [Filter] in the *1D-processing Toolbox* to open the *Filter Toolbox* if it is not currently opened.
- Type exp or choose [Exp] in the toolbox to open a dialog, as shown in the diagram below.
- Adjust the slider to set a desired decaying parameter to produce an ideal FID, such as the one in the next diagram.
- Choose [Save] in the dialog box.

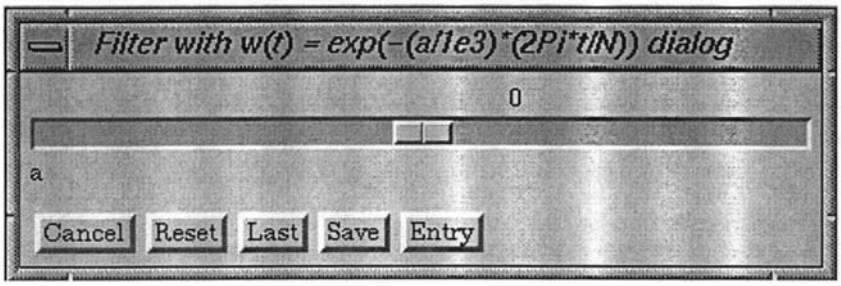


Figure 6.6: The Exp-filter dialog.

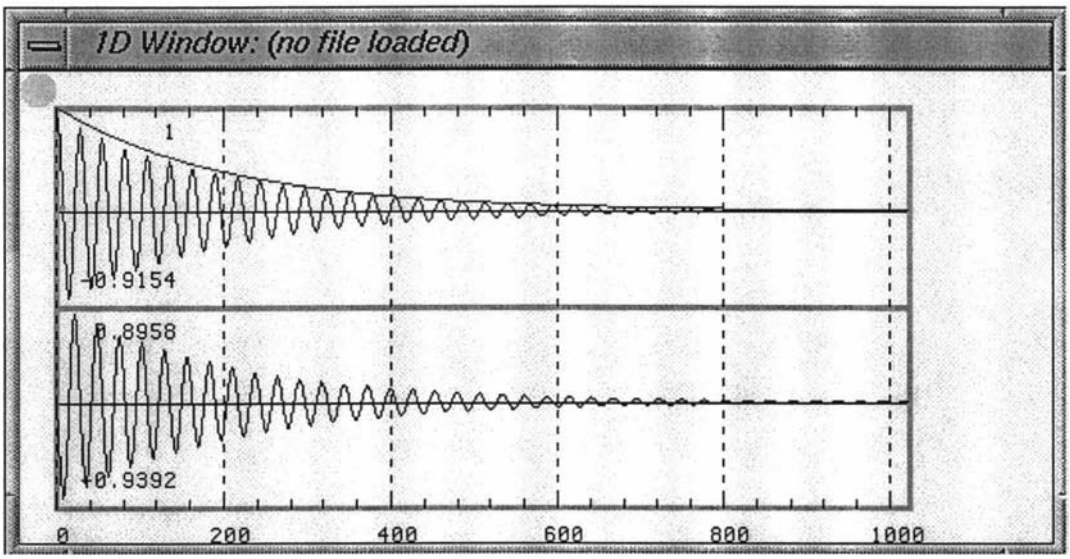


Figure 6.7: An Ideal FID

6.5 Changing the Size of 1D Data Sets

➢ **To double the size of a 1D NMR data set (by zero-filling)**

- Type sup or choose [SizeUp] in the *1D-processing Toolbox*.

➢ **To half the size of a 1D NMR data set (by trucation)**

- Type sdw or choose [SizeDown] in the *1D-processing Toolbox.*

6.6 Filtering 1D Spectra

The *Filter Toolbox* (Figure 6.7) contains command buttons for filtering t-domain data.

Figure 6.8: The *Filter Toolbox*

➢ **To open the *Filter Toolbox***

- Type oft or choose [Filter] in the *1D-processing Toolbox.*

➢ **To close the *Filter Toolbox***

- Type xft or double-click on the window-control button.

➢ **To apply a filter**

- Type a filter-command or choose a corresponding button in the *Filter Toolbox.*

➢ **To apply the enhanced sine-bell filter**

- Type esb or choose [EnhSineBell] in the *Filter Toolbox.*

➢ **The macro statements**

In a macro, a filter statement is of the form

```
<filter command> (<par 1>, <par 2>, ...)
```

A list of all filter commands can be found in Chapter 12. For example, to perform a 2nd order enhanced sine-bell filter that is shifted 90° with the EndPoint at 1024 and an enhancement of −168 peu (PAW's exponential unit — see Appendix 6.d, Volume I), it will be

```
esb (1024, 90, 2, -168)
```

In a standard 2D processing-parameter macro (see Chapter 4) for interactive processing using a dialog, two filter statements are assigned to the macro variables fltCom1 and fltCom2, e.g.,

```
fltCom1 = "esb (1024, 90, 2, -168)"
```

6.7 The Fourier Transform and Its Inverse

➢ **To perform a FFT**

- Type fft or choose [FFT] from the *1D-processing Toolbox.*

➢ **To perform an IFT**

- Type ift or choose [IFT] from the *1D-processing Toolbox.*

6.8 The Real Fourier Transform

➢ **To perform a RFT**

- Type rft or choose [RFT] from the *1D-processing Toolbox.*

Note that this operation transforms a single real function from the time domain to frequency domain, as described in Chapter 6, Volume I. Examples of the operation can be found in Chapter 8 of both volumes.

6.9 The Hilbert Transform

➢ **To perform the HT(slow)**

- Type hts or choose [HT(slow)] from the *1D-processing Toolbox.*

➢ **To perform the HT(fast)**

- Type ht or choose [HT(fast)] from the *1D-processing Toolbox.*

To provide an example, it is recommended that the **GMP01b.1Ddata** be loaded (see Section 6.1) and then copied to 1DBuf #1 as follows:

- Choose [WrtBuf0→1] in the *1D-MiscProc Toolbox.*

The operation below generates real-only data:

- Choose [ZeroImag] in the *1D-MiscProc Toolbox.* (If the imaginary part is not shown, choose [RealOnly] again to set the display mode to complex display mode.)

The operation below re-generates the imaginary part:

- Choose [HT(fast)] in the *1D-processing Toolbox* to reproduce the imaginary part.
- Choose [SetBaselevel] in the same toolbox and then use MsBtn#1 to set a region between point #0 to point #1000 as a baseline segment.
- Choose [RmvOffset] in the same toolbox.
- Switch on the [0] and [1] button in the *1DBufDisplayed Checkbox* of the *1D-display Toolbox.*

- Switch on the [DspBufs] button in the same toolbox.
- Switch off the [Dsp1DRealOnly] button in the same toolbox.

Figure 6.9 shows the result of these operations.

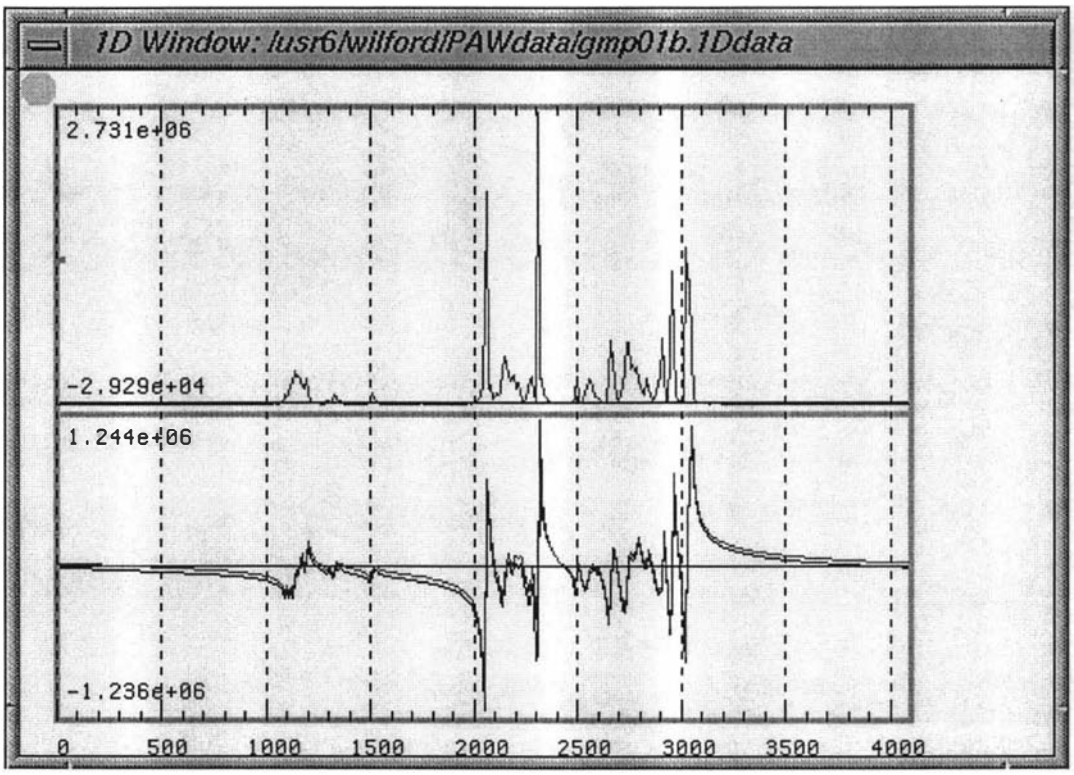


Figure 6.9: An overlay of the real and imaginary parts of the Hilbert transform result along with the original data set. (1) The real parts are exactly the same. (2) The imaginary part of the HT result is only slightly higher in this case due to a significant decay in the time-domain data.

6.10 Phasing 1D Spectra

The *Phasing Toolbox* contains a number of command buttons as shown below.

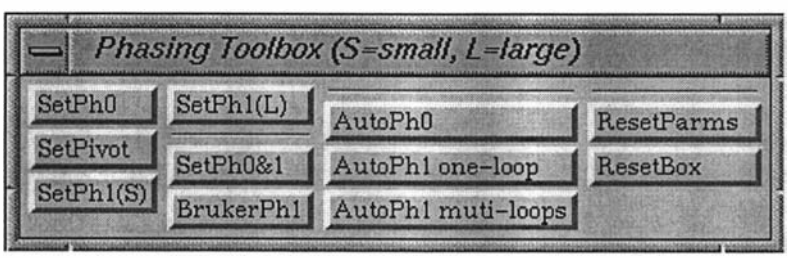


Figure 6.10: The *Phasing Toolbox*

➢ To open the *Phasing Toolbox*

- Type oph or choose [Phase] in the *1D-processing Toolbox*.

➢ **To close the *Phasing Toolbox***

- Type xph or double-click on the window-control button.

➢ **To set the zeroth-order phase correction**

- Type ph0 or choose [SetPh0] in the *Phasing Toolbox.*

➢ **To set the pivot point at which the phase remains unchanged**

- Type piv or choose [SetPivot] in the *Phasing Toolbox.*

➢ **To apply a small first-order phase correction**

- Type ph1 or choose [SetPh1] in the *Phasing Toolbox.*

➢ **To apply a large first-order phase correction**

- Choose [SetPh1(L)] in the *Phasing Toolbox.*

➢ **To set all three phasing parameters in one dialog**

- Type ph or choose [SetPh0&1] in the *Phasing Toolbox.*

➢ **To set Bruker's first-order phase correction**

- Type phb or choose [SetBrukerPh1] in the *Phasing Toolbox.*

Note that this does not require a pivot entry because it uses two predefined pivots in the correction (namely, 0 and centre of the spectrum).

➢ **The macro statement**

The single macro statement for phasing is

```
ph (<phase0>, <phase1>, <pivot>)
```

For example,

```
ph (-72. -21973. 1023)
```

In a standard 2D processing-parameter macro (see Chapter 4) for interactive processing using a dialog, two phase-correction statements are assigned to the macro variables phCom1 and phCom2, e.g.,

```
phCom1 = "ph (-72. -21973. 1023)"
```

6.11 Correcting Baseline of 1D Spectra

➢ **The baseline toolbox**

PAW provides a *baseline toolbox* for baseline correction as shown below.

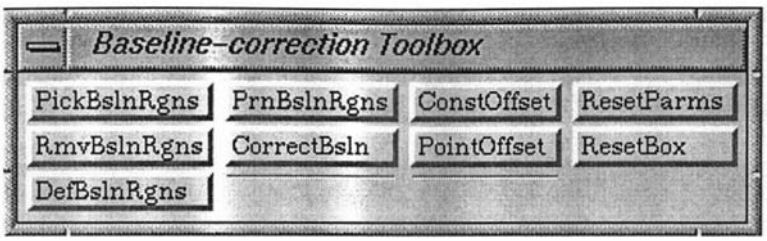


Figure 6.11: The *Baseline-correction Toolbox*

➢ **To open the *Baseline-correction Toolbox***

- Type obc or choose [Baseline] in the *1D-processing Toolbox.*

➢ **To close the *Baseline-correction Toolbox***

- Type xbc or double-click on the window-control button.

➢ **To set the baseline level**

- Type sbl or choose [SetBslnLevel] in the *Baseline-correction Toolbox.*

➢ **To pick the baseline segments**

- Type pbr or choose [PickBslnSegms] in the *Baseline-correction Toolbox.*

➢ **To remove the baseline segments**

- Type rbr or choose [RmvBslnSegms] in the *Baseline-correction Toolbox.*

➢ **To add the baseline segments**

- Type abr or choose [AddBslnSegms] in the *Baseline-correction Toolbox.*

➢ **To print the baseline segments on the Unix shell**

- Type pbr or choose [PrnBslnSegms] in the *Baseline-correction Toolbox.* (This is designed to ease the copying of parameters required for 2D processing.)

➢ **To correct the baseline**

- Type cb or choose [CorrectBsln] in the *Baseline-correction Toolbox.*

Example

To provide an example, it is recommended that the **GMP01b.1Ddata** be loaded (see Section 6.1) and then displayed with the YScale enlarged and *y*-origin centred as follows:

- Choose [SetYScale] from *1D-display Toolbox* to open a dialog.
- Adjust the slider to 30, and choose [Save].
- Choose [CentreY] in the *1D-display Toolbox.*

Figure 6.12 shows the result of the operations.

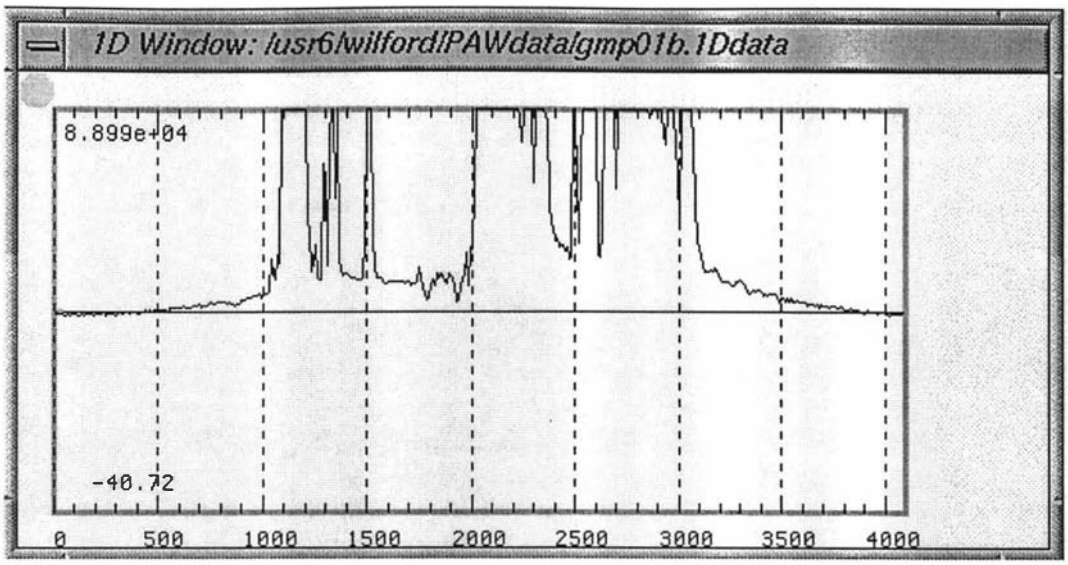


Figure 6.12: The real and imaginary plots of the **GMP01b.1Ddata** when displayed with the y-origin centred.

➢ Step 1: Setting the baseline level

- Type obc or choose [BslnCorr] in the *1D-processing Toolbox* to open the *Baseline-correction Toolbox*. (This will automatically set RealOnly=YES, and DspBuf=NO.)
- Choose [SetBsLevel] in the toolbox.
- Surround a baseline segment (say, from point 0 to 800) in the spectrum with MsBtn#1 so that PAW can calculate the baseline level.

(Note: This step is essential for correctly picking the baseline segments.)

➢ Step 2: Picking the baseline segments

- Choose [PickBslnSegms] to open a dialog as shown on the right, in which the fields are as follows:
  - The Noise Factor sets a base level that is equal to the factor's value times the noise SD.
  - The Min NofContPnts sets the minimum length of a 1D segment in which no peak is detected — otherwise, the segment is excluded from the baseline.
  - The Min PeakWidth sets the minimum length of a region in which at least one peak is detected — otherwise, the segment is included in the baseline.

Figure 6.13: The PickBsPnts dialog

- Simply accept the default values, and choose [Execute].

The detected baseline segments will then be overlaid on top of the spectrum (with the detected baseline segments displayed in different colour on the screen) as shown in the next figure.

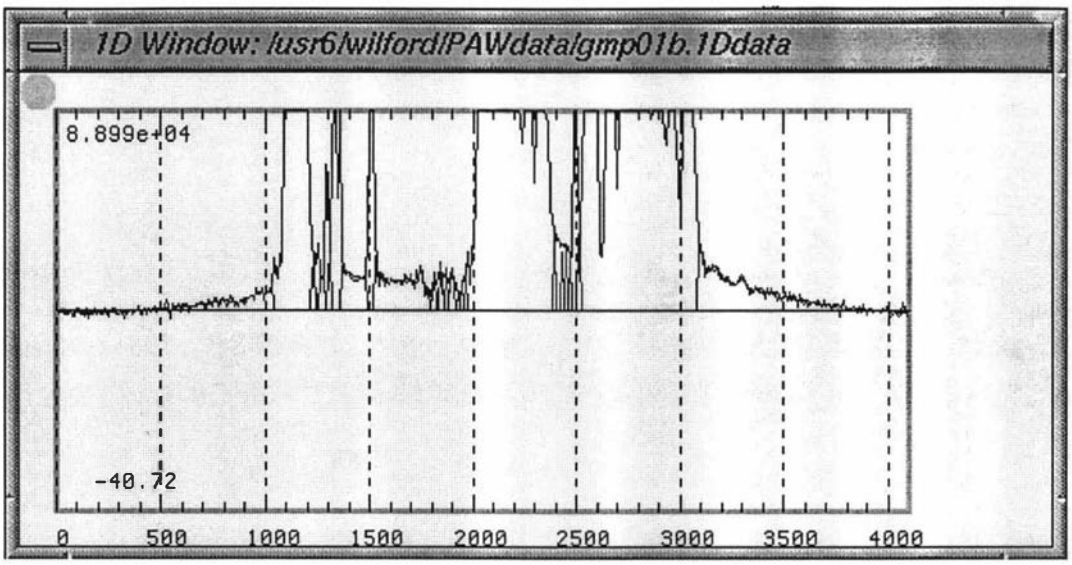


Figure 6.14: 1D spectrum overlaying with a red baseline display with YScale=50

➢ Step 3: Refining the segments

As you can see, not all segments detected are correct, because we did not use the right set of parameters to pick the baseline. It is, however, important to define the baseline segments properly, because PAW takes data sampled from the segments to fit to a polynomial for the baseline correction. Therefore, the next task is to remove those incorrect segments (or add some segments) so that PAW will have correctly defined baseline segments for the operation.

- Choose [RmvBslnSegms] in the *Baseline-correction Toolbox*; then use MsBtn#1 to set a segment (say, from 1020 to 1080) to be removed.
- Repeat the above process until the plot looks similar to the one below.

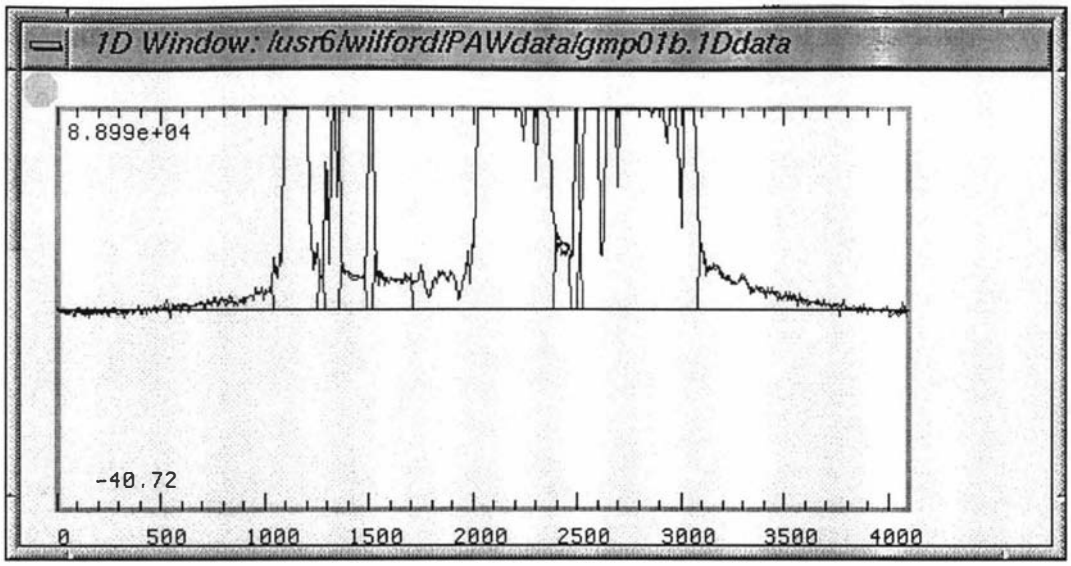


Figure 6.15: The same spectrum with correct baseline segments

To add more segments that have not been detected, you can use the [AddBslnSegms] command in a similar way to the two steps above. Note that,

- Not all baseline segments have to be included in the definition as long as there are sufficient for the correction;
- If there are several 1D spectra to process, the baseline segments do not need to be picked again (or redefined) as long as they are the same. This is useful in 2D-processing because using the auto-pick process for baseline correction may produce incorrect results.

➢ Step 4: Correcting the baseline

Finally, we can start correcting the baseline as follows:

- Choose [CorrectBsln] in the baseline toolbox to open a dialog as shown on the right, where the default is to use a polynomial of degree 5 in the fitting.
- Choose [Execute] to accept the default.

The result is shown in the next diagram.

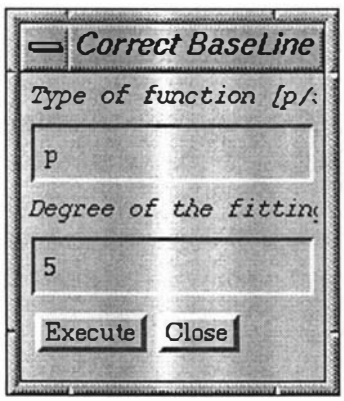


Figure 6.16: The CorrectBsln dialog

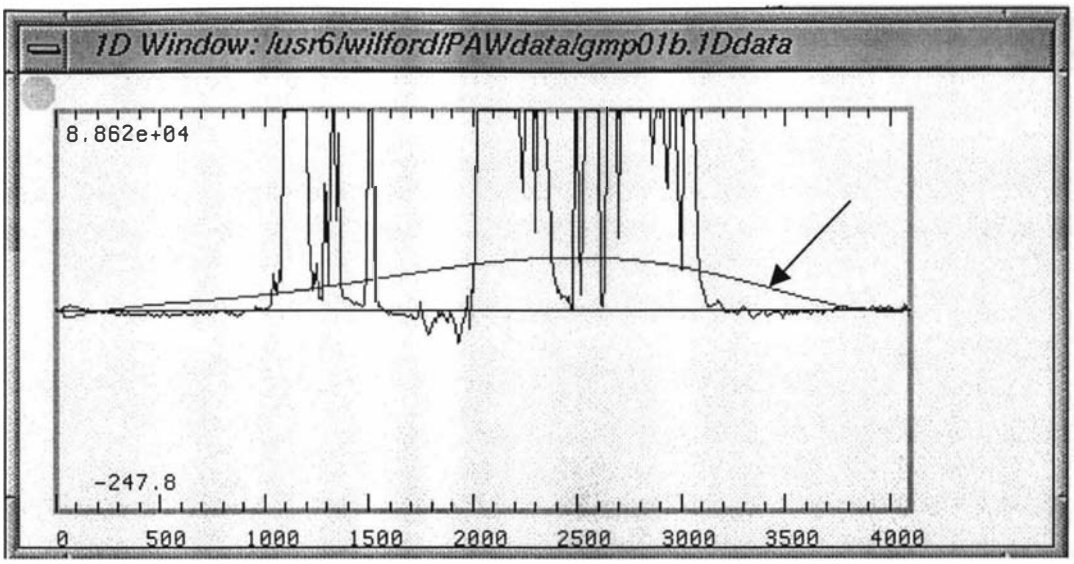


Figure 6.17: A 1D spectrum after baseline correction using polynomial of degree 5, where the arrowed curve was the baseline used in fitting.

For comparison, the following figure shows the result of fitting the baseline with a cubic spline. This is obtained by the same operations but with the Type of function set to s.

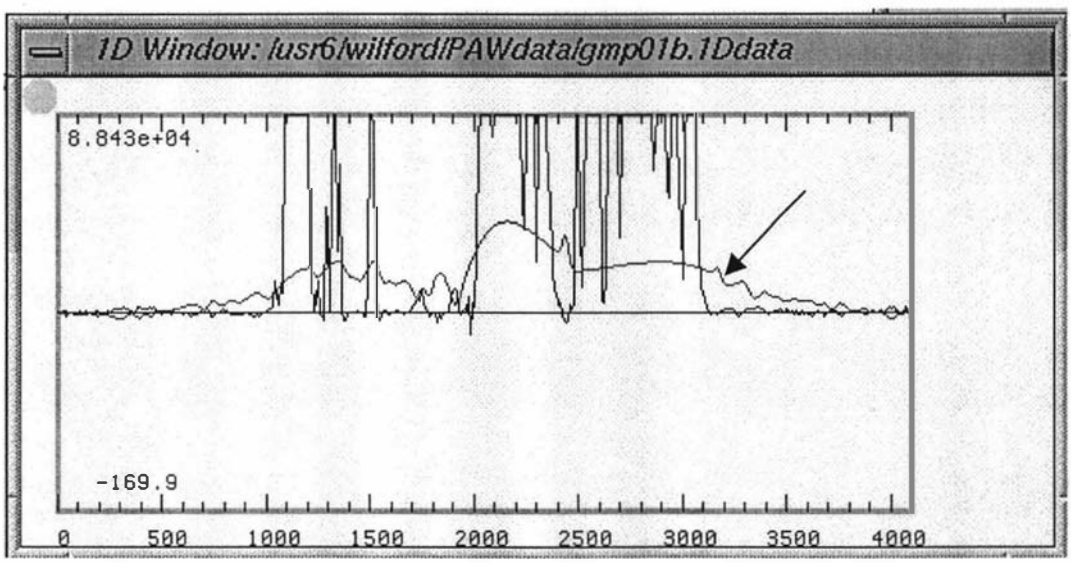


Figure 6.18: A 1D spectrum after baseline correction using cubic spline functions, where the arrowed curve is the baseline used in fitting.

➢ The macro statement

The macro statement for setting the baseline segments is

```
sbs (<number of segments>, <Segm 1>, <Segm 2>, ...)\
```

For example, to set four baseline segments as [44,140], [340,480], [900,936], [960,1020], the statement will be

```
sbs (4, 44,140, 340,480, 900,936, 960,1020)
```

where the first segment is to be used for the calculation of mean, SD and standard base-level.

The macro statement for baseline correction is

```
bc (<fitting function>, <degree of the function>)
```

For example, to perform the baseline correction with a 5th order polynomial fit, it will be

```
bc (p, 5)
```

In a standard 2D processing-parameter macro (see Chapter 4) for interactive processing using a dialog, two baseline-correction statements are assigned to the macro variables bcCom1 and bcCom2, e.g.,

```
bcCom1 = "bc (p, 5)"
```

6.12 Linear Prediction (LP)

➢ To perform linear prediction

- Type lp or choose [LPredict] in the *1D-processing Toolbox*.

The macro statement for linear prediction is

```
lp (%FirstPntUsedForLP,%NofPntsUsedForLP,
    %NofPolesForLP,%NofPntsToPredict,%PartToPred,SegmToPred)
```

Here, the meanings of the parameter names are mostly obvious. The PartToPred that specifies the data part to predict is either r (for real), i (for imaginary), or b (for both). The SegmToPred that specifies the segment to predict is either t (for predicting the tail part or the future) or h (for the head part or the past).

For example, to predict the future 288 point of a real data set using 412 points starting at point 100 with 20 poles, the lp command is

```
lp (100,412,20,288, r,t)
```

In a standard 2D processing-parameter macro (see Chapter 4) for interactive processing using a dialog, two linear-prediction statements are assigned to the macro variables lpCom1 and lpCom2, e.g.,

```
lpCom1 = "lp (100,412,20,288, r,t)"
```

6.13 Water-signal Suppression

The following operations produces a sample data set that is used to illustrate water-signal suppression:

- Load **data5dqfcosy.2Ddata**.
- Load Row 100 into 1DBuf#0 (Figure 6.18).

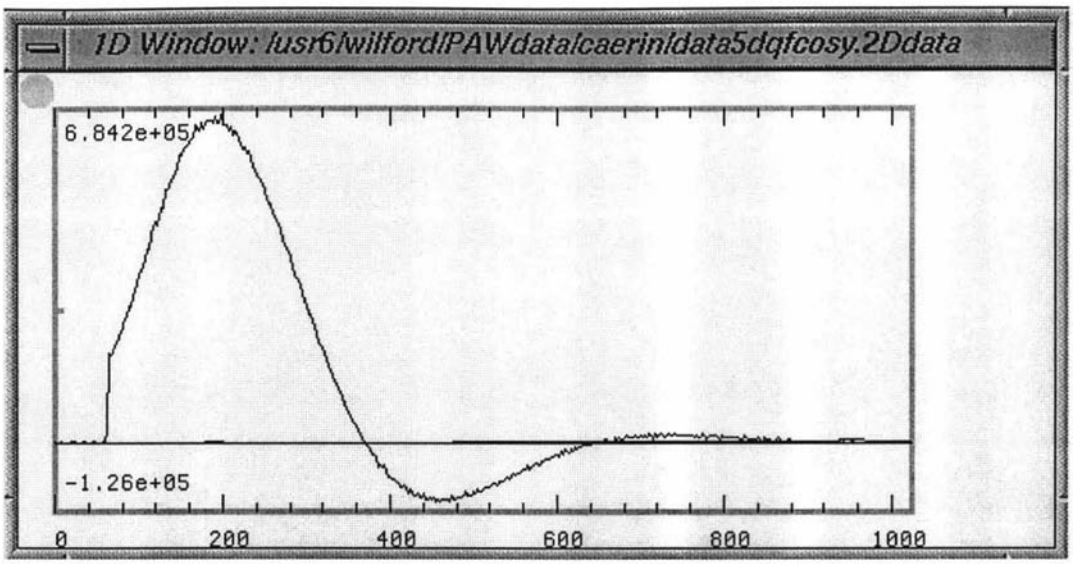


Figure 6.19: Row 100 of **data5dqfcosy.2Ddata**.

- Perform a left shift of 62 data points to remove the initial points caused by digitally filtering (Figure 6.18).

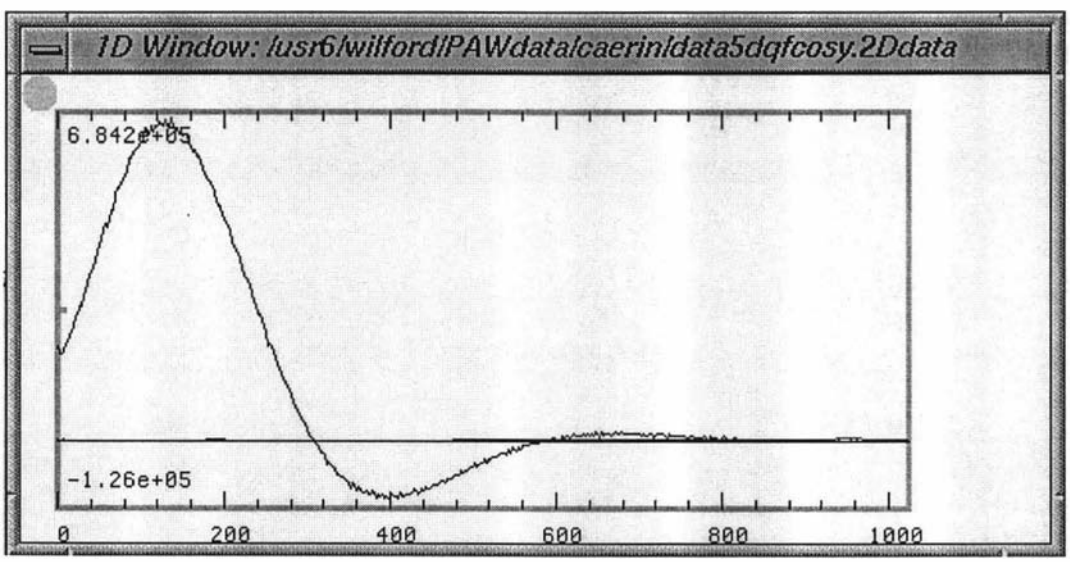


Figure 6.20: Row 100 of **data5dqfcosy.2Ddata** after a left-shift of 62 data points

➢ To suppress the water signal

- Choose [T-dWSuppr] in the *1D-processing Toolbox*.

Repeat this step if the water signal is still present. This results in a further improvement in the t-domain signal, as shown in the next diagram.

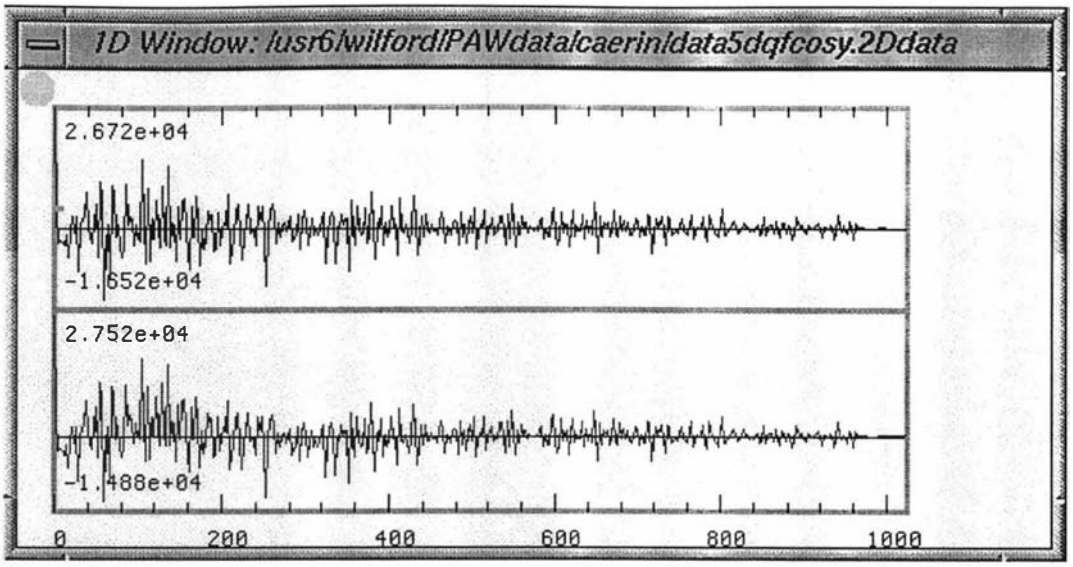


Figure 6.21: The signal from the last figure after water suppression has been applied. The data set on the top has been suppressed twice.

➢ Processing the suppressed signal

To obtain a better spectrum, the data can be filtered as follows:

- Choose [Filter] in the *1D-processing Toolbox* to open the *Filter Toolbox*.
- Choose [EnhSineBell] in the *Filter Toolbox*; set the parameters to be 960, 0, 1 and 0, respectively, from top to bottom; then, choose [Save]. (Note: the first parameter (i.e., 960) is to flatten the last 62 data points resulting from the circular shift; the second and third parameters determine a sine-bell filter that is often used when processing COSY spectra.)

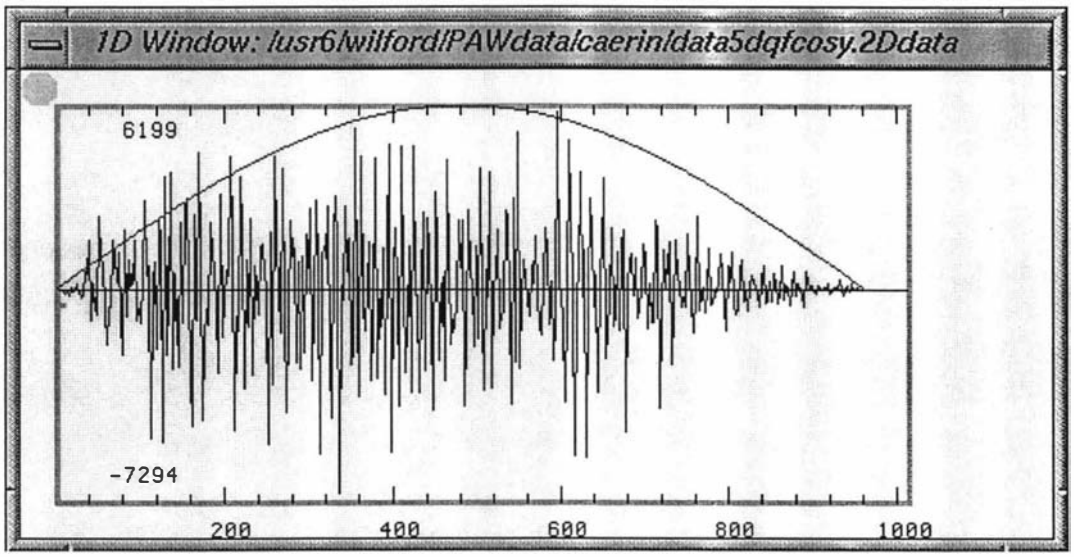


Figure 6.22: The signal from the last figure after filtering with a shifted sine function.

- Choose [FFT] in the *1D-processing Toolbox*.

In the following diagram are three spectra processed with and without water suppression.

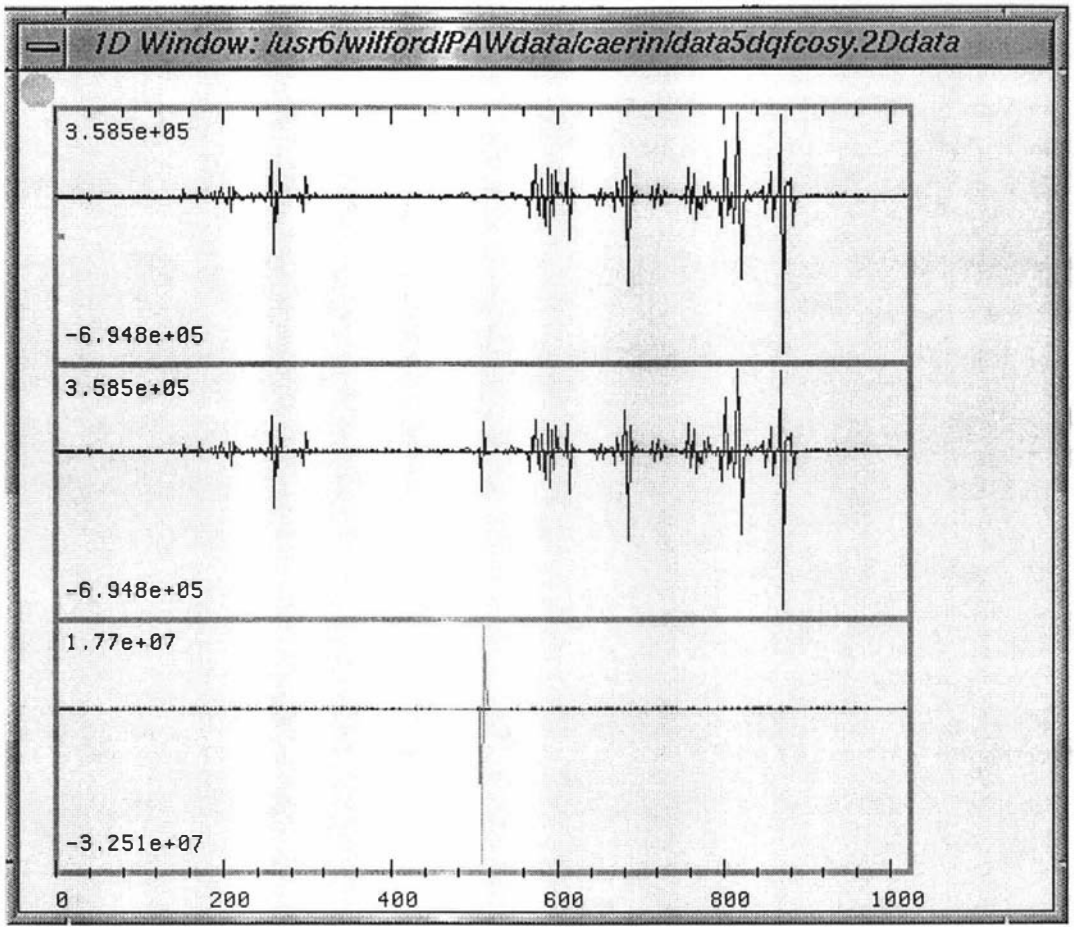


Figure 6.23: Three spectra processed with and without water-signal suppression:. The middle spectrum was obtained by suppressing the water signal once. and the top one, twice. The bottom spectrum was obtained without the suppression.

6.14 Fitting 1D Peaks

In the following example, four isolated peaks will be fitted by applying a peak-fitting operation.

- Load Row #573 of the **CaerinNoesy150C** spectrum (see Chapter 5 or 9).
- Choose [RealOnly] in the *1D-display Toolbox*.
- Zoom into the region from points 140 to 250, as shown in Figure 6.24.

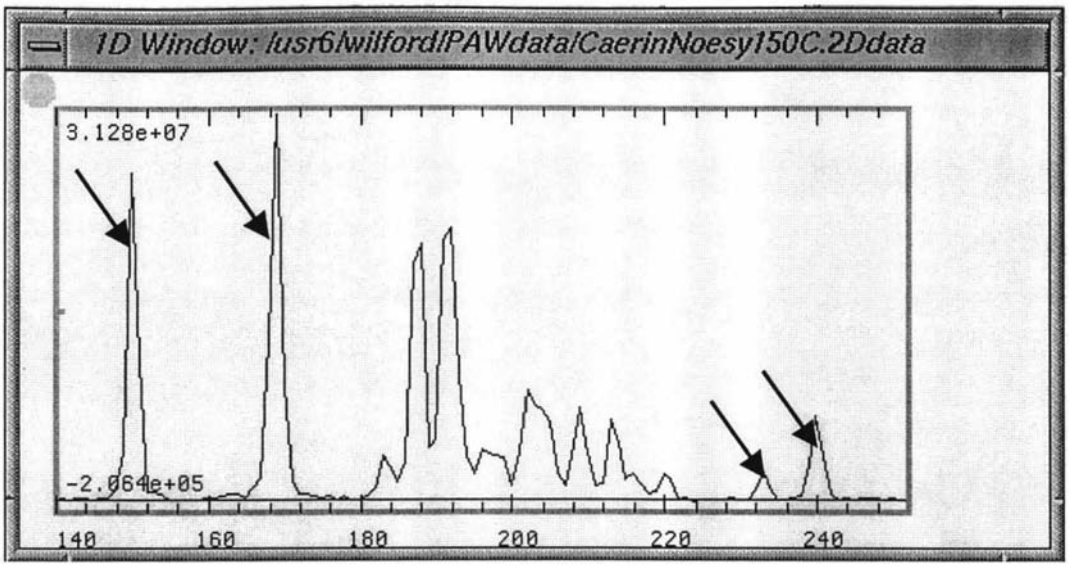


Figure 6.24: An expanded region of Row #573 of the **CaerinNoesy150C** spectrum, where the peaks arrowed are to be fitted.

- Type ftg or choose [GaussianFit] in the *1D-processing Toolbox* to open the *GaussianFit Dialog* as shown below, where NofPntsToFit for each peak is the number of points used in the fitting operation.

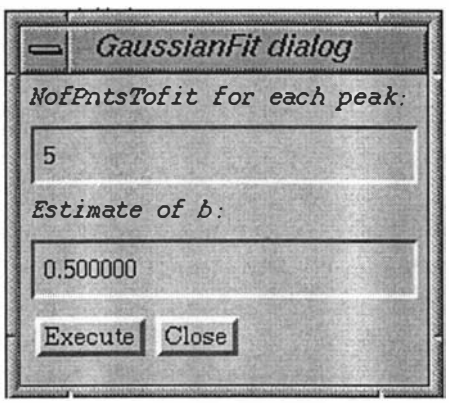


Figure 6.25: The *GaussianFit Dialog*

- With MsBtn#2, pick the four peaks arrowed in Figure 6.24.
- Choose [Execute] in the dialog to start the fitting. (Note: the original spectrum will be in buffer #8 after this step.)
- Display buffers #0 and #8, which contain the fitted curve (Figure 6.26 a) and the original curve (Figure 6.26 b).

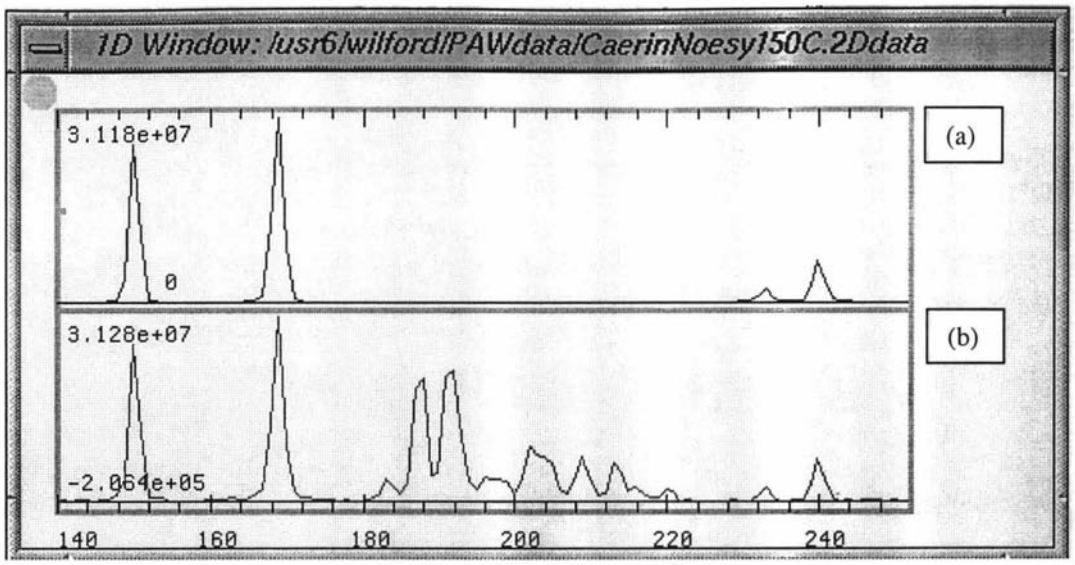


Figure 6.26: (a) The spectrum containing four fitted isolated peaks. (b) The original spectrum.

- Choose [DspBufs] and [Overlay] in the *1D-display Toolbox* to see how well the curve is fitted to the original spectrum

The result is shown in the next figure.

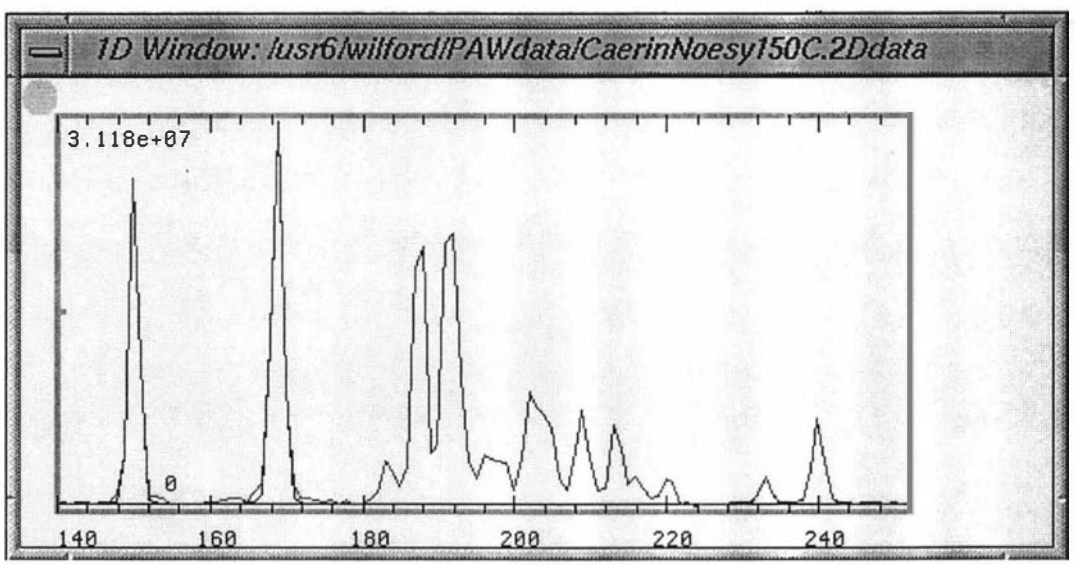


Figure 6.27: An overlay view of the two plots in the last figure.

Overlap peaks can be fitted in exactly the same way, but require better estimate of the parameter b, which can be obtained through isolated peak fitting. The parameter values of any fitted peak are always printed on the Unix shell, including the height, centre, b value, area and volume.

The next two figures show the outcome of an operation using 0.5 as an estimate for b.

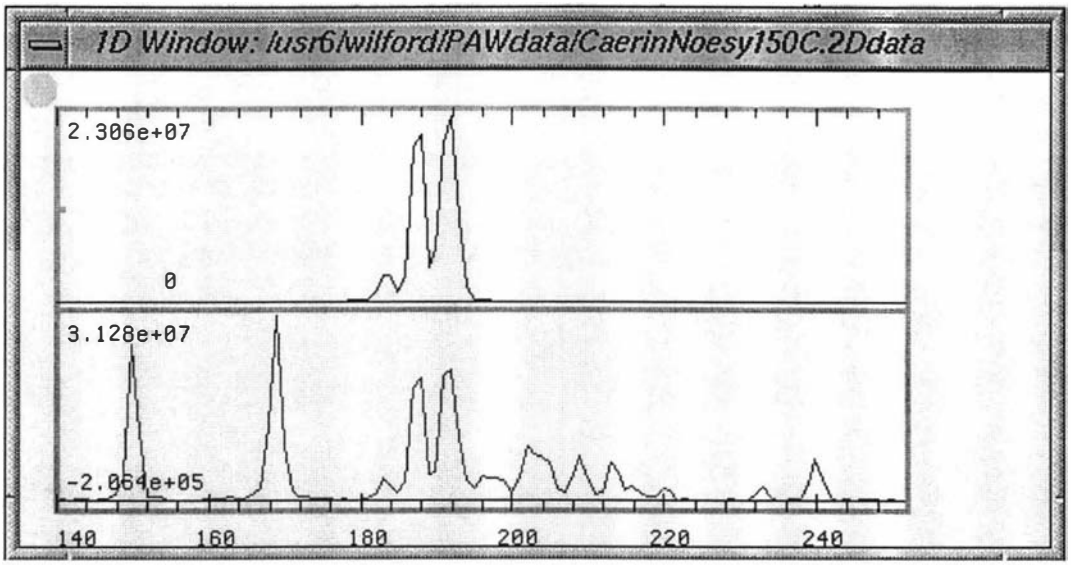


Figure 6.28: (a) The spectrum containing fitted overlap peaks. (b) The original spectrum.

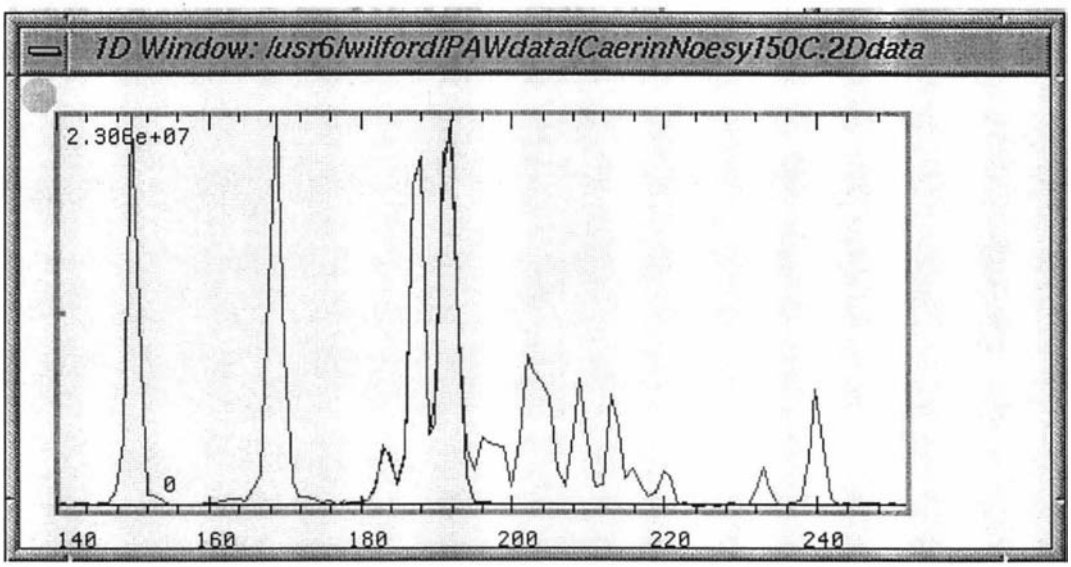


Figure 6.29: An overlay view of the two plots in the last figure.

Chapter 7:

Handling 2D NMR Data

7.1 Introduction

This chapter describes how to manage and display 2D data sets that are loaded in different buffers of 2D draw-windows.

To provide a platform for the operations in this chapter, it is recommended that a workbench be opened after starting PAW as follows:

- Start PAW.
- Choose [MyWorkbench] from the *Window Menu* to open the workbench that contains three 2D draw-windows, one 1D draw-window, a 2D over-view window, and a number of toolboxes, as shown in Figure 7.1 to 7.4.

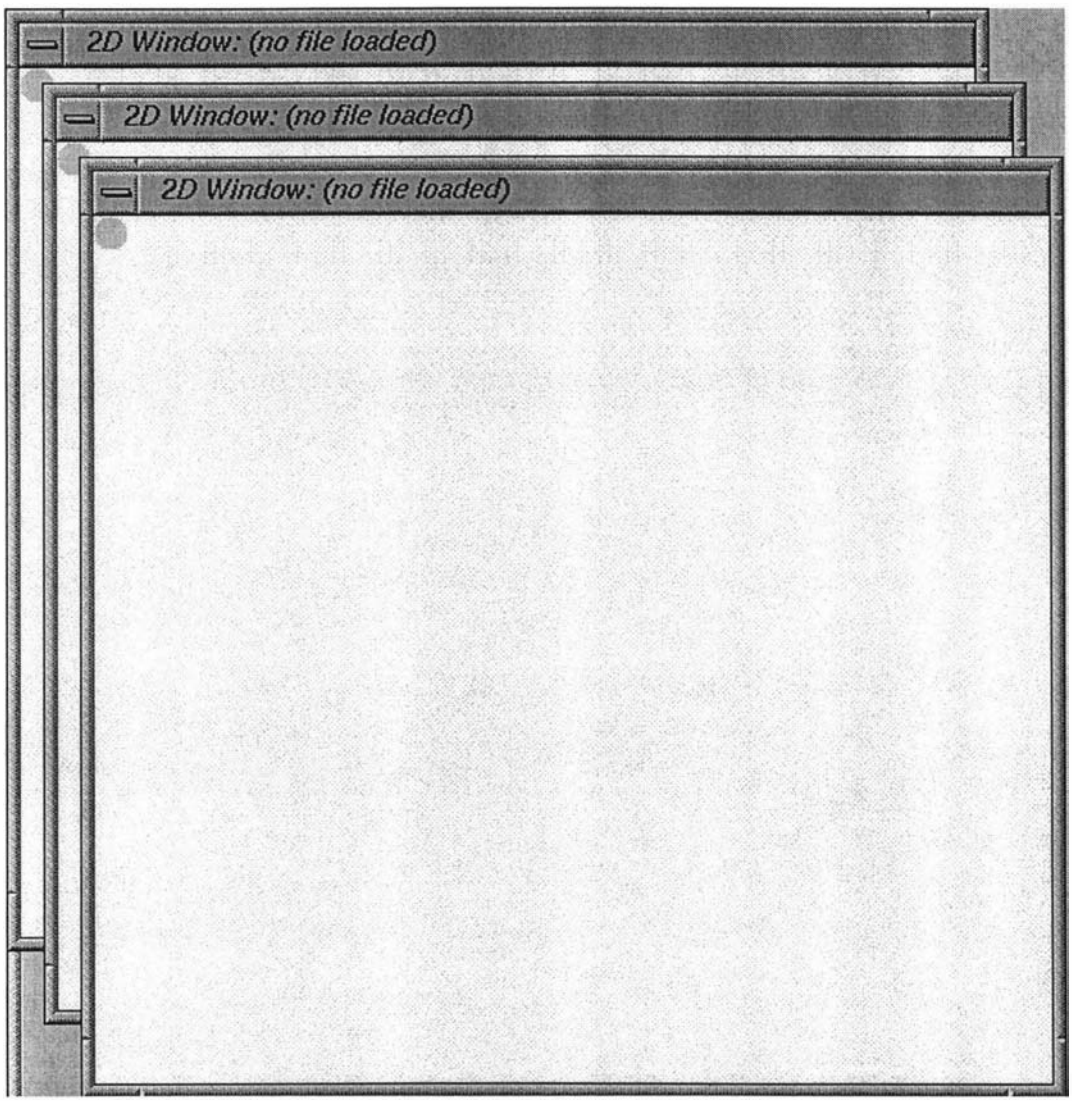


Figure 7.1: The three 2D draw-windows in MyWorkbench.

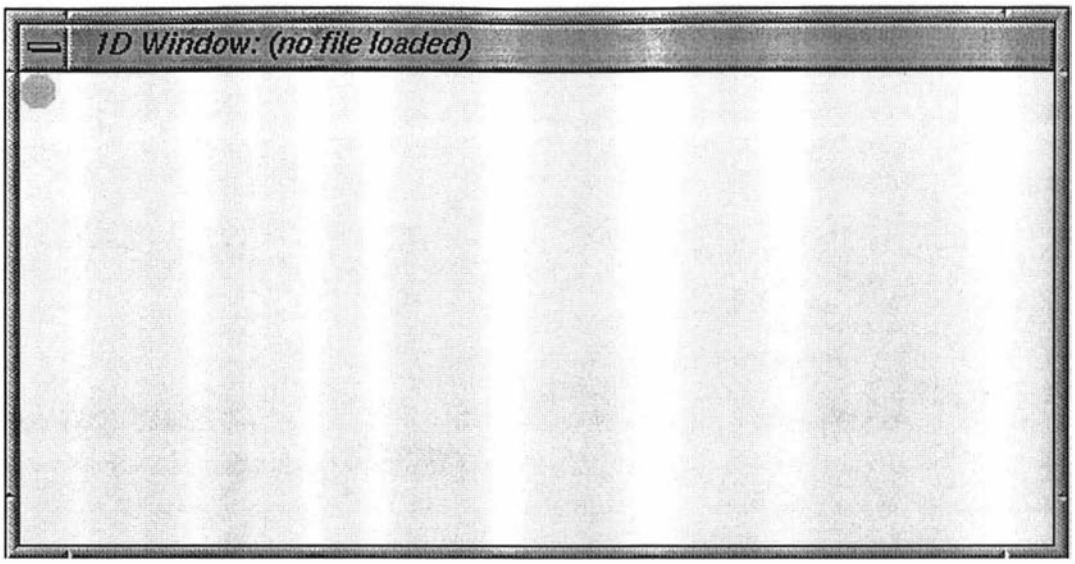


Figure 7.2: The 1D draw-window in MyWorkbench.

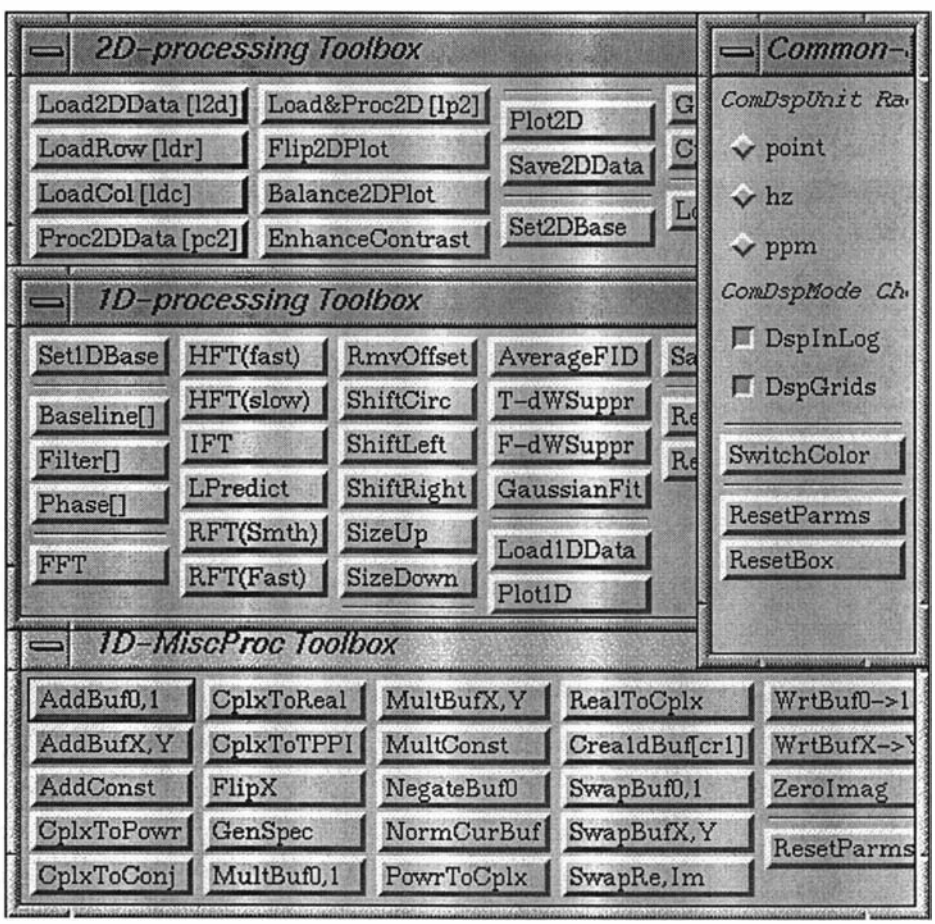


Figure 7.3: Four of the toolboxes in MyWorkbench.

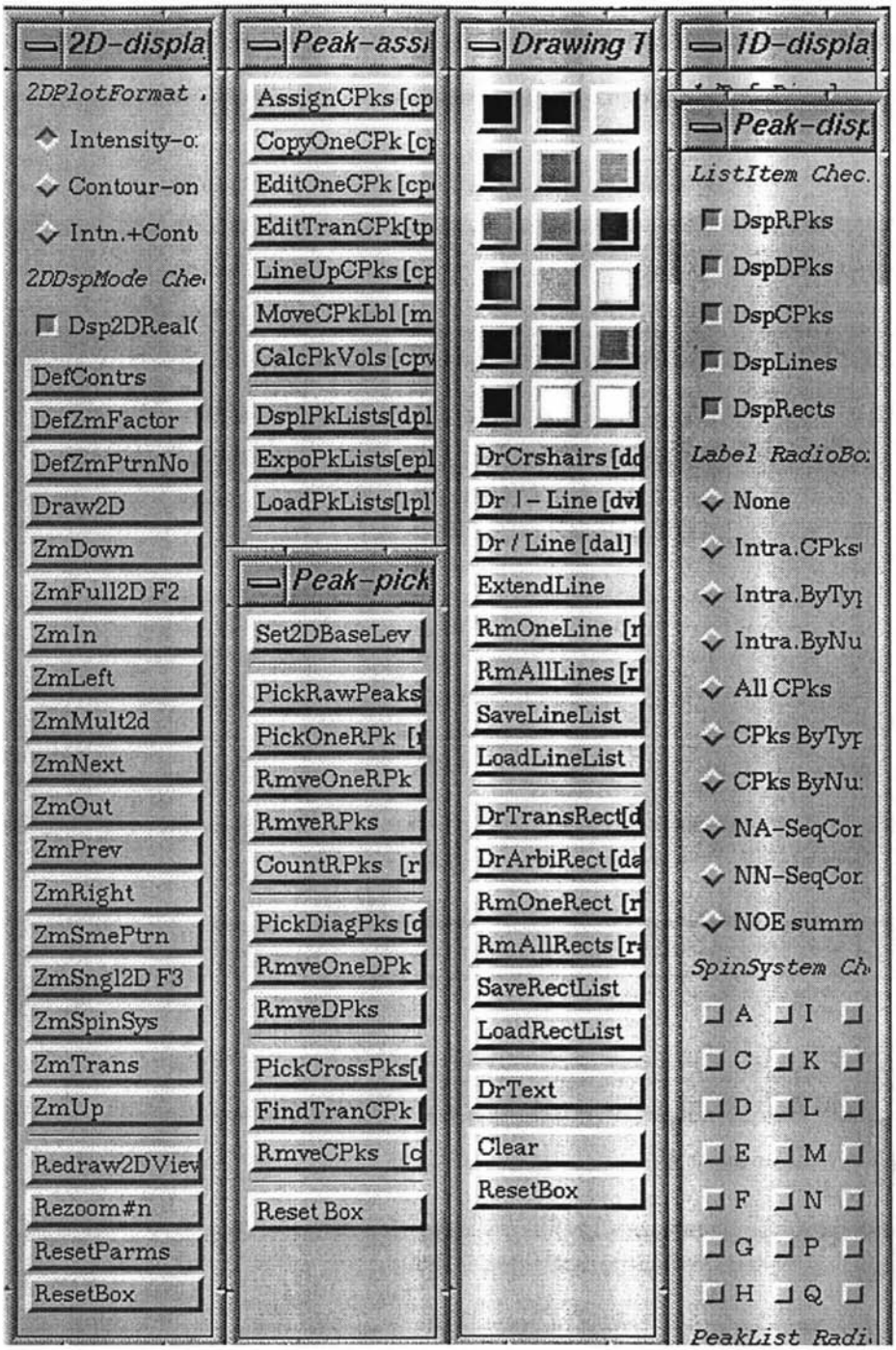


Figure 7.4: Other toolboxes in MyWorkbench.

7.2 Loading 2D NMR Data

Two-dimensional NMR data can be loaded either manually or by running a macro. The methods have been described in Chapter 4.

To provide examples in this chapter, three processed spectra should be loaded as follows:

- In the *Common-display Toolbox*, switch on the [ppm] and [DspInLog] buttons.
- Click in one of the 2D-windows that shows '*(no file loaded)*' on its title-bar.
- Choose [Load2Ddata] in the *2D-processing Toolbox* to open the *file-selection dialog* for loading NMR data (Figure 7.5).

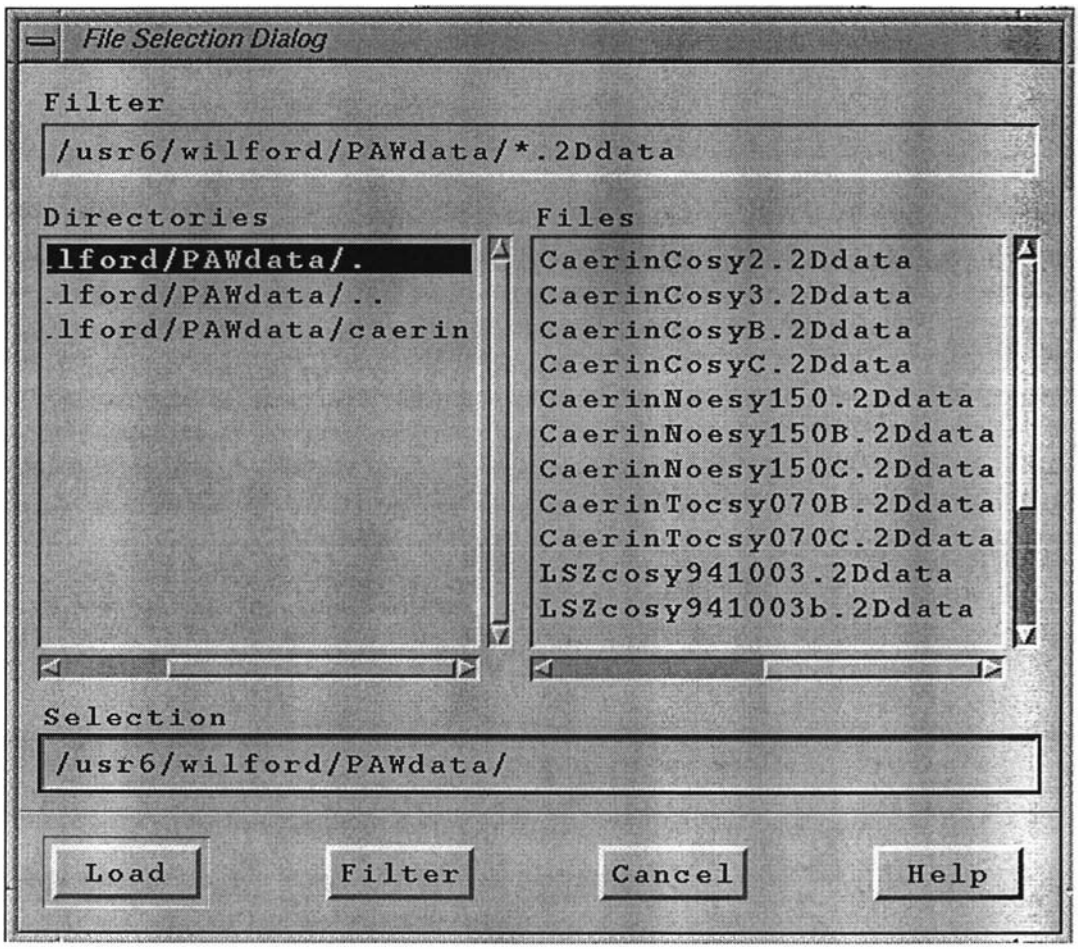


Figure 7.5: The file-selection dialog for loading NMR data file.

- Double-click on the **CaerinNoesy150C.2Ddata** in the file list. The processed Caerin 4.1 NOESY spectrum will be loaded and displayed.
- Repeat the above steps to also load the **CaerinTocsy150C.2Ddata** and **CaerinCosyC.2Ddata.**

These three spectra will be called the **CaerinNoesy150C**, **CaerinTocsy070C**, and **CaerinCosyC**, respectively.

Figure 7.6 shows the **CaerinNoesy150C** spectrum loaded without colour-map adjustment. The other two spectra are covered by this window.

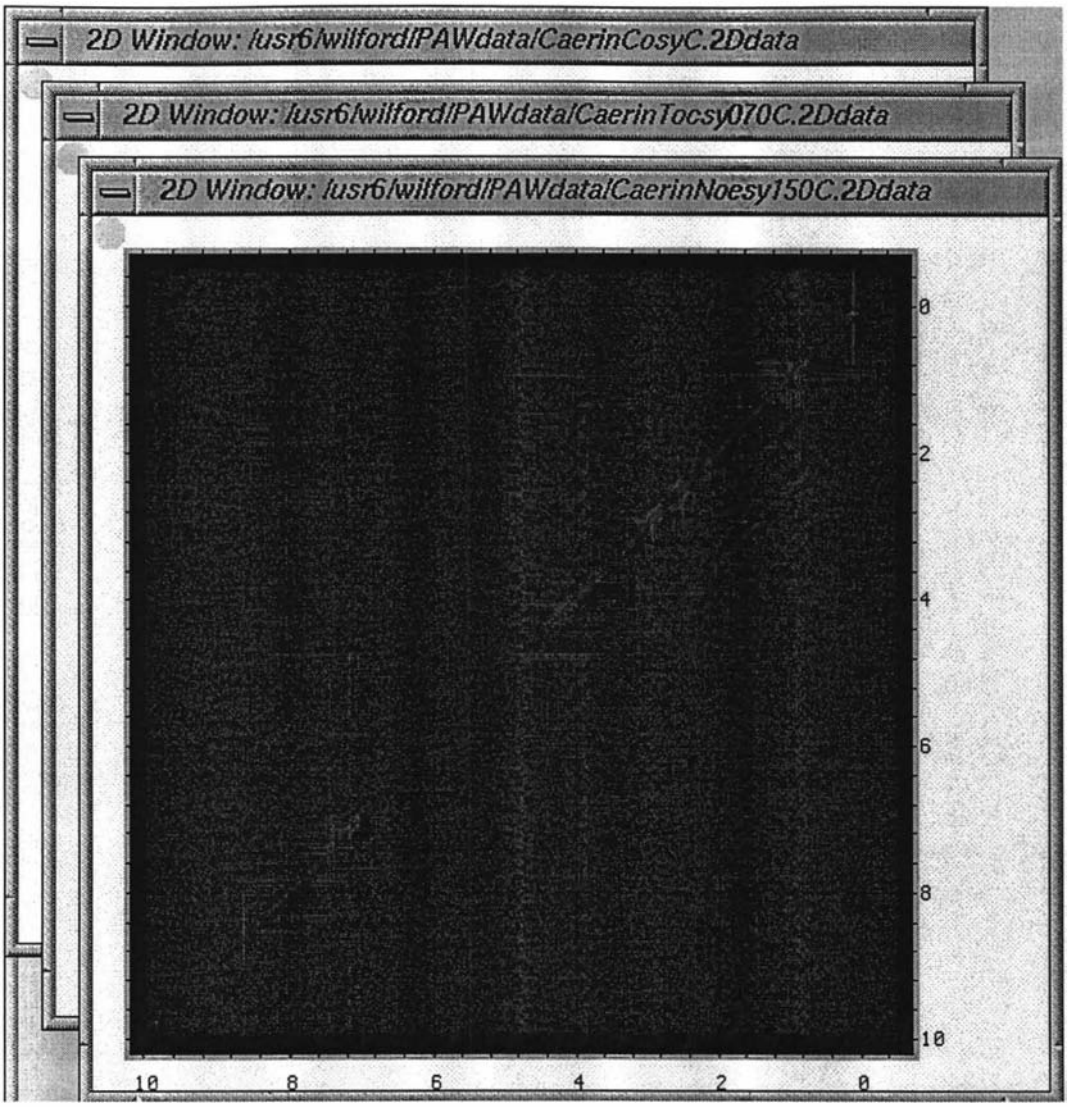


Figure 7.6: Three spectra loaded without colour-map adjustment, with the **CaerinNoesy150C** on top.

7.3 Displaying 2D NMR data

This section describes how to set various 2D-display options, including the use of buttons in the toolboxes for 2D display.

The toolboxes for 2D display are briefly introduced in Section 7.3.1. The operations to change the 2D-display colour scheme are then discussed, because they are required for the presentation of the following sections. The command buttons in the *2D-display Toolbox* are described extensively in Section 7.3.3 to 7.3.5, followed by those in the *Common-display Toolbox*. Four of the buttons in the *Common-display Toolbox* have been described in Chapter 5 for the 1D display, their functions are the same for the 2D display and will not be repeated.

7.3.1 The toolboxes for 2D display

There are two toolboxes that are related to 2D display: the *2D-display Toolbox* and the *Common-display Toolbox.*

The *2D-display toolbox* (see the one on the left in Figure 7.4) has a radio-box to set 2D-plot format, a checkbox to set 2D-display modes, and a list of command buttons for 2D display operations.

The Common-display Toolbox (see the one on the right in Figure 7.3) can be used to set the parameters that are common to both 2D and 1D displays.

➢ **To open the *2D-display Toolbox* (o2d)**

- Type o2d or choose [2DDspTBox] from the [Display] menu.

➢ **To close the *2D-display Toolbox* (x2d)**

- Type x2d or double-click on the window-control button on the top-left corner of the toolbox.

The methods used to open and close the *Common-display Toolbox* have been described in Chapter 2.

7.3.2 Displaying 2D data in different colour schemes

2D NMR data can be displayed in different modes as follows:

- Choose [SwitchColour] from the *Common-display Toolbox.* The *SetColorMap Dialog* (Figure 7.7) will be opened and the 2D-window will be re-displayed with a new colour scheme, as shown in Figure 7.8.

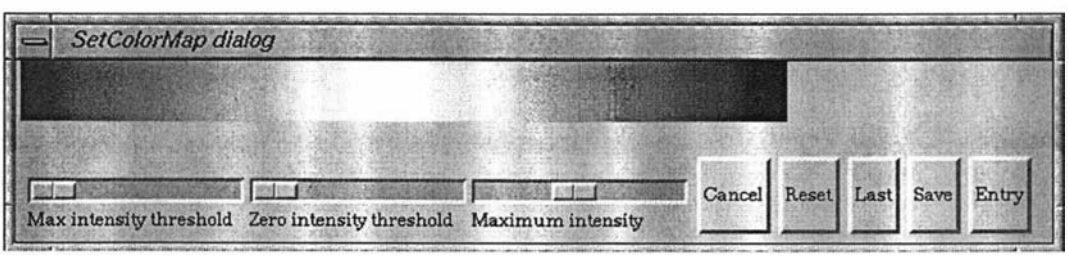


Figure 7.7: The *SetColorMap Dialog.*

- Choose [Save] from the dialog.

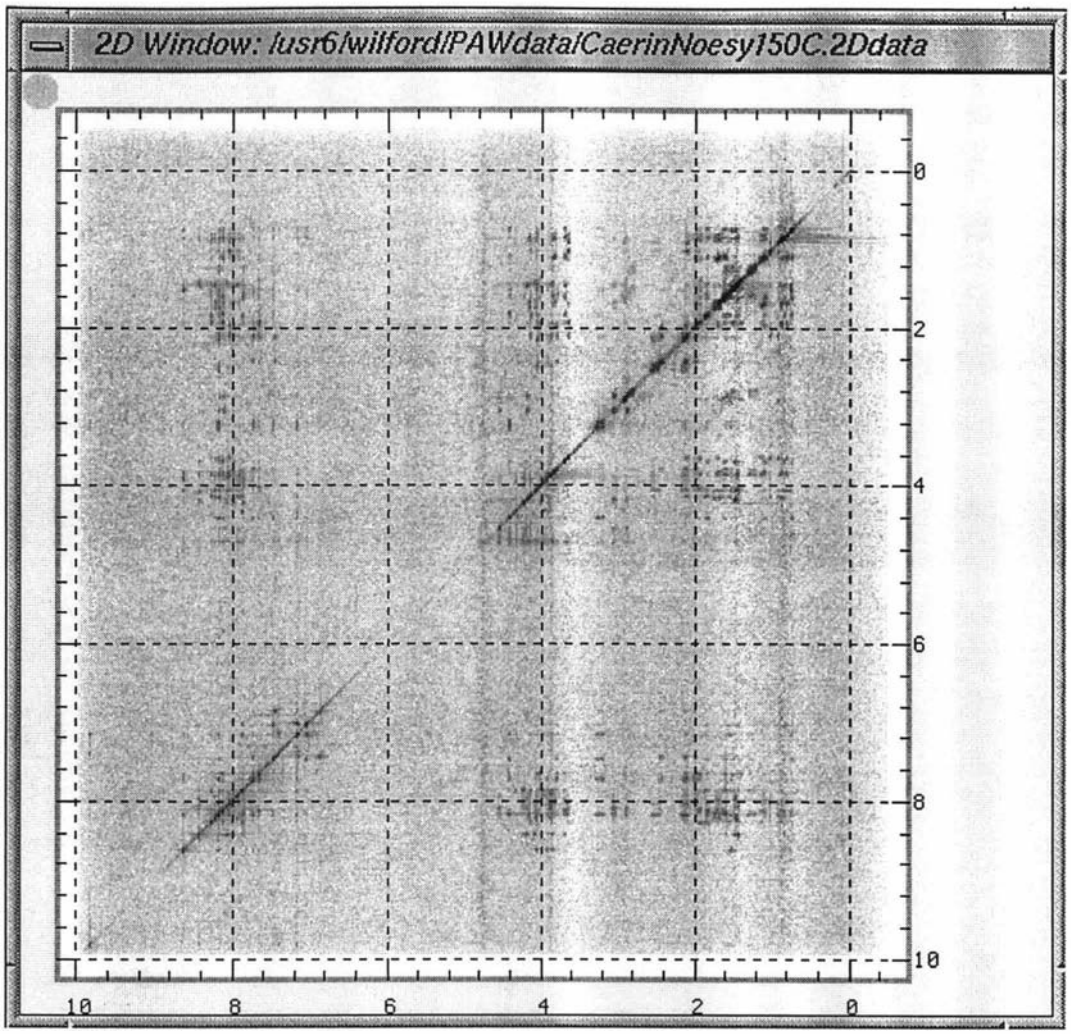


Figure 7.8: The **CaerinNoesy150C** displayed with colour scheme 2 and no colour–map adjustment. The bright background of this scheme provides better contrast and is perhaps preferable for publications.

- Choose [Save] in the *Colour-map dialog*.

Most of the 2D NMR data are presented with the colour scheme 2 in this thesis.

7.3.3 Viewing different regions of 2D data sets

There are many zooming options in the *2D-display Toolbox* to view regions of 2D data sets in different ways. This subsection describes the zooming operations, many of them are presented with examples.

➢ **To display a full view of a 2D spectrum (zf or zf2)**

- Click on the multi-region plot of the **CaerinNoesy150C**.
- Type zf or choose [ZmFull2D] in the *2D-display Toolbox*.

So far, all the figures in this chapter were presented in the full view.

➢ **To display a single region of a 2D spectrum (zs or zs2)**

- Click in the window for the **CaerinNoesy150C**.
- With the mouse-pointer in the 2D-window, type zs or choose [ZmSngl2D] in the *2D-display Toolbox*. The instruction 'Use Btn 1 to set a zoom-region.' will appear on top of the 2D-window.
- Use the left mouse-button to select the upper-left region from approximately [8.65, 0.54] to [6.75, 4.64]. (Note that all coordinates are measured in ppm from here on.)
- Display the plot in the Intensity+Contour mode (see Section 7.3.5).

The result is shown in Figure 7.9.

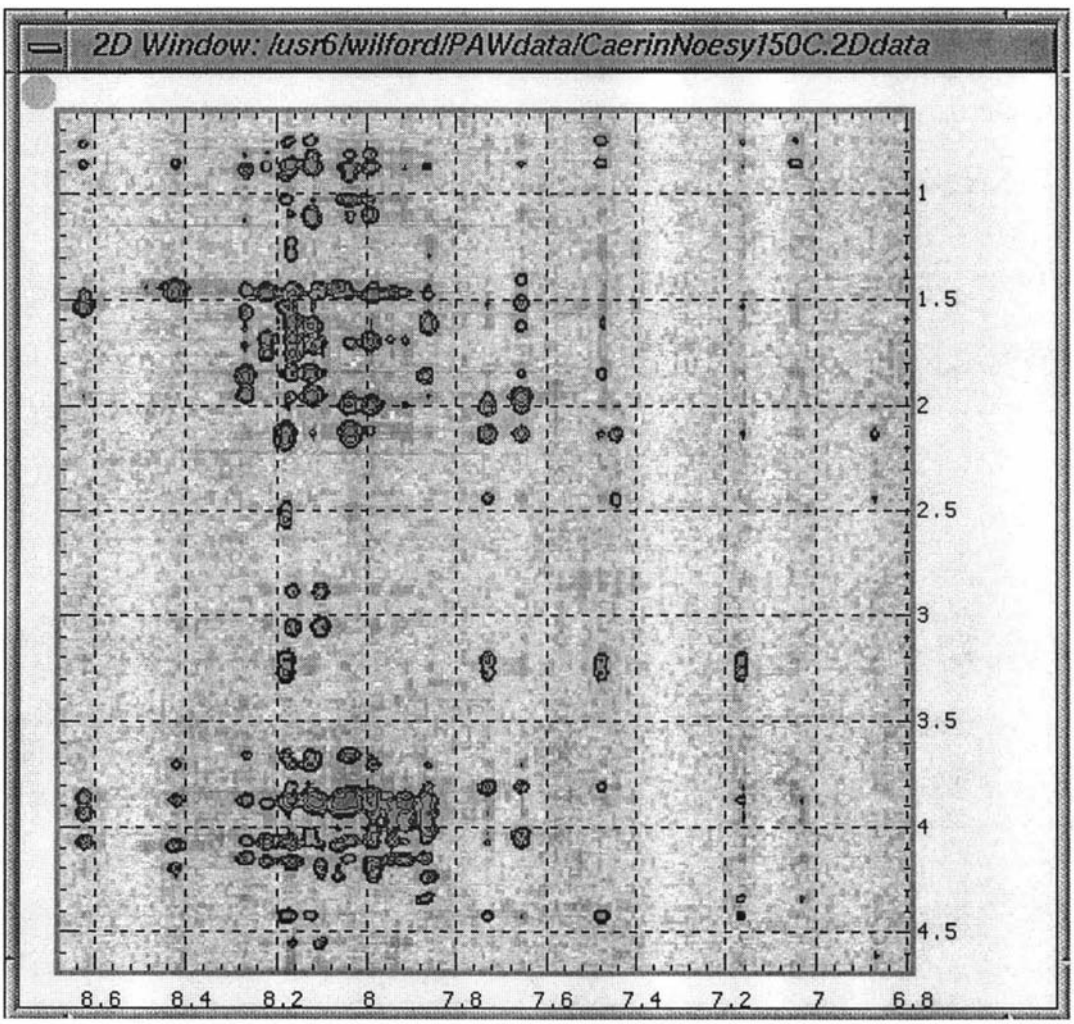


Figure 7.9: The upper-left region of the **CaerinNoesy150C**.

➢ **To display a transposed region of the current displayed region (zt)**

- Keep the cursor on the **CaerinNoesy150C** draw-window and type zt to zoom into the transposed fingerprint region. The contour plot in this region has a slightly lower base level, as shown in Figure 7.10. (The *First Level Threshold* of this contour plot can be re-defined to 23 if desired.)

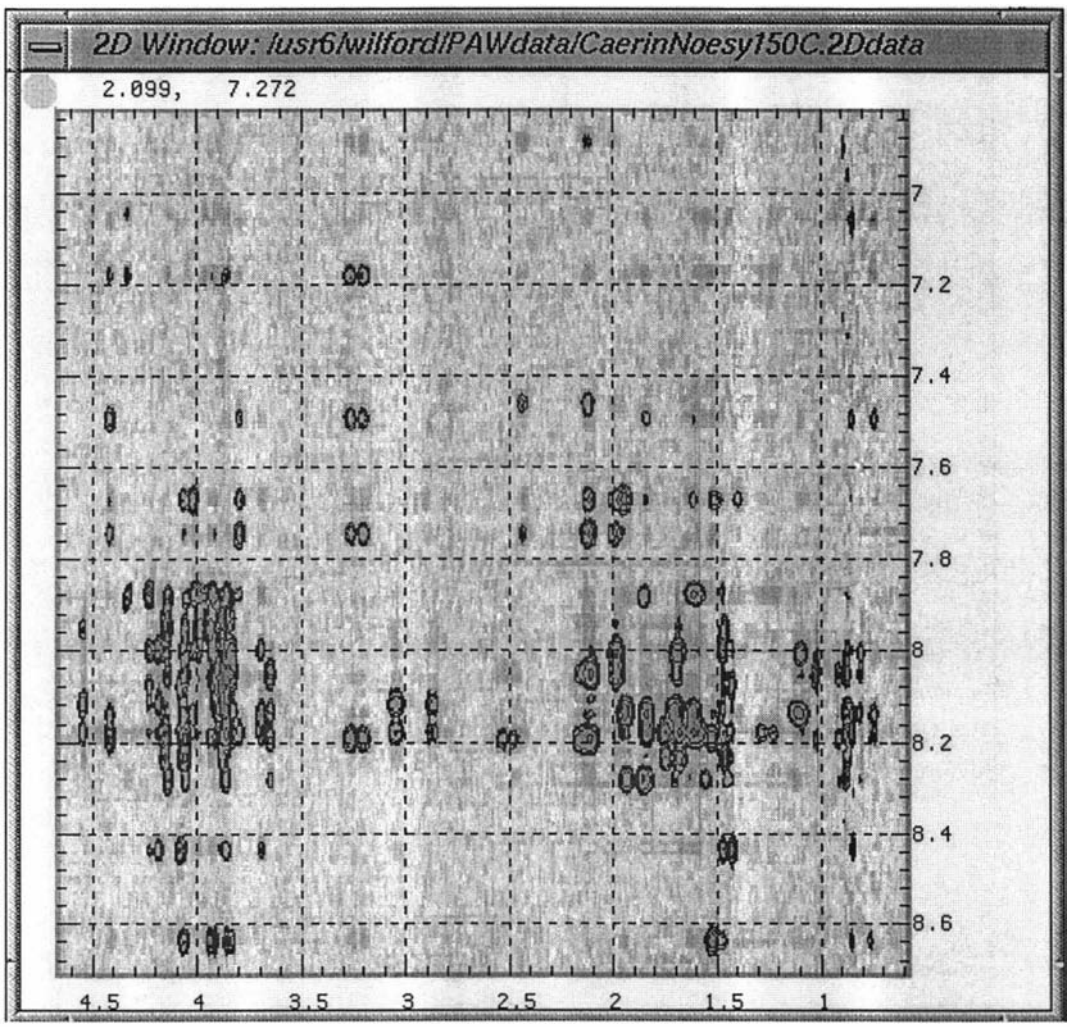


Figure 7.10: The transposed region of that shown in the last figure.

➢ **To display multiple regions of a 2D spectrum (zm or zm2)**

- Click in the window for the **CaerinNoesy150C** again.
- Type zf to zoom into a full-view plot.
- With the mouse-pointer in the 2D-window, type zm. The instruction 'Hold and drag Btn 1 to set the regions.' will be displayed on top of the 2D-window.
- Select three regions from the diagonal as shown in Figure 7.11.

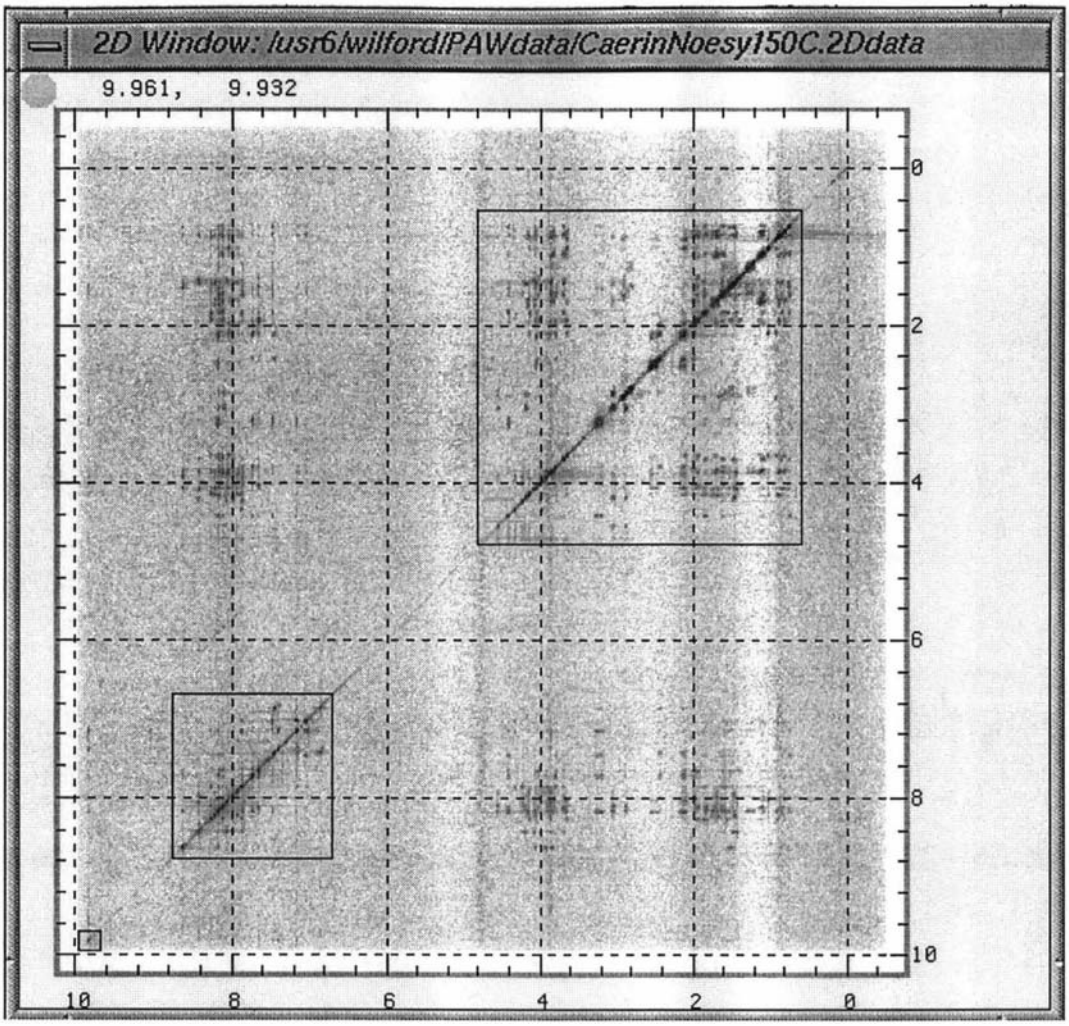


Figure 7.11: The rectangles on a 2D spectrum that set the regions for a 2D multi-region plot.

- Type dr to display the regions in a multi-region plot (Figure 7.12).

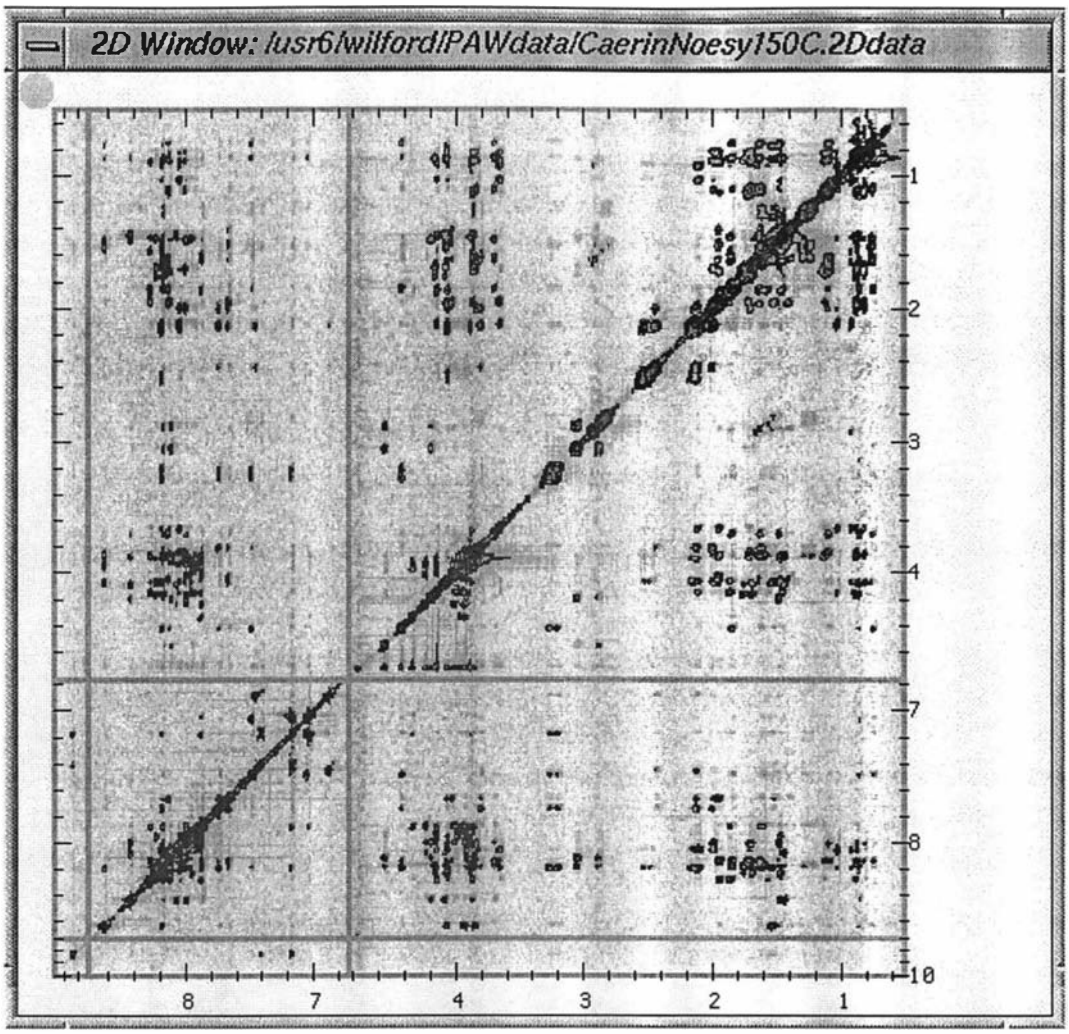
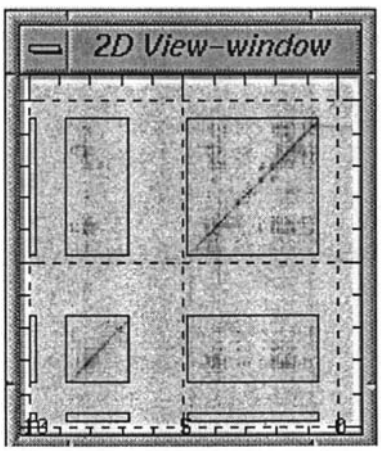


Figure 7.12: A 2D multi-region plot of the regions set by the rectangles in the last figure.



The 2D View-window (Figure 7.13) can be re-plotted with rectangles that shows the locations of the regions in the full spectrum.

Figure 7.13: The overview locations of the regions in the 2D View Window.

Note that the following rules are followed when setting the regions for a multi-region plot.

1. If a row of regions is enclosed, the top edge of the row in the multi-region view is set as the highest edge of the enclosed regions in the row; the bottom edge is the lowest edge of the regions in the row.
2. If a column of regions is enclosed, the left edge of the column in the multi-region view is the left-most edge of the enclosed regions in the column; the right edge is the right-most edge of the regions in the column.
3. If any enclosed regions in two adjacent rows (or columns) are overlapping, the two rows (or columns) are combined into one.
4. PAW does not require that all regions be enclosed as long as the regions that have been enclosed are enough to define a multi-region pattern. This was how the three rectangles in figure 7.11 defined a multi-region view in Figure 7.12.

The following nine simple operations are presented without example.

➢ **To display a smaller region of the current displayed region (zi)**
- Type zi or choose [ZmIn] in the *2D-display Toolbox*.

➢ **To display a larger region of the current displayed region (zo)**
- Type zo or choose [ZmOut] in the *2D-display Toolbox*.

Note that, in the last two options, the zooming factor for both contracting and expanding is set to 30% in the distribution version. This can be reset to any value between 0 and 100.

➢ **To redefine the zooming factor for the zi and zo commands (dzf)**

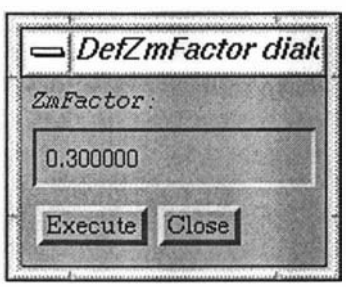


Figure 7.14: The DefZmFactor dialog

- Type dzf or choose [DefZmFactor] in the *2D-display Toolbox* to open the *DefZmFactor Dialog*, as shown on the left.
- Enter any value between 0.0 and 1.0.

➢ **To display the lower region of the current displayed region (zd)**
- Type zd or choose [ZmDown] in the *2D-display Toolbox*.

➢ **To display the upper region of the current displayed region (zu)**
- Type zu or choose [ZmUp] in the *2D-display Toolbox*.

➢ **To display the next defined zooming-pattern (zn)**
- Type zn or choose [ZmNext] in the *2D-display Toolbox*.

➢ **To display the previously defined zooming-pattern (zp)**

- Type zp or choose [ZmPrev] in the *2D-display Toolbox.*

➢ **To display the left region of the current displayed region (zl)**

- Type zl or choose [ZmLeft] in the toolbox.

➢ **To display the right region of the current displayed region (zr)**

- Type zr or choose [ZmRight] in the *2D-display Toolbox.*

➢ **To display a multi-region plot that shows all peaks of a spin system (zss)**

- Type zss or choose [ZmSpinSys] in the *2D-display Toolbox.*

An example of this will be given in Chapter 11 for the spectral assignment of Caerin 4.1.

➢ **To assign a number to the current display-pattern (dz#)**

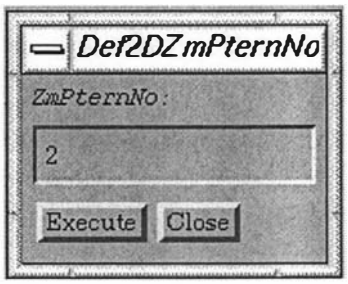


Figure 7.15: The DefZmPtmNo dialog

- Type dz1, dz2, ..., or dz8.

Alternatively,

- Type dzp or choose [DefZmPtmNo] in the *2D-display Toolbox* to open a dialog, as shown on the left.
- Enter any value between 1 and 8.

Note that the defined patterns are named Zm#1, Zm#2, and so on. The definitions allow any of the defined patterns to be easily recalled by a simple command such as z1, z2, etc.

For example, to define the multi-region pattern defined in Figure 7.12 as Zm#8, the following two steps are required:

- Keep the cursor in the **CaerinNoesy150C** draw-window and type dz8. The message '*Click on a 2D window to set Zm#8.*' will appear on the Unix shell that runs PAW.
- Click in the **CaerinNoesy150C** draw-window to define the current zoomed pattern as Zm#8. The message '*Zm#8 set.*' will appear on the Unix shell.

➢ **To display other spectra with the same zooming-pattern (zsp)**

- Click in the window for the **CaerinNoesy150C** again.
- Type zsp. The message '*Click on another widow with Btn 1.*' will appear above the plot.

- Click in the window for the **CaerinTocsy070C**. The spectrum will be re-plotted with the multi-region pattern as that for the **CaerinNoesy150C**. (Figure 7.16).
- Define this multi-region pattern as Zm#8. (Optional)

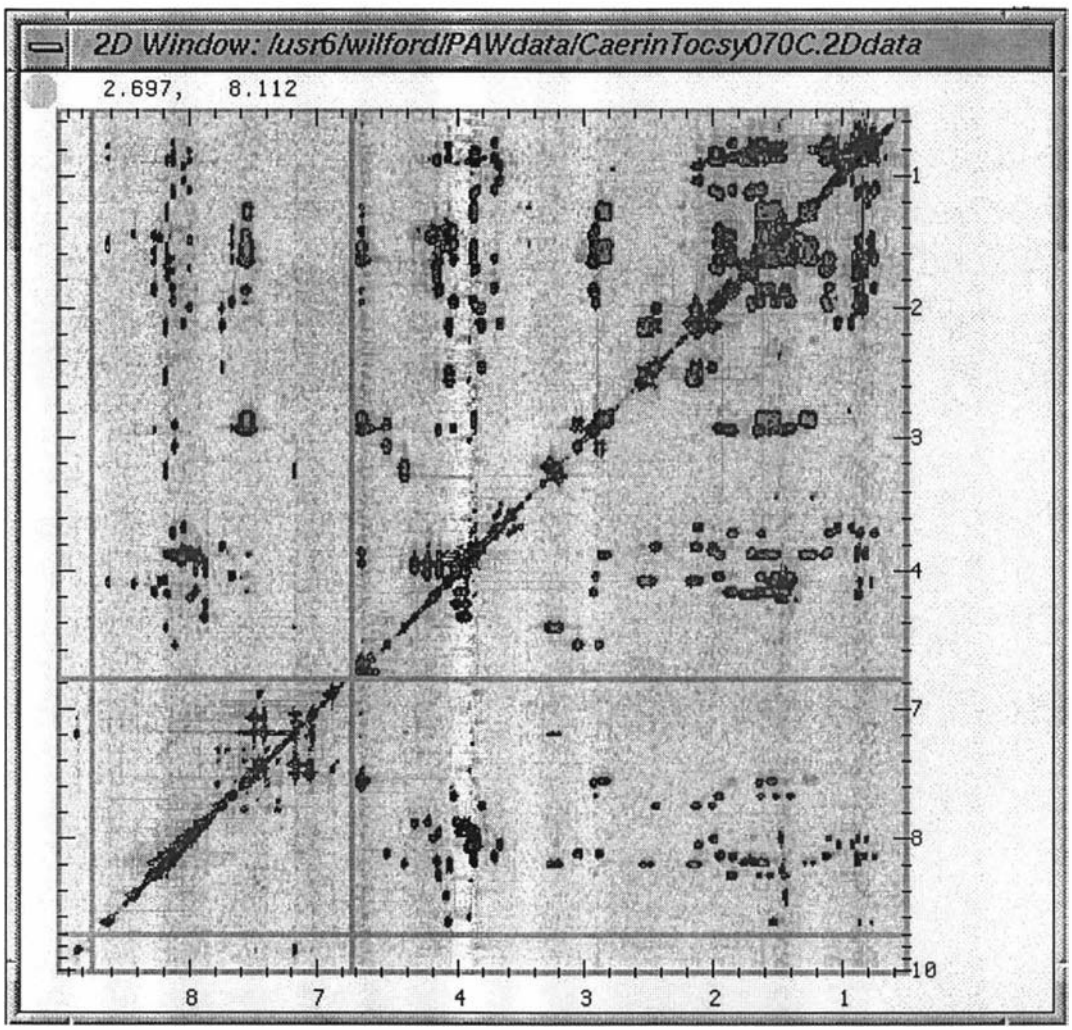


Figure 7.16: The **CaerinTocsy070C** that is zoomed into the same multi-region view as the **CaerinNoesy150C** in Figure 7.12.

- Repeat the above operations for the **CaerinCosyC** to obtain the same multi-region plot for it, as shown in Figure 7.17.

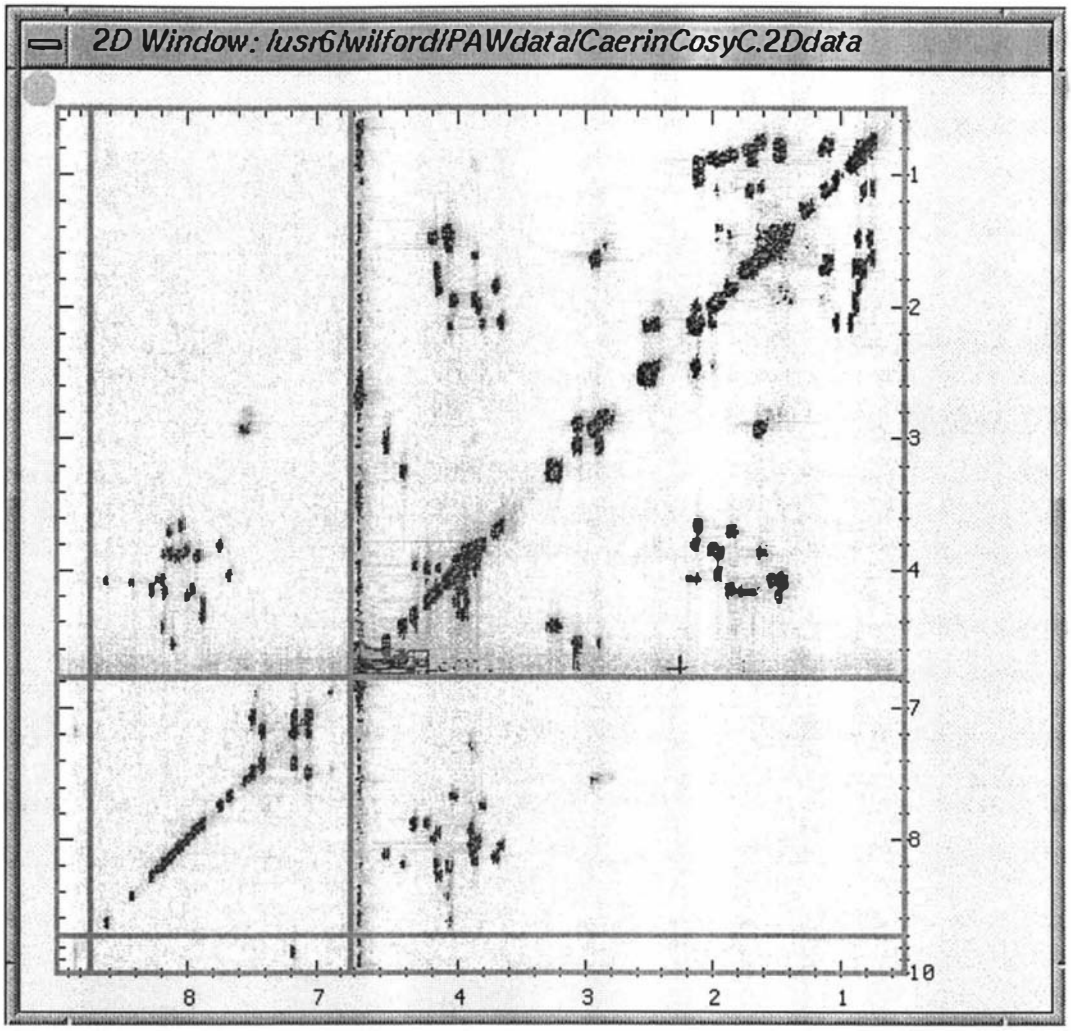


Figure 7.17: The CaerinCosy070C that is zoomed into the same multi-region view as the CaerinNoesy150C in Figure 7.12.

➢ To load a numbered zooming-pattern to the current 2D draw-window (z#)

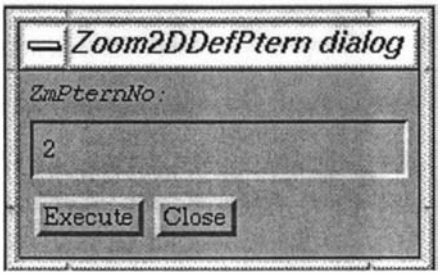


Figure 7.18: The *ZmPtrn Dialog*

- Type either z1, z2, ..., or z8.

or

- Type rzp or choose [ZmPtrn#n] in the *2D-display Toolbox* to open a dialog, as shown on the left.
- Enter any value between 1 and 8.

➢ To draw or re-draw the active 2D draw-window (dr2)

- Type dr2 or choose [Draw2D] in the toolbox.

Note that rectangles in the 2D view-window are always redrawn after every zooming. To speed up the drawing processes, the background spectrum of the view-window is

not redrawn automatically for each successive zooming. Minor confusion may arise if more than one spectrum is loaded.

➢ **To draw or re-draw the current 2D view-window (drv)**

- Type drv or choose [Draw2DView] in the toolbox.

7.3.4 Viewing different parts of 2D data sets

Displaying the imaginary part of 2D NMR data is mainly for checking the quality of 2D time-domain data set. 2D frequency-domain NMR data sets usually contain only real data, and hence the operation is not required.

Unlike the 1D display, PAW's 2D draw-window cannot simultaneously display the real and imaginary parts of the data. Therefore, if a complex 2D data set is loaded, PAW displays only either the real part or the imaginary part of the data.

➢ **To display only real or imaginary data**

- Click in the 2D draw-window that currently displays the data.
- Either type dr2 or choose [Dsp2DRealOnly] in the *2D-display Toolbox*.

When the button is pushed down, the display is real-only; otherwise, it is imaginary-only.

Note that you can also explicitly convert a complex 2D data with the [2DCmplxToReal] or [2DCmplxToImag] command. Both commands perform an irreversible process. The only way to regain the complex data is to reload it.

7.3.5 Displaying 2D data in different modes

A 2D NMR data set can be displayed in a combination of different modes:

- In terms of data intensities and contours, the data set can be displayed either in the intensity-only mode, contour-only mode, or both.
- In terms of the units used, it can be displayed either in points, Hertz, or ppm.
- In terms of the calculation of colours for different data intensities, it can be displayed either in linear scale or the natural logarithmic scale.

In addition, the display can either be with or without grids.

So far the expanded views of all figures in this chapter have been presented in the Intensity+Contour mode. The following operations change the 2D display to a different mode:

➢ **To set the Intensity-only mode**

- Click in a 2D draw-window.

- Choose [IntensityOnly] in the *2D-display Toolbox*.

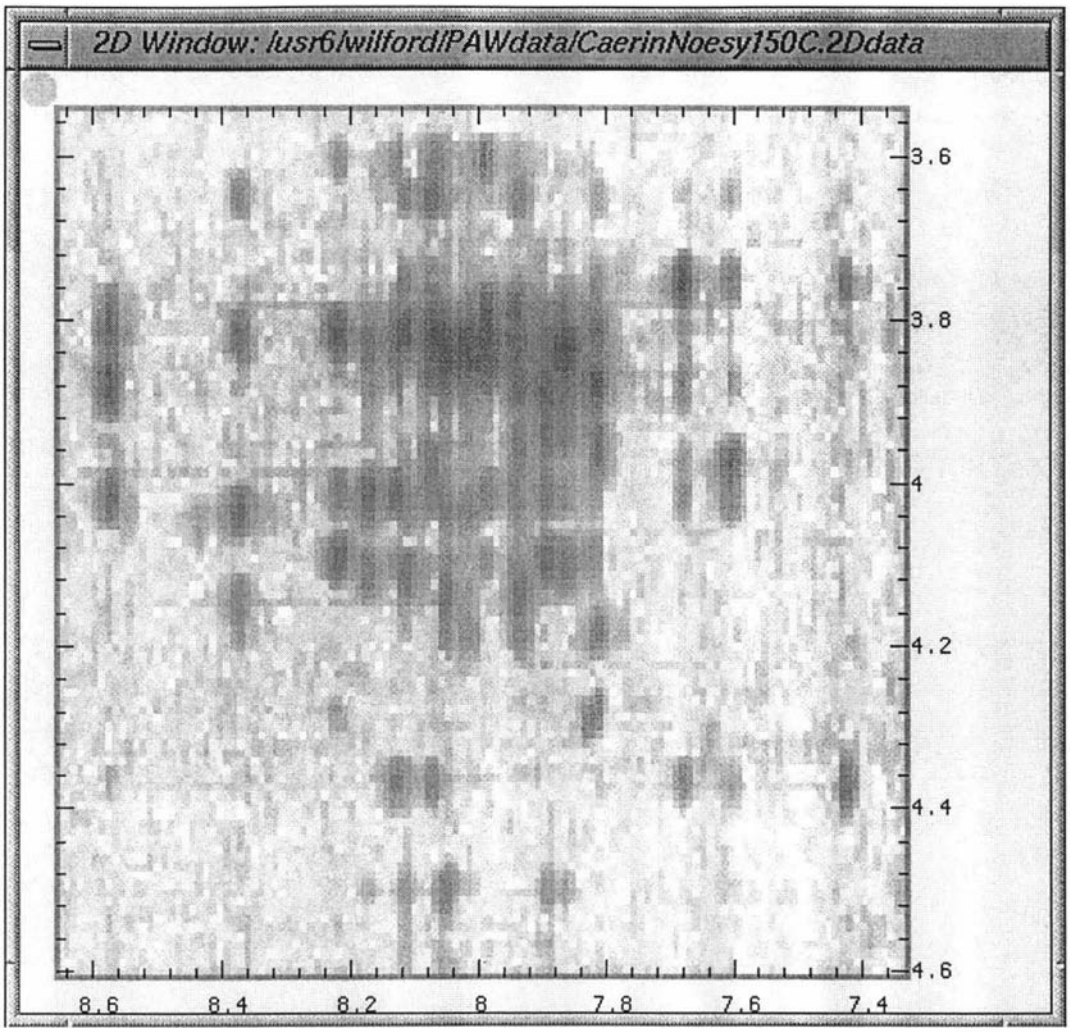


Figure 7.19: An expanded view of a 2D region in the intensity-only mode.

➢ To set the contour-only mode

- Click in a 2D draw-window.
- Choose [ContourOnly] in the *2D-display Toolbox*.

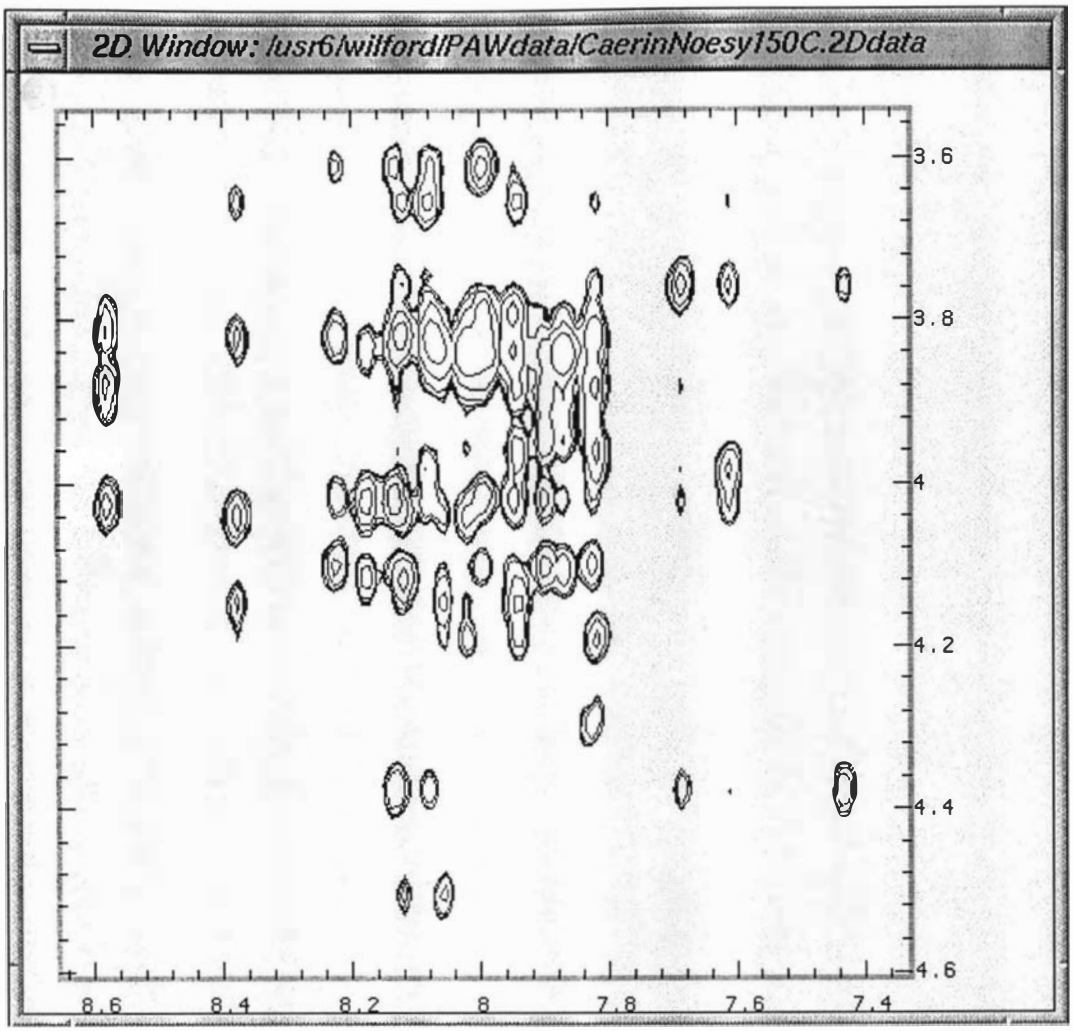


Figure 7.20: An expanded view of a 2D region in the contour-only mode.

➢ To set the intensity and contour mode

- Choose [Intn.+Contour] in the *2D-display Toolbox*.

Note that PAW does not produce full-view contour plots. To produce a full-view contour plot, it is required to zoom into a slightly smaller region that covers almost the entire spectrum.

➢ To set the contour format

This section describes the operations needed to set the contour format and define proper contouring parameters for displaying the Caerin spectra.

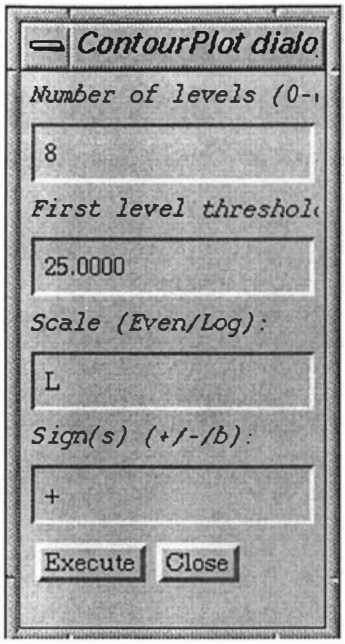


Figure 7.21: The *ContourPlot Dialog*.

- Choose [Intn.+Contour] in the *2D-display Toolbox*.
- Click in the draw-window that displays the **CaerinNoesy150C** data.
- Choose [DefContrs] in the *2D-display Toolbox* to open a *Contour-plot Dialog* (Figure 7.23).
- In the dialog, change the *First-level Threshold* to 28. Then, choose [Execute].

Note that these operations only affect the contour display parameters for this particular draw-window.

7.3.6 Displaying 2D data in different scales

This subsection describes the scale-changing operations. The examples given in this subsection require that the upper-right region of **CaerinCosyC** be loaded as follows:

- Click in the draw-window for the **CaerinCosyC** spectrum.
- Type z8 to recall the defined multi-region pattern, if necessary.
- Type zs to zoom into a region from approximately [0.50,0.49] to [4.64,4.64]. A bit more noise is shown in this region. If desired, redefine the *First Level Threshold* to be 28, which will make the contour plot much tidier, as shown in Figure 7.22.

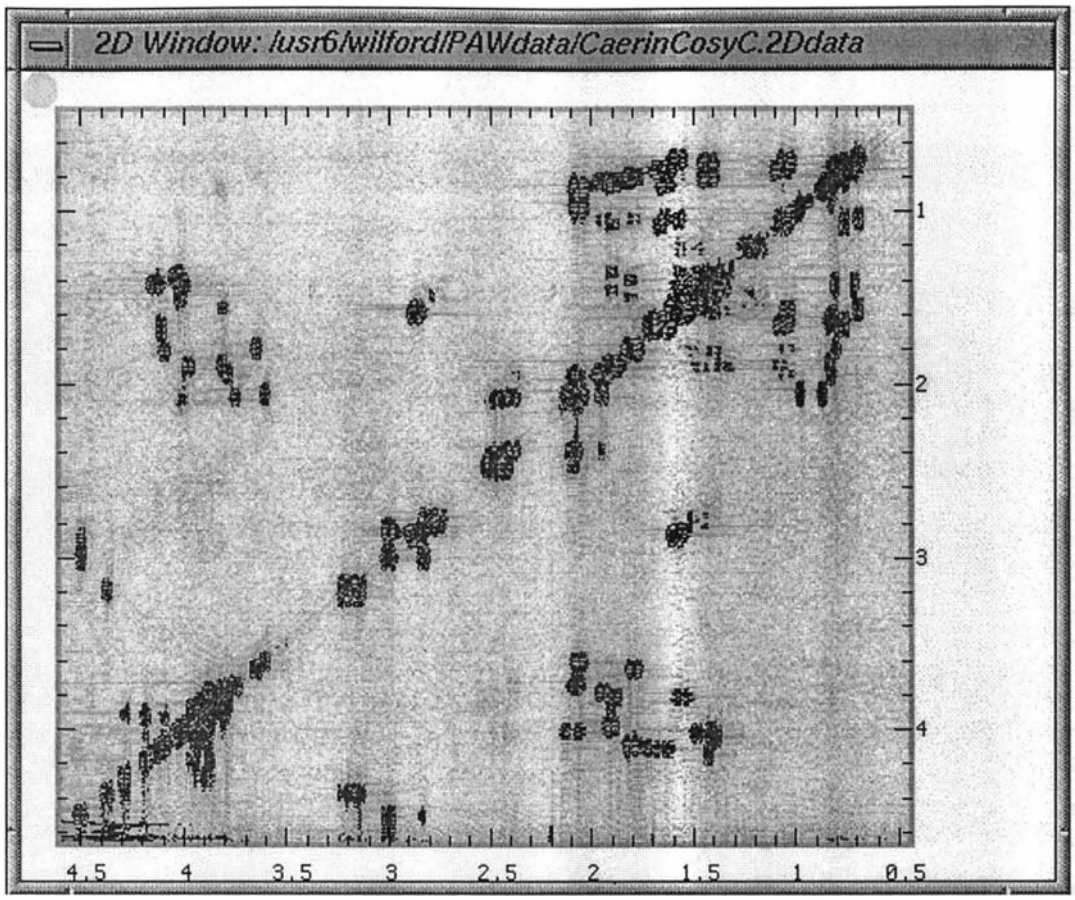


Figure 7.22: The upper-right region of **CaerinCosyC** that is displayed using a natural-log scale and colour scheme 2 without colour-map adjustment.

➢ To display in the natural-log scale

- Switch on the [DspInLog] button in the *Common-display Toolbox*.

This operation will switch on the natural-log scale flag in the calculation for 2D colour displays. In fact, the last two plots are displayed in a natural-log scale.

Note that, when switching to this scale PAW will automatically redisplay the plot in the intensity-only mode. After this, the plots will be displayed in the original mode.

➢ To display in the linear scale

- Switch off the [DspInLog] button in the *Common-display Toolbox*.

This operation will switch off the natural-log scale flag and result in the use of a linear scale for 2D colour displays. The following two plots are displayed using the linear scale.

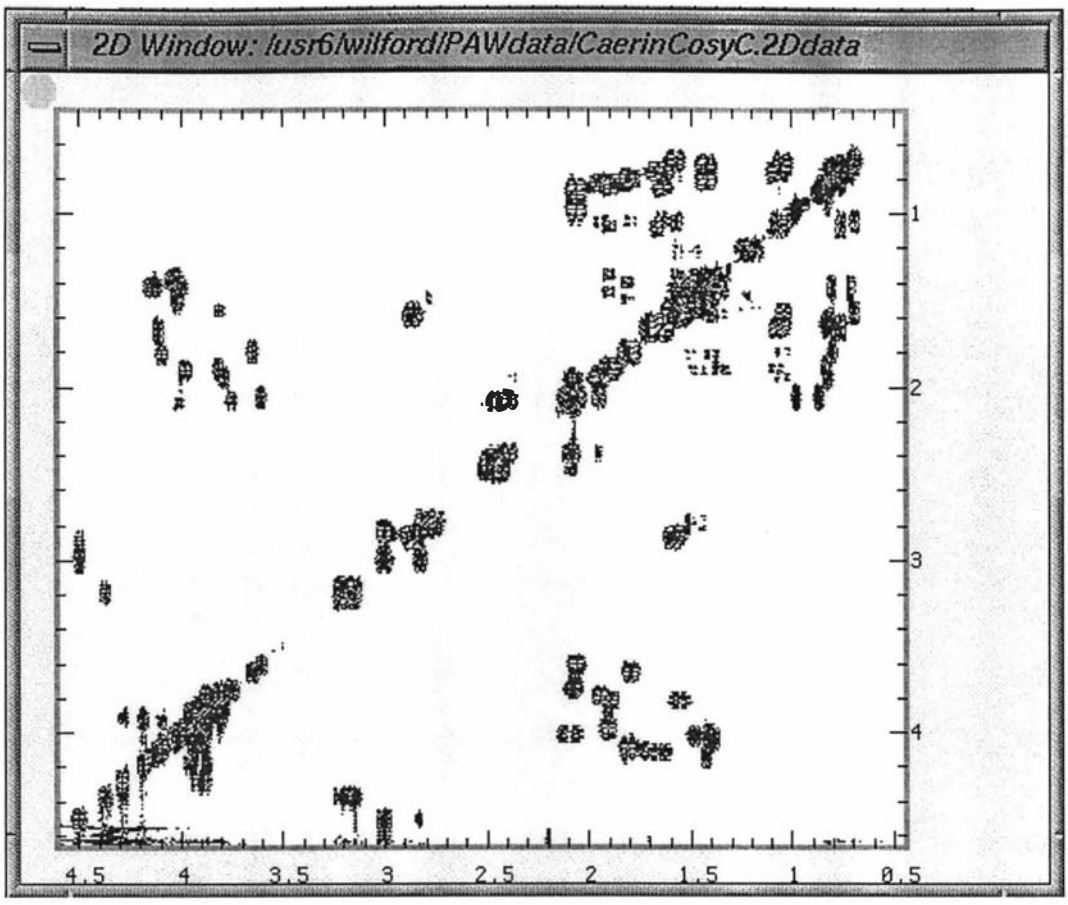


Figure 7.23: The same NMR data displayed using a linear scale and colour scheme 2 without colour–map adjustment.

The same data displayed using the linear scale and colour scheme 1 without colour–map adjustment is shown in the next figure (see section 7.3.2).

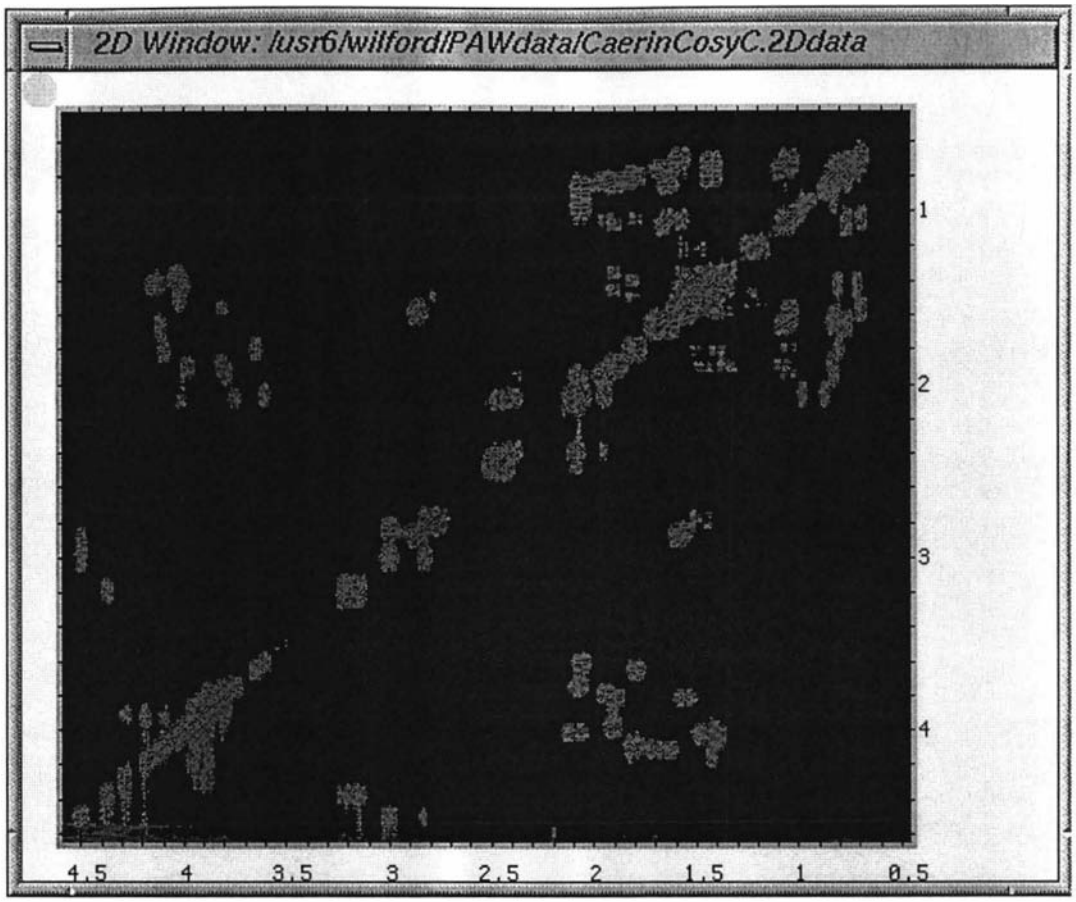


Figure 7.24: The same NMR data displayed using a linear scale and colour scheme 1 without colour–map adjustment.

Note that the background colour in each of the above display modes can be changed by adjusting the colour-map parameters. For example, a brighter background can be obtained when using the natural-log scale with colour scheme 2 by setting the zero-intensity threshold of the colour-map parameters to a higher value, as shown in the next figure. This plot was obtained by applying the following operations:

- Switch back to the colour scheme 2.
- Switch on the [DspInLog] button in the *Common-display Toolbox*.
- Choose [ColourMap] from the Display Menu to open the *SetColourMap Dialog*.
- Adjust the colour map by dragging the sliders in the dialog so that the values are, from left to right, 0, 18 and 255.
- Choose [Save] in the dialog.

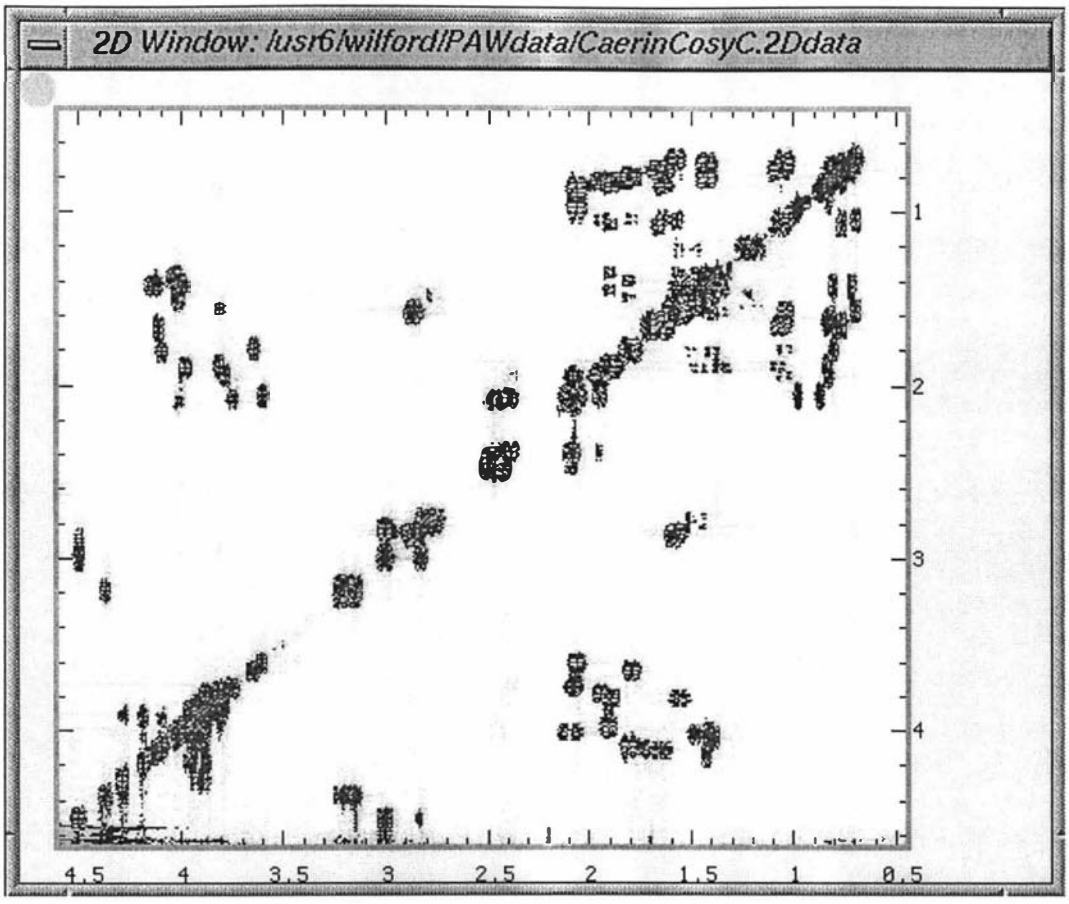


Figure 7.25: The same NMR data displayed in natural-log scale with colour scheme 2 and the zero-intensity threshold set to 18.

Two-dimensional NMR plots are often displayed in natural-log scale because this allows both weak and intense peaks to be viewed simultaneously.

7.4 Calibrating 2D Spectra

The calibration of 2D spectra is similar to that for 1D spectra and the same spectrum calibration dialog is used. All parameters in the dialog must be set. An example will be given in Chapter 8 after processing the Caerin 4.1 NMR data.

7.5 Plotting a 2D Display

This is similar to the 1D plotting process described in Chapter 5.

➢ **To plot a 2D display**

- Type pl2 or choose [Plot2D] in the *2D-processing Toolbox* to open the *2D-plotting Dialog* (Figure 7.26).

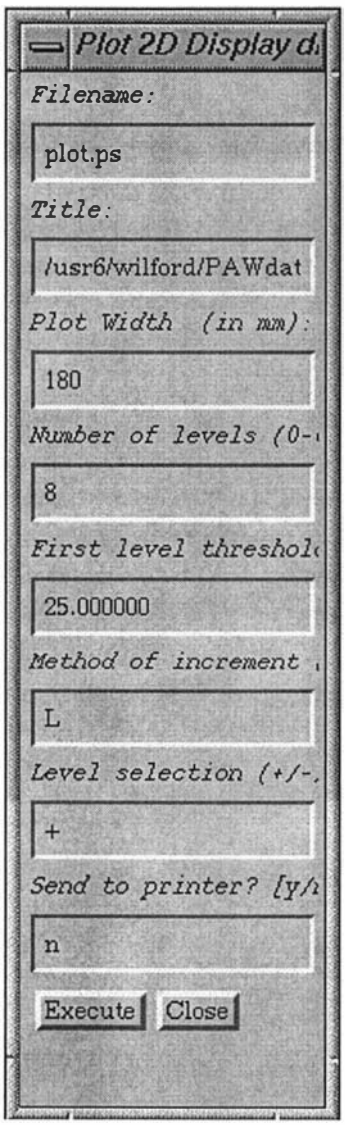


Figure 7.26: The *2D-plotting Dialog*.

The entries in the dialog are as follows:

- Entry 1 specifies the name of the output postscript file.
- Entry 2 specifies the plot header shown on top of the plot.
- Entry 3 specifies the width of the output plot in mm. (The height will be calculated from the shape of the plot on the screen.
- Entry 4 specifies the number of contour levels (from 0 to 64).
- Entry 5 specifies the first level threshold (from 0 to 64).
- Entry 6 specifies the method of increment between levels, either in even or natural log scale.
- Entry 7 specifies the sign(s) of peaks to be plotted, either +, – or b for both.
- Entry 8 specifies if the output is also sent to the printer.

Examples of 1D plots can be found in Chapter 11, Volume I.

Chapter 8:

Processing the Caerin 4.1 NMR Data

8.1 Introduction

This chapter describes the processing operations applied to three Caerin 4.1 NMR data sets. The operations for the Caerin 4.1 DQF-COSY data set are presented in detail; for the TOCSY and NOESY, only a brief description is given because the operations are similar.

All 2D graphics in this section are intentionally presented without colour-map adjustment to display their intensity background on the actual screen. No comments on the results are presented in this chapter, as they can be found in Volume I of the thesis. In addition, it is expected that the reader is familiar with the basic 1D and 2D operations described previously.

The workbench for the operations in Chapter 7 is also used in this chapter. It contains three 2D draw-windows, one 1D draw-window, and a number of toolboxes.

During the process, the commands provided by the display and processing toolboxes for both 1D and 2D operations are frequently used. Of these, only the *2D-processing Toolbox* has not been described.

➢ **The *2D-processing Toolbox***

This contains command buttons for 2D processing, as shown in Figure 8.1.

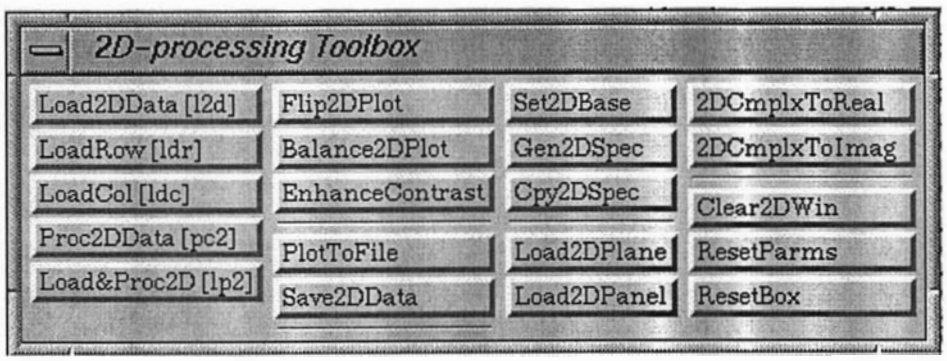


Figure 8.1 The 2D-processing Toolbox.

To open the toolbox, either type o2p or choose [2DProcTBox] from the *Process Menu*. To close it, either type x2p or double-click on the window-control button at its top-left corner.

8.2 Processing the Caerin 4.1 DQF-COSY NMR Data

This section describes the processing operations of the Caerin 4.1 DQF-COSY NMR data. A conventional method is presented first to obtain a DQF-COSY spectrum, which, unfortunately, cannot be properly phased. This is followed by the high-resolution processing method that produces a properly-phased and better-resolved spectrum.

8.2.1 Preliminary

➢ **Loading the raw data**

- Choose [Load2DData] in the *2D-processing Toolbox* to open a file-selection dialog.
- Double-click on the filename **caerin/data5dqfcosy.2Ddata** to load the data.

Figure 8.2 shows the intensity plot of the Caerin 4.1 DQF-COSY raw data.

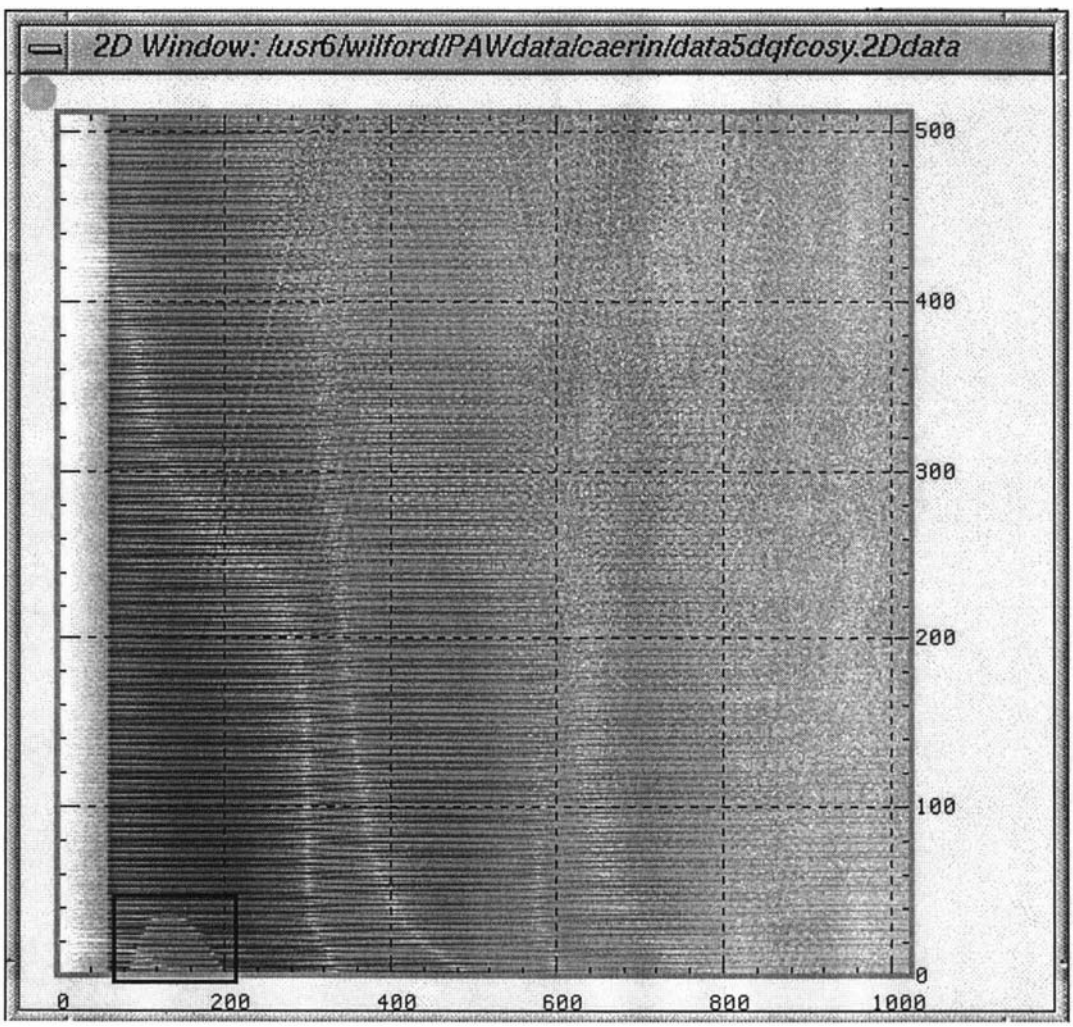


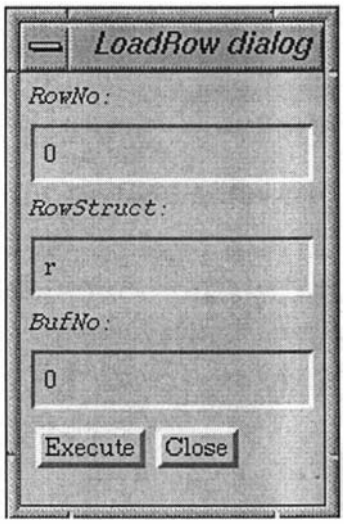
Figure 8.2 The intensity plot of the Caerin 4.1 DQF-COSY raw data. The blocked area contains cut-off values due to receiver overflow.

The white area on the left contains data added by the digitally filtered and over-sampled method in the experiment. The blocked area at the lower-left corner contains spurious values due to receiver overflow. Further details of these unusual data features are discussed in the next subsection.

➢ Inspecting problems in the raw data

The following operations load four rows of the DQF-COSY data into different 1D buffers for inspection. (See also Chapter 5 for the loading method.)

- Turn off the [Dsp1DRealOnly] button in the *1DDspMode Check-box* of the *1D-display Toolbox*. This enables the display of both real and imaginary data in two 1D slices.
- Choose [DspBufs] of the same check-box to display more than one 1D buffer.
- Turn on buttons [0], [1], [2] and [3] of the *1DDspBuf Check-box* in the *1D-display Toolbox* to display 1D buffers #0 to #3.
- Type ldr to open the *LoadRow Dialog* (Figure 8.3).



- Enter 0, c and 0 for the three fields in the dialog. These are the row number, data structure (c for complex), and buffer number, respectively.
- Choose [Execute] in the dialog to load Row 0 into Buffer 0.
- Repeat the last three steps to load rows 200, 400 and 511 into buffers 1, 2 and 3.

Figure 8.3 The 2D-processing Toolbox.

Figure 8.4 shows the result.

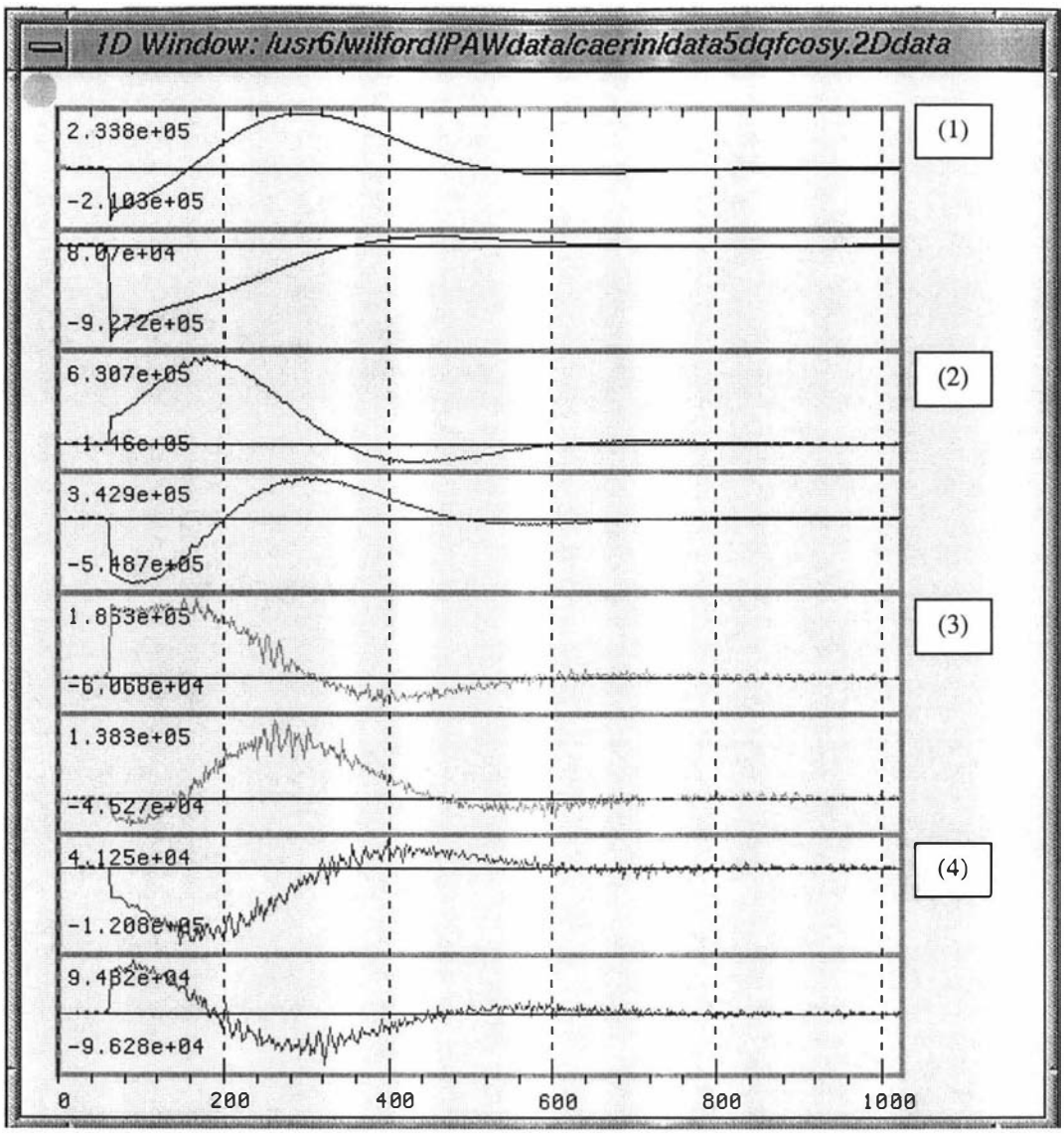


Figure 8.4 The real and imaginary parts of Row 0, 200, 400 and 511 of the Caerin 4.1 DQF-COSY. The predominant water signal is characterised by a low-frequency curve in each FID.

The following operation provides an expanded view of an initial part drawn from a row of the data.

- Click in the 1D draw-window.
- Type zs. The message "*Use Button 1 to set a region.*" will appear on top of the window.
- Select a region from point 0 to around 70.

Figure 8.5 shows that the first 62 values of a row form a zigzag-like curve.

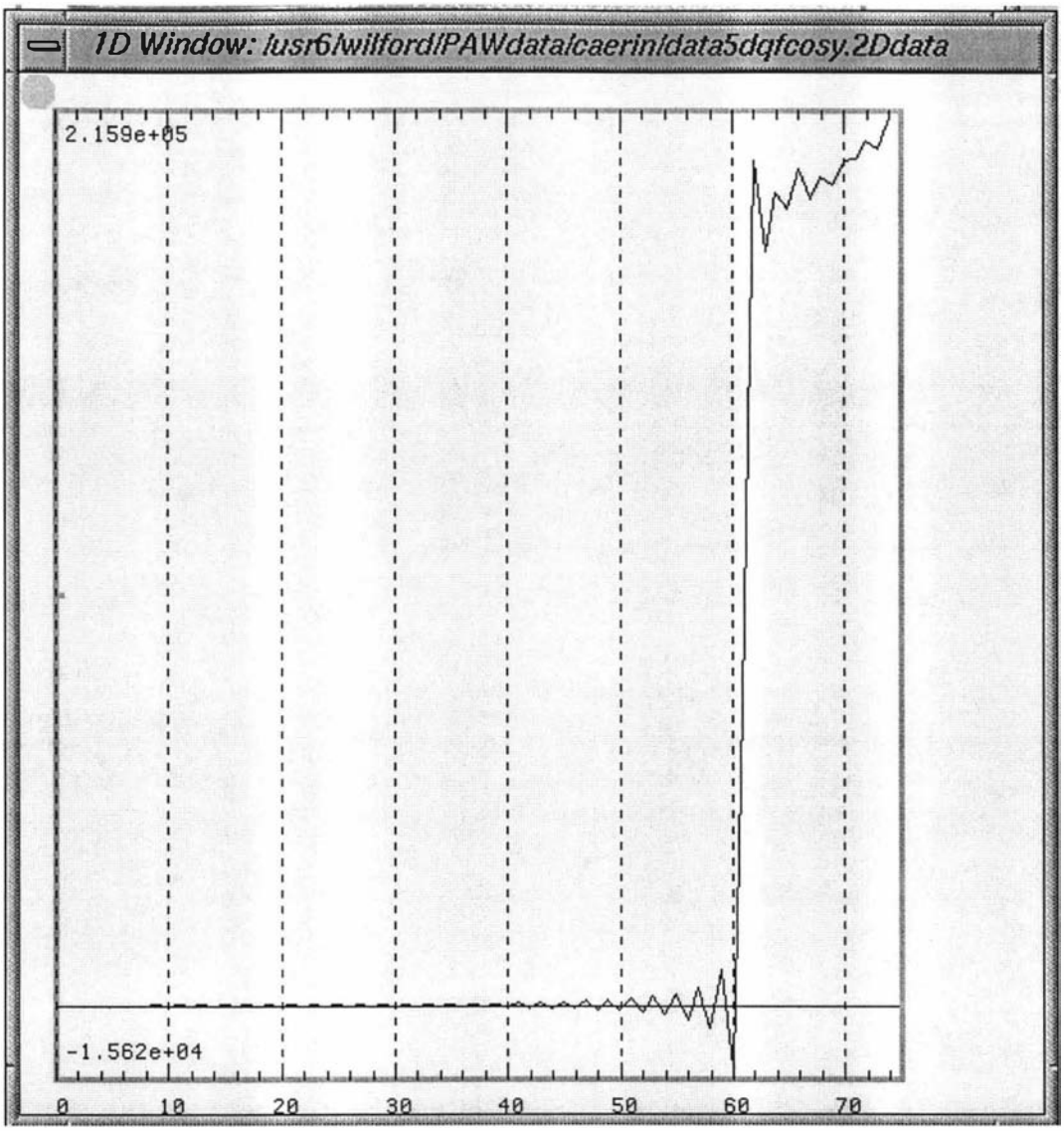


Figure 8.5 An expanded view of a time-domain data set characterised by a filter response build-up curve at the beginning. The zigzag-like curves at the beginning characterise the filter-response data obtained using the digitally filtered and over-sampled method in the experiment.

Similar loading operations can be used to inspect the damaged rows due to receiver overflow, as shown in Figure 8.6.

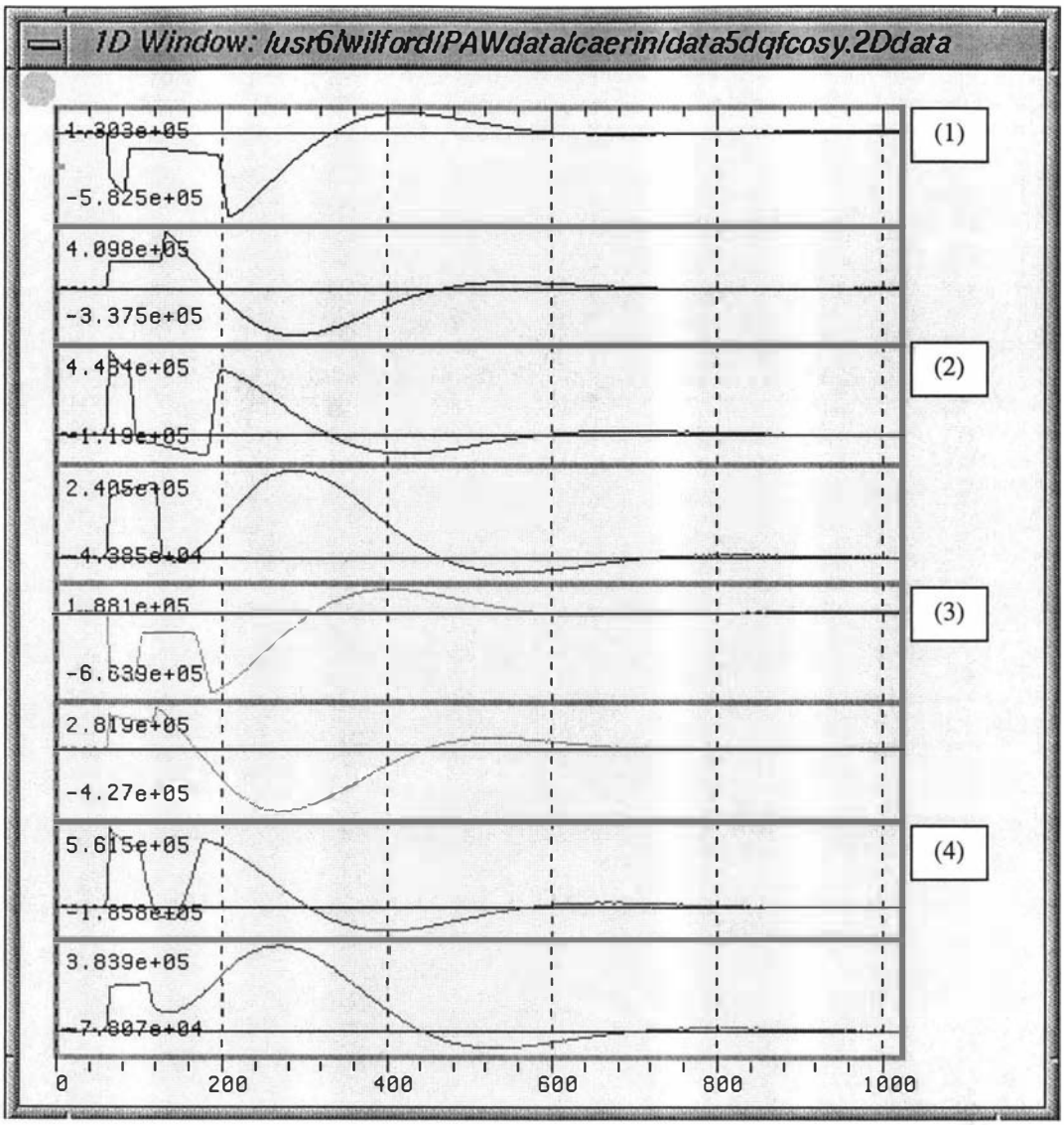


Figure 8.6 The real and imaginary plots of rows 1, 11, 21 and 31 of the Caerin 4.1 DQF-COSY. The platform-like regions characterise the receiver-overflow problem. A total of 44 rows were damaged in the same way.

8.2.2 Conventional processing operations and results

This subsection describes the conventional operations for processing the Caerin 4.1 DQF-COSY data. It involves the removal of the unusual initial parts and the time-domain water-signal suppression in D1, as well as conventional filtering and phasing in both dimensions.

➢ Processing in D1

The following operation loads Row 100 of the Caerin 4.1 DQF-COSY data, filters the row with a sine function, and applies a fast Fourier transform.

- Disable 1D buffer display by turning off the [DspBufs] button in the *1DDspMode Checkbox* of the *1D-display Toolbox*.

- Load Row 100. (See Chapter 8 of Volume I for the reason.)
- Choose [ShiftLeft] in the *1D-processing Toolbox* to open the *ShiftLeft Dialog* (Figure 8.7).

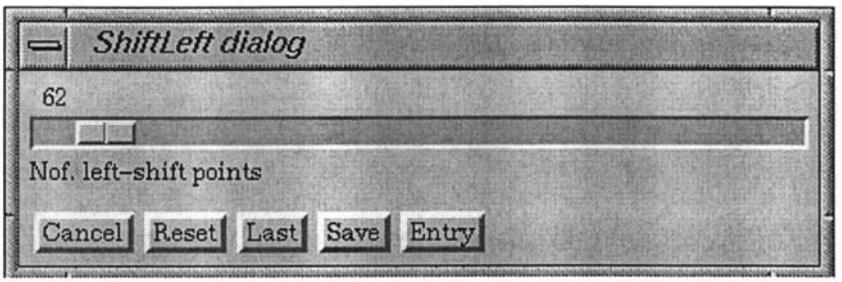


Figure 8.7 The *ShiftLeft Dialog*.

- Adjust the slider (using MsBtn#1) so that the Number of Left-shift Points is 62.
- Choose [Save] in the dialog.
- Choose [T-dWSuppr] in the *1D-processing Toolbox* to suppress the water signal in time domain.

The result is shown by Plot 1 in Figure 8.8, along with the original and left-shifted data.

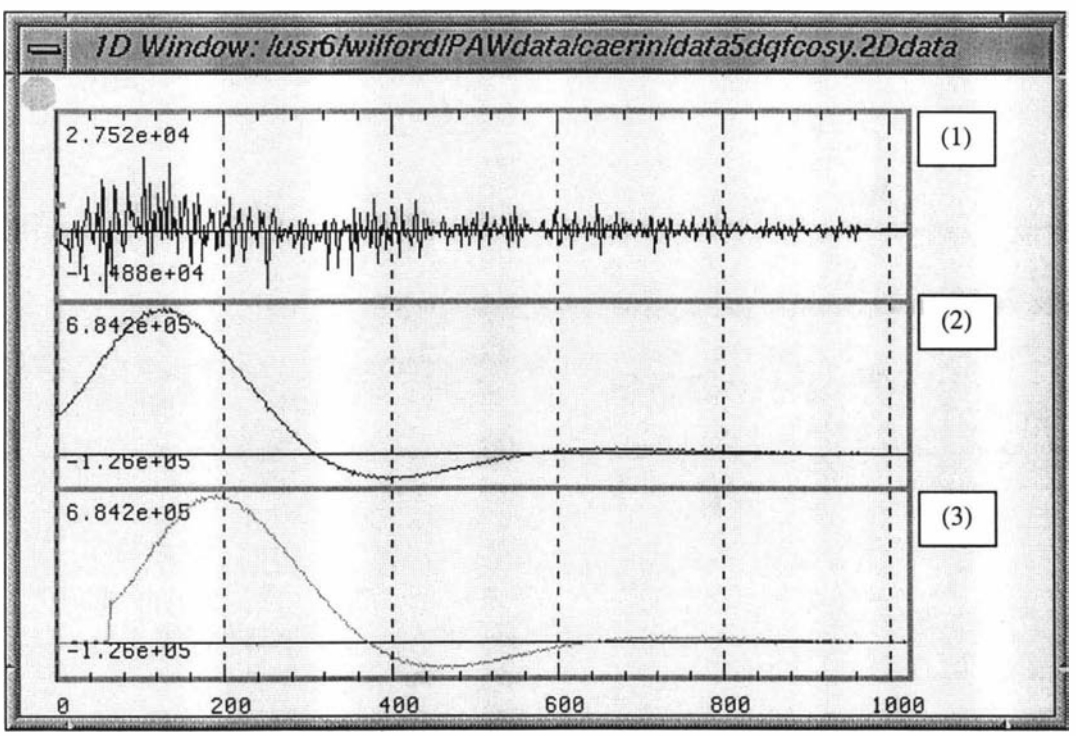


Figure 8.8 Row #100 and two processed results. The plots are, from bottom to top, the original, left-shifted, and water-signal-suppressed data.

Note that the [WrtBufX→Y] and [SwapBufX,Y] buttons in the *1D-MiscProc Toolbox* have been used to produce the figure. The operations are not necessary for the following processing, and hence are omitted.

The following operations apply a filter and a Fourier transform to the data.

- Choose [Filter] in the *1D-processing Toolbox* to open the *Signal-filtering Toolbox* (Figure 8.9).

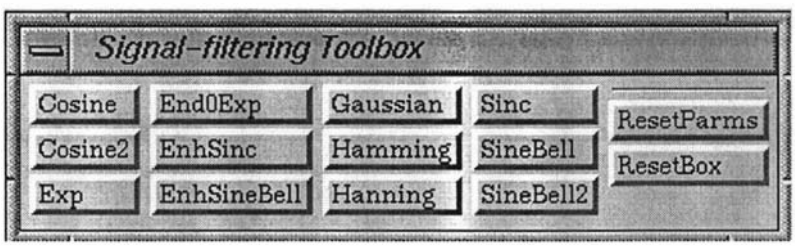


Figure 8.9 The Signal-filtering Toolbox.

- Choose [EnhSineBell] in the toolbox to open the dialog for the enhanced sine-bell filter (Figure 8.10).

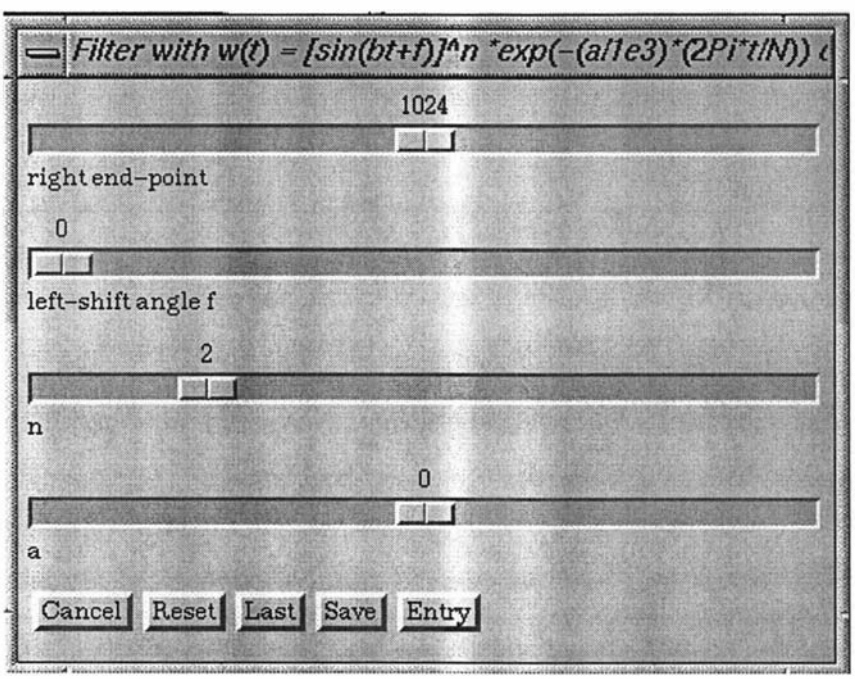


Figure 8.10 The dialog for the enhanced sine-bell filter.

- Adjust the sliders (using MsBtn#1) so that their values are, from top to bottom, 960, 0, 1 and 0. The first three parameters are the end-point from which the filter becomes zero, the angle by which the sine function is left-shifted, and the degrees of the sine function. The last parameter is the degrees of exponential enhancement.
- Choose [Save] in the dialog.
- Type fft to apply a fast Fourier transform.

The first-order sine-bell filter with no exponential enhancement applies a full suppression to the initial part by setting the shift-angle to 0 degree. In addition, setting the end point of the sine-bell filter to 960 makes the end part gradually reduce to zero in order to suppress noise.

The results are shown in Plot 1 and 2 of Figure 8.11. Also shown in the figure are the corresponding results without water-signal suppression.

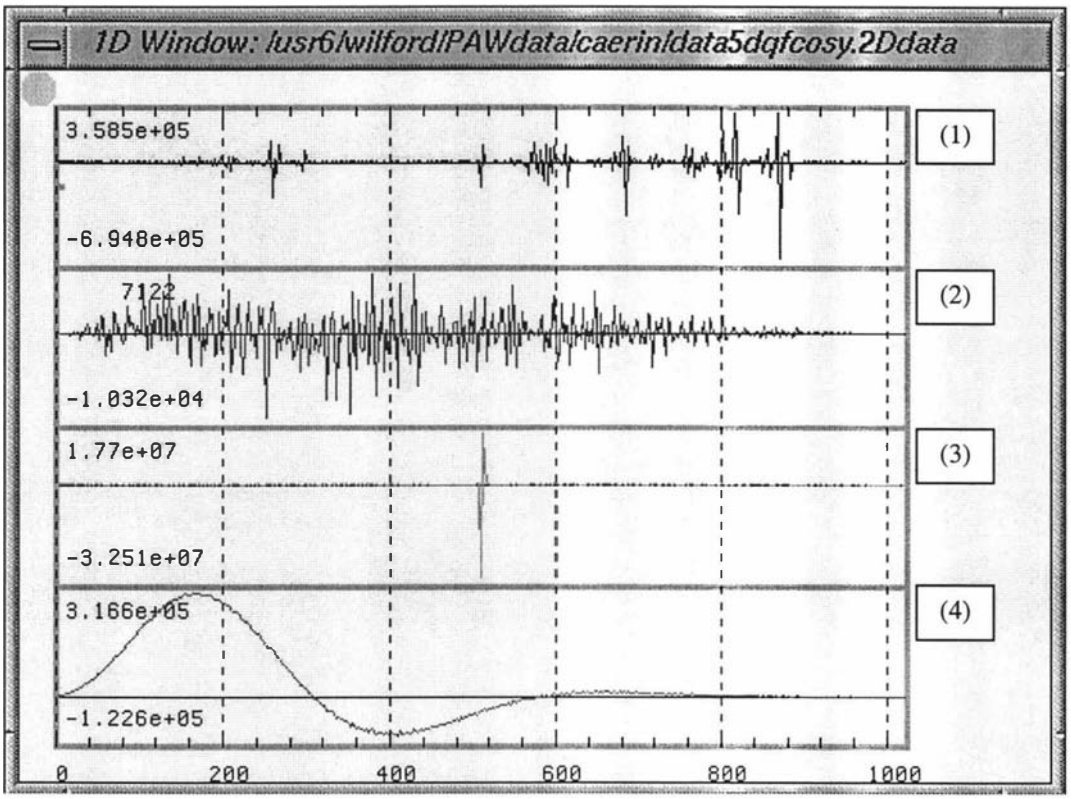


Figure 8.11 Second-stage results of processing Row #100. Plot 2 is the result after filtering, and Plot 1, after Fourier transforming. Plot 4 and 3 are the corresponding results without water-signal suppression.

The following operations determine the phasing parameters for the row.

- Disable the 1D-buffer- and filter-display options by turning off [DspBufs] and [DspFlt] in the *1D-display Toolbox*.
- Choose [Phase] in the *1D-processing Toolbox* to open the *Phasing Toolbox* (Figure 8.12). The message "Please set the Baselevel if AutoPh* is to be used." will appear in a pop-up window, where * is a wildcard character that can either be 0 or 1.

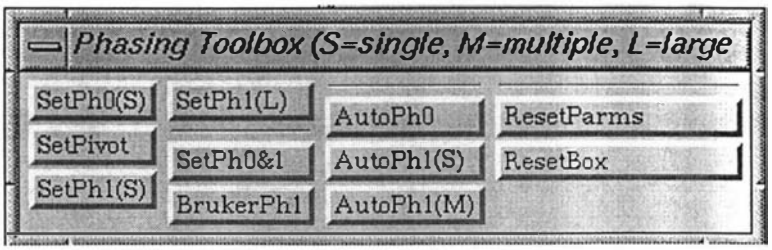


Figure 8.12 The *Phasing Toolbox*.

- Choose [OK] in the pop-up window to close it, and ignore the base-level setting at this stage because there is no need to use the auto-phase commands in the conventional processing method for the DQF-COSY data.
- Click in the 1D draw-window, then type zs.
- Select a region from points 460 to 560 approximately.
- Type ph0 or choose [SetPh0(S)] in the *Phasing Toolbox* to open the *SetPh0 Dialog* (Figure 8.13).

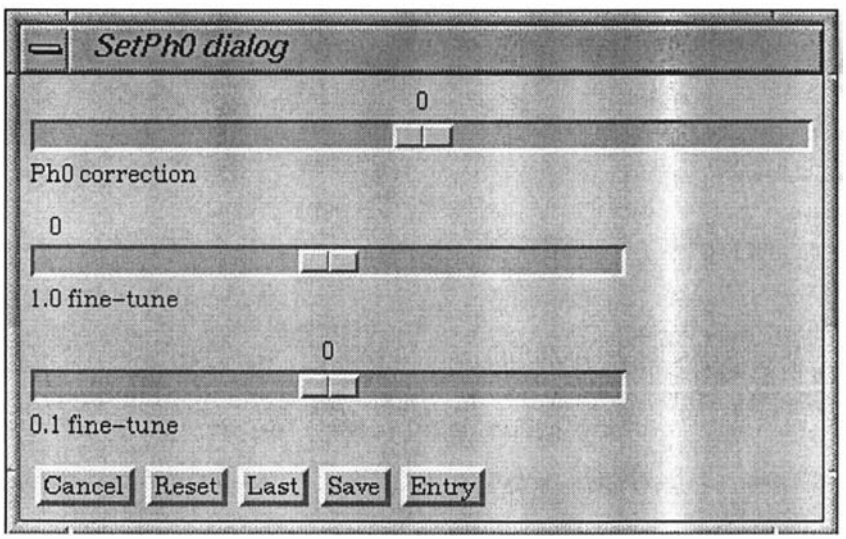
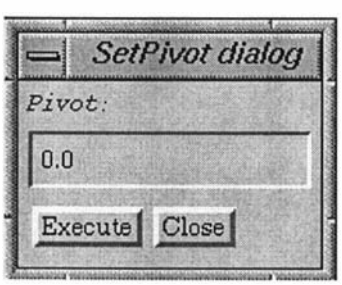


Figure 8.13 The *SetPh0 Dialog*.

- Adjust the sliders so that the sum of their values is −10 degrees. Then, choose [Save] in the dialog.
- Type pvt or choose [SetPivot] in the *Phasing Toolbox* to open the *SetPivot Dialog* (Figure 8.14).



- Enter 510 in the dialog, then choose [Execute]. (Alternatively, use MsBtn#2 to select a point close to 510 on the 1D plot.) The message "Pivot = 510.000" will appear on the Unix shell.

Figure 8.14 The *SetPivot Dialog*.

- Type ph1 or choose [SetPh1(S)] in the *Phasing Toolbox* to open the *SetPh1 Dialog* (Figure 8.15).

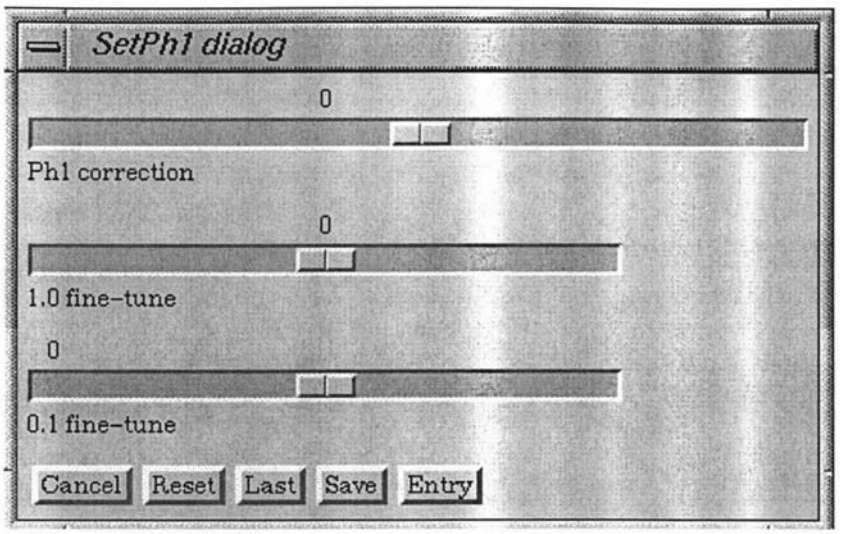


Figure 8.15 The *SetPh1 Dialog*.

- Adjust the sliders so that the sum of their values is –200 degrees. Then, choose [Save] in the dialog. (Note that to check this value, smaller regions on both sides of the spectrum must be inspected. These operations are omitted here.)

Figure 8.16 shows the results of the above operations.

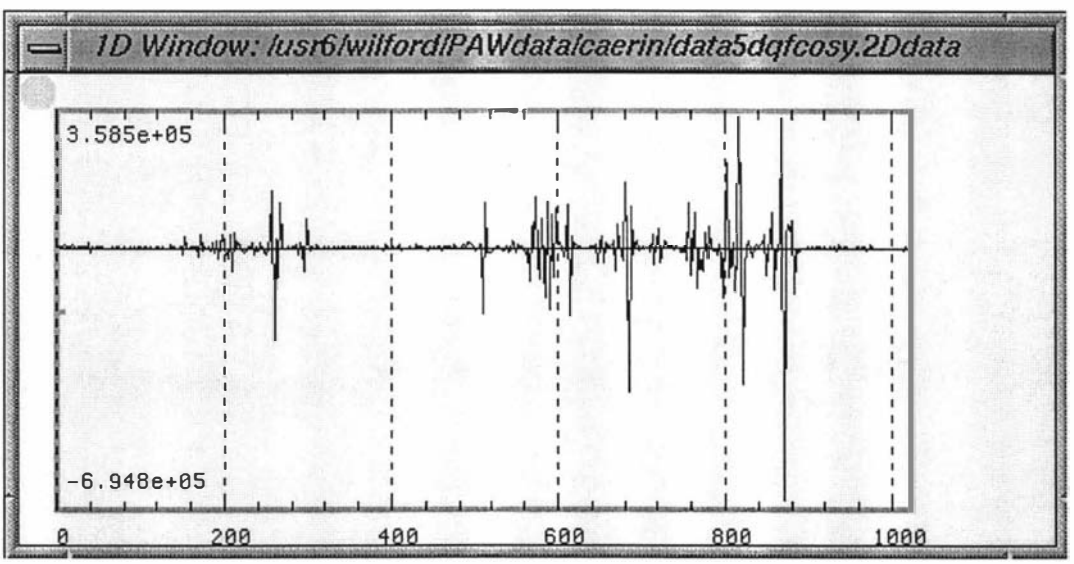


Figure 8.16 The final processing result for Row 100.

The operations above can be summarised into the following five macro commands:

```
ldr (100, c)          # Load a complex row
shl (62)              # Left-shift
tws                   # Time-domain water-signal suppression
esb (960,0,1,0)       # Enhanced sine-bell
fft                   # Fast Fourier transform
ph  (-10,-200,510)    # phase
```

PAW provides a command called pc2 (or [Proc2D]) to perform a set of common 2D-processing operations using a dialog. However, the dialog does not contain the uncommon shl (i.e., left-shift) and tws (i.e., time-domain water-signal suppression) operations. Therefore, a special treatment is required to suppress the water signals and convert the digitally filtered over-sampled 2D data set into a normal one. This can be achieved by running a macro as follows:

```
# CrnCosyD1Shift.mcr
D1 = 1024           # Dimension size in D1
D2 = 512            # Dimension size in D2
D1Struct = c        # Data structure in D1
D2Struct = r        # Data structure in D2
for (%I=0, 511, 1)      # Start a FOR-loop
  ldr (%I, c)       # Load a row
  shl (62)          # Left-shift
  tws          #Time-domain water-signal suppression
  wrr (%I, c)       # Write a row
next           # End a FOR-loop
dr1            # Draw a 1D plot
dr2            # Draw a 2D plot
```

To run the macro,

- Type run to open a file-selection dialog, then double-click on the name **CrnCosyD1Shift.mcr**.

Figure 8.17 (next page) shows the result after running the macro.

After this treatment, a standard macro format is required to perform the pc2 command. The macro for setting up parameter values for processing the Caerin DQF-COSY with the conventional method is as follows:

```
# CrnCosyShifted.2Dproc
MD1  = 1024         # Matrix size in D1
MD2  = 1024         # Matrix size in D2

lpCom1    = "n"     # D1 linear-prediction command
lpCom2    = "n"     # D2 linear-prediction command

fltCom1   = "esb (960, 0, 1, 0)"  # D1 filtering command
fltCom1   = "esb (512, 0, 1, 0)"  # D2 filtering command

phCom1    = "ph (-10, -200, 510)"      # D1 phasing command
phCom2    = "ph ( 88, -6,   512)"      # D2 phasing command

bcCom1    = "n"     # D1 baseline-correction command
bcCom2    = "n"     # D2 baseline-correction command

DimNoToTrans = b    # Dimension to transform
ExprmtType   = t    # Experiment type (t for TPPI)

# Set baseline segments
sbs(4, 340,480, 44,140, 900,930, 960,1020)
```

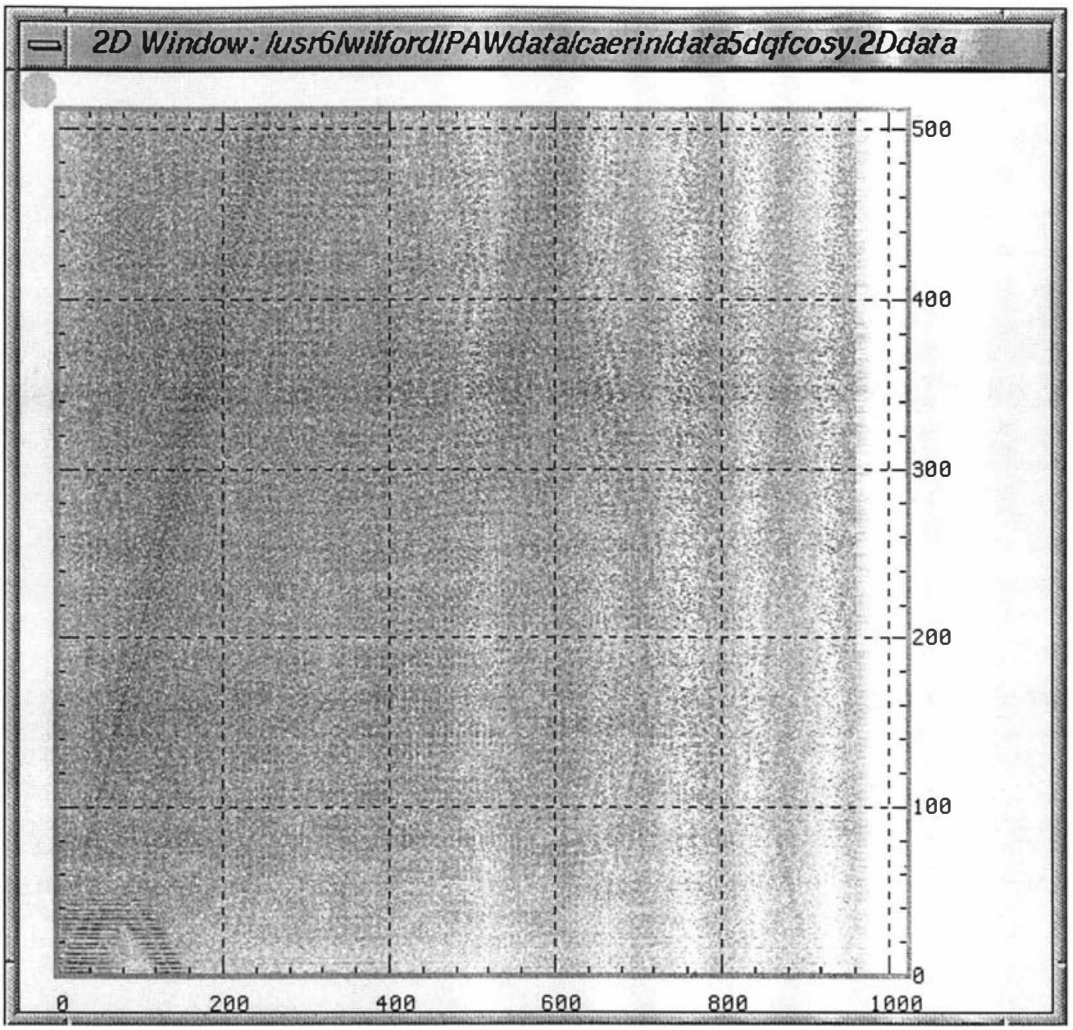


Figure 8.17 A full view of the Caerin 4.1 DQF-COSY spectrum after it has been left-shifted 62 points and water-signal suppression has been applied to each row.

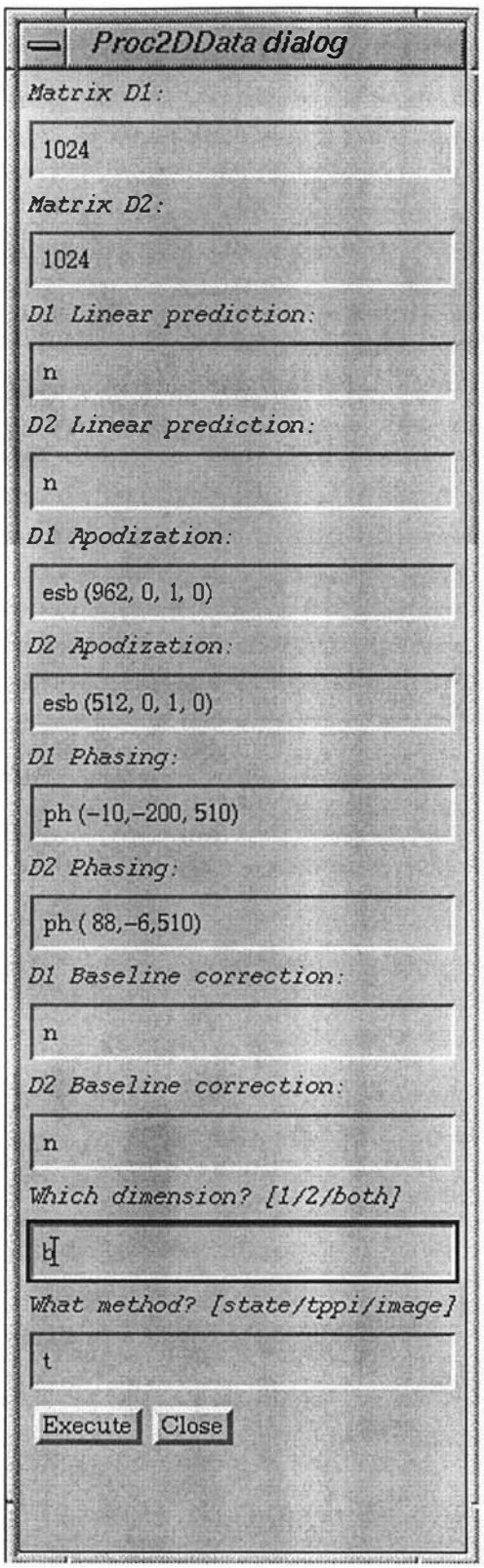


Figure 8.18 The Proc2DData Dialog.

The sbs statement in the macro sets four baseline segments, which are [340,480], [44,140], [900,930] and [960,1020]. These segments are selected from the 1D spectrum of Row 100. (Ignore the commands for processing in D2 at this stage, for which the parameters will be obtained later.)

Note that the 2D processing routine of PAW requires the baseline segments be correctly defined for correcting baselines and removing offsets. The first segment defined by the sbs command is used to calculate base-level parameters for every row and column of data, including the mean and standard deviation. The segment [340,480] is selected here to be the first simply because it is the largest baseline segment found in the 1D spectrum, and is the best for the base-level calculation.

To run the macro for processing in D1,

- Type pc2 or choose [Proc2Ddata] in the *2D-processing Toolbox* to open a file-selection dialog.
- Double-click on **CrinCosyShifted.2Dproc**. This will load the macro and open the *Proc2DData Dialog* (Figure 8.18).
- Change the value for the 'Which dimension?' entry to 1, then choose [Execute].

The operation produces an intermediate 2D data set, as shown in Figure 8.19, where the top half is filled with zeros to expand the D2 dimension.

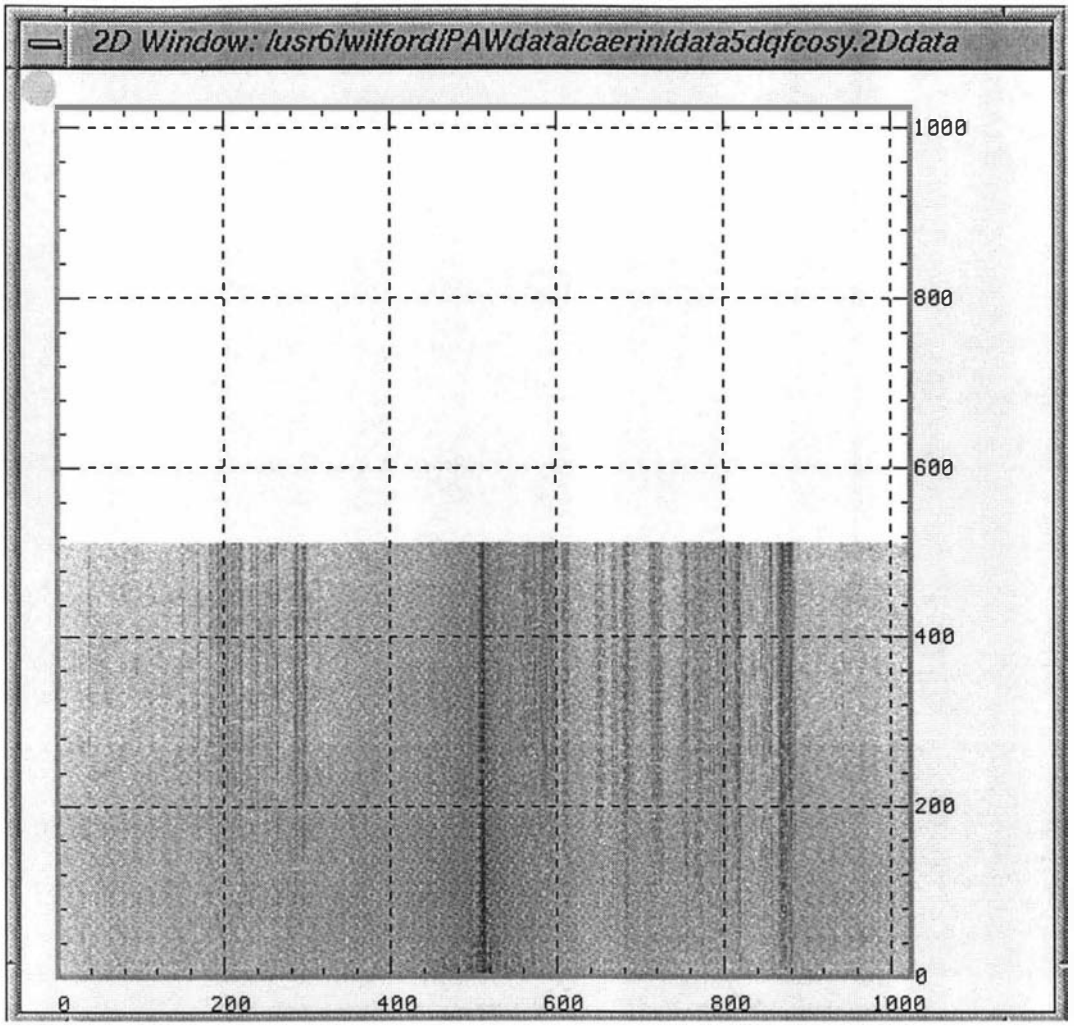


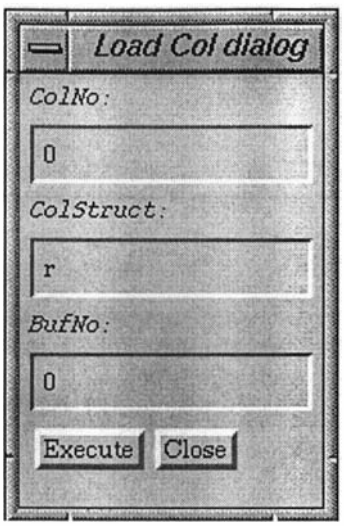
Figure 8.19 A whole view of the Caerin 4.1 DQF-COSY spectrum after D1--processing with conventional sine filter.

➢ Processing in D2

Processing in D2 is similar to that in D1, except that it requires at least two columns of data be combined in order to obtain proper phasing parameters.

The following operations combine columns 37 and 885 into one.

- Type ldc or choose [LoadCol] in the *2D-processing Toolbox* to open the *LoadCol Dialog* (Figure 8.20).



- Change the value for the 'ColNo' entry to 37, then choose [Execute] to load Column 37 into Buffer 0.
- Choose [WrtBuf0→1] in the *1D-MiscProc Toolbox* to save the result into Buffer 1.
- Repeat the first two steps to load Column 885 into Buffer 0.
- Choose [AddBuf0,1] in the *1D-MiscProc Toolbox* to add the data in Buffer 1 to Buffer 0.

Figure 8.20 The LoadCol Dialog.

The following operations process the combined data.

- Apply the enhanced sine-bell filter with the parameters 512, 0, 1 and 0.
- Type rft or choose [RFT(smth)] in the *1D-processing Toolbox*.
- Phase the 1D spectrum with the parameters 88, –6 and 512.

Figure 8.21 shows the combined column and the filtered result.

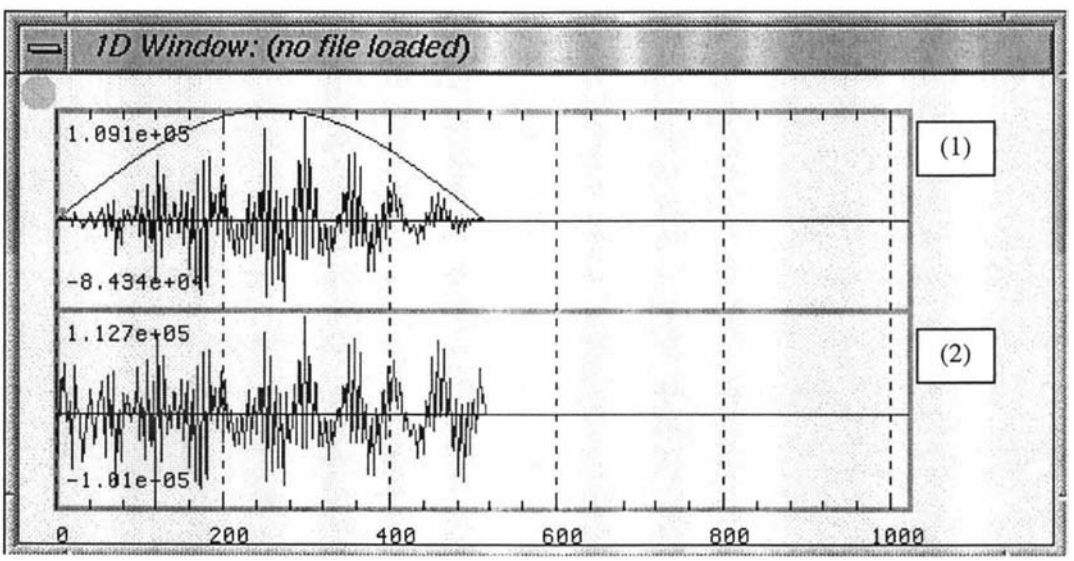


Figure 8.21 The combined column and the filtered result.

Figure 8.22 shows the FFT and phase results for the combined column.

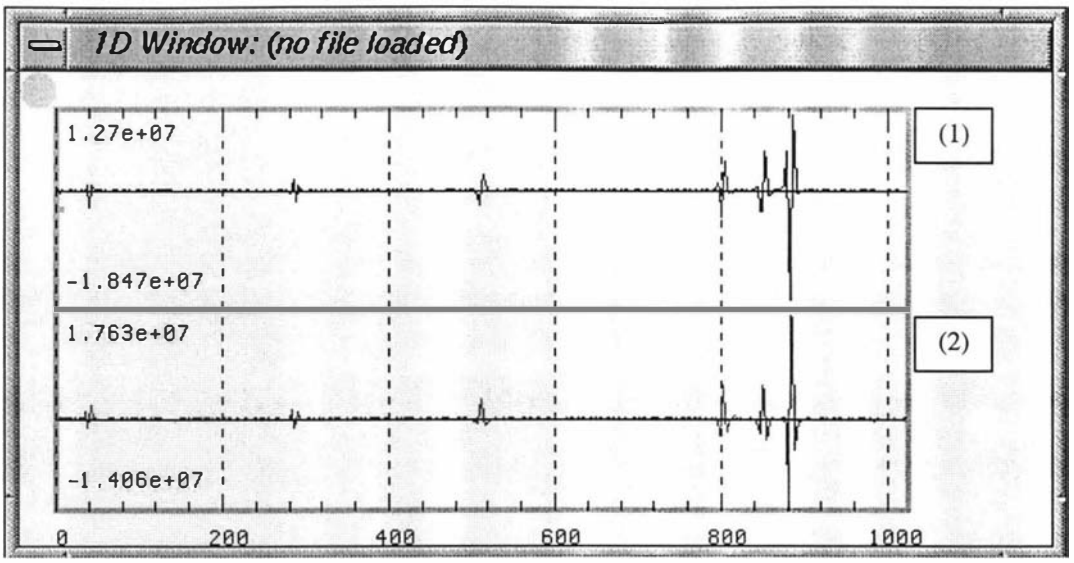


Figure 8.22 The FFT result (bottom) and phase result (top) for the combined column.

Figure 8.23 shows a multi-region view of the FFT and phase results.

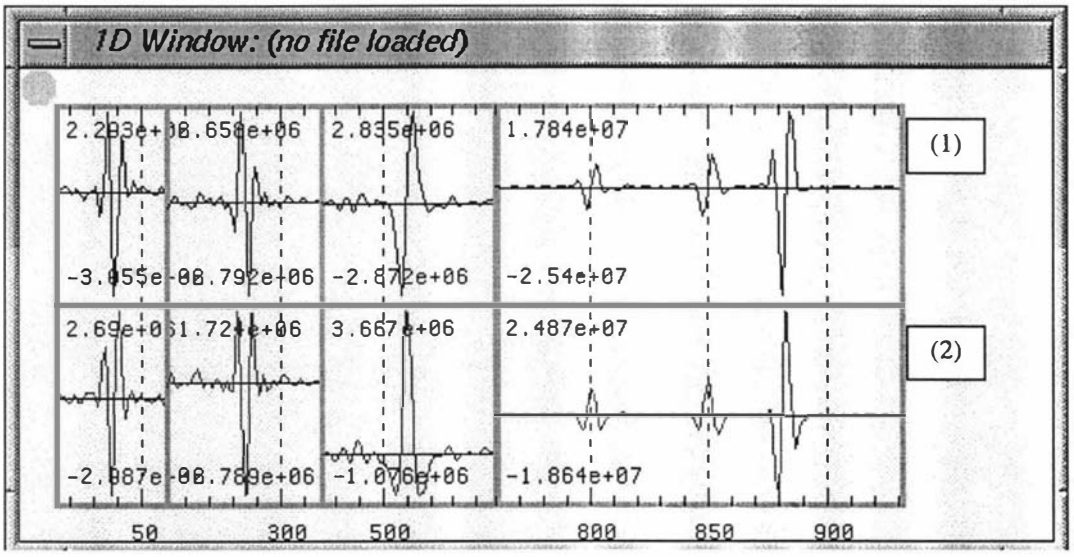


Figure 8.23 A multi-region view showing some significant peaks. The bottom result is after FFT, and the top, after FFT with phase correction.

Note that not all peaks can be phased to the desired shapes because of the problem in the shifted data. The criteria for choosing the above parameters are the acceptable anti-phase line-shapes for the peaks centred at points 285, 800 and 850.

The operations above can be summarised into three macro commands:

```
esb (512,0,1,0)          # Enhanced sine-bell
rft                      # Real Fourier transform
ph  (88,-6,512)          # phase
```

These commands have been embedded into **CrnCosyShifted.2Dproc** presented in the last subsection, where the t for the ExprmtType defines the use of RFT for column processing.

To load **CrnCosyShifted.2Dproc** for processing in D2,

- Type pc2 or choose [Proc2Ddata] in the *2D-processing Toolbox* to open a file-selection dialog.
- Double-click on the filename **CrinCosyShifted.2Dproc** in the dialog. This will load the macro and open the *Proc2DData Dialog* (see Figure 8.18 in the last subsection).
- Change the value for the 'Which dimension?' entry to 2, then choose [Execute].

The operation produces a processed 2D DQF-COSY spectrum, as shown in Figure 8.24.

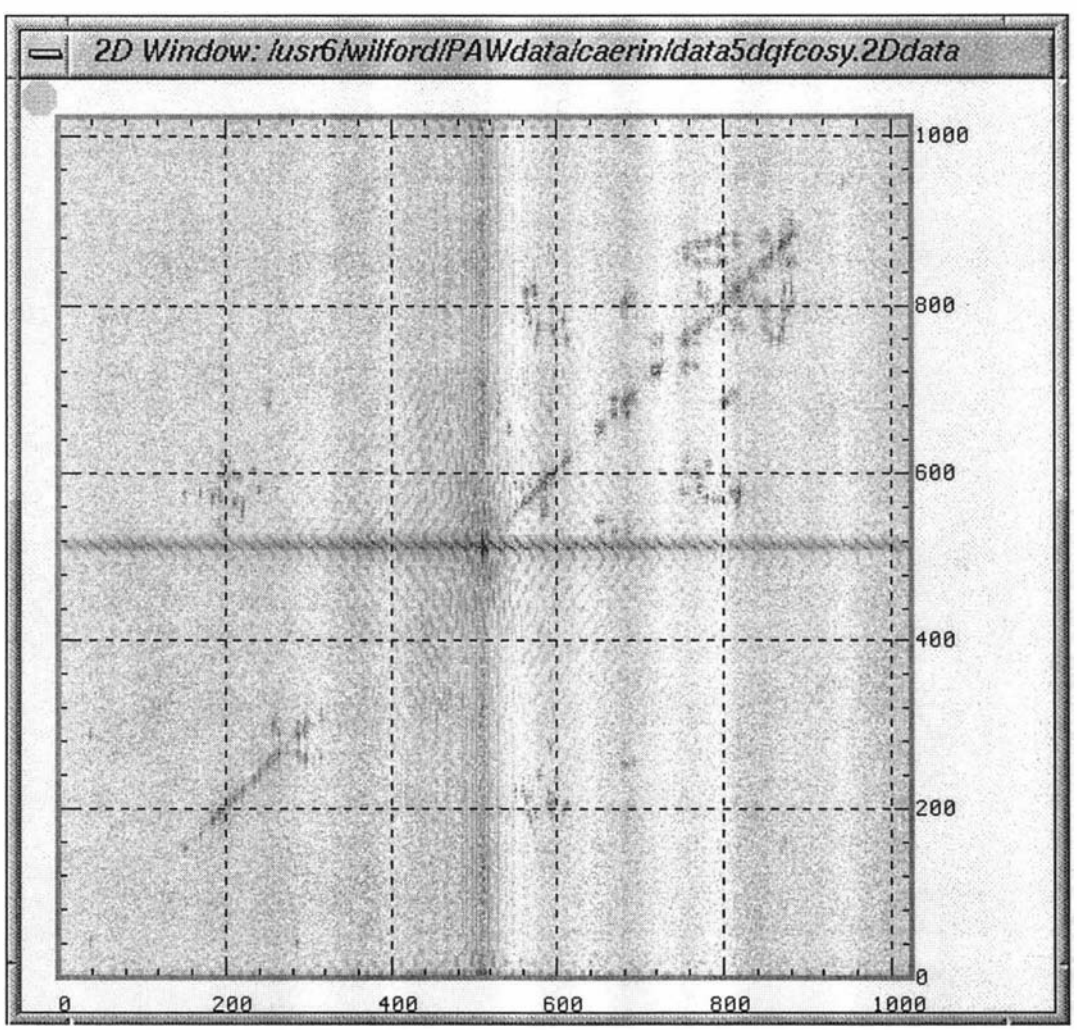


Figure 8.24 A full view of the Caerin 4.1 DQF-COSY spectrum after D2 -processing with conventional method.

The following operations display an expanded view of the upper-left (fingerprint) region in the last figure:

- Click on the DQF-COSY draw-window.
- Enable the intensity and contour plotting mode by turning on [Inten.+Contour] in the *2D-display Toolbox*.
- Type zs.
- Select the fingerprint region.

The result is shown in the next figure.

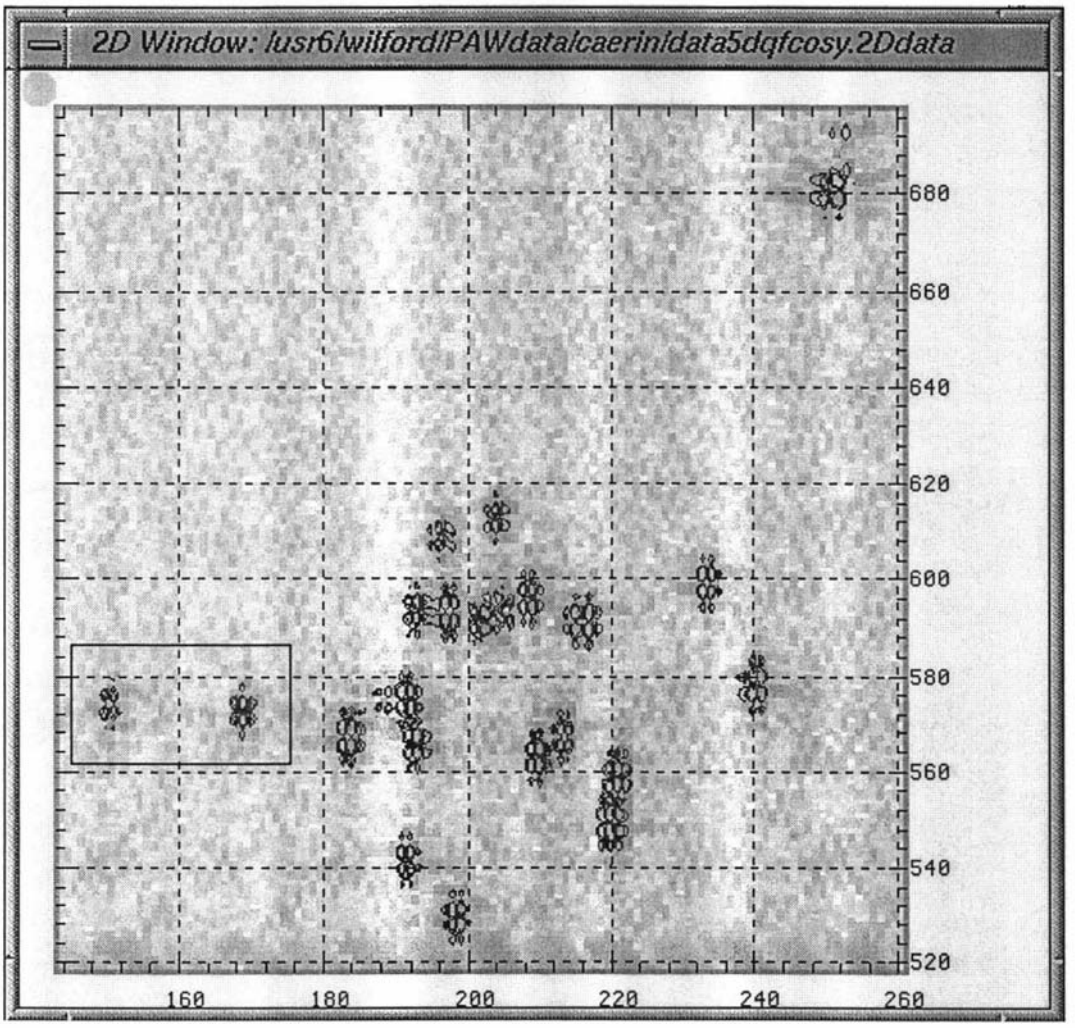


Figure 8.25 The upper-left (fingerprint) region of the Caerin 4.1 DQF-COSY spectrum processed by a conventional method.

Comments on the results can be found in Chapter 8 of Volume I.

8.2.3 High-resolution processing operations and results

This subsection describes the high-resolution processing operations for the Caerin 4.1 DQF-COSY data. It involves the application of an enhanced sine-bell filter and a large first-order phase correction in D1, as well as linear prediction and baseline correction in D2.

To maintain a compact description, the reader is assumed to be familiar with the basic operations described previously, including those in the last subsection. Therefore, only the relatively unfamiliar operations such as linear prediction and auto-phase correction are described in detail.

➢ Processing in D1

This sub-subsection describes the processing in D1 for the DQF-COSY spectrum. It involves loading and processing Row 100 of the data with an enhanced sine-bell filter, Fourier transforming, and the search for proper phasing parameters using automatic phase correction. The operations are then summarised and embedded into a macro containing commands used in the final processing.

The following operation loads Row 100 of the DQF-COSY data, filters the row with an enhanced sine-bell function, and applies a fast Fourier transform.

- Disable the 1D buffer display.
- Load Row 100 into Buffer 0. (This is selected for the same reason mentioned in the last subsection.)
- Apply the enhanced sine-bell filter with the parameters 1024, 70, 2 and −100.
- Type fft to apply a fast Fourier transform.

Figure 8.26 shows the time-domain data of the row and the processed results.

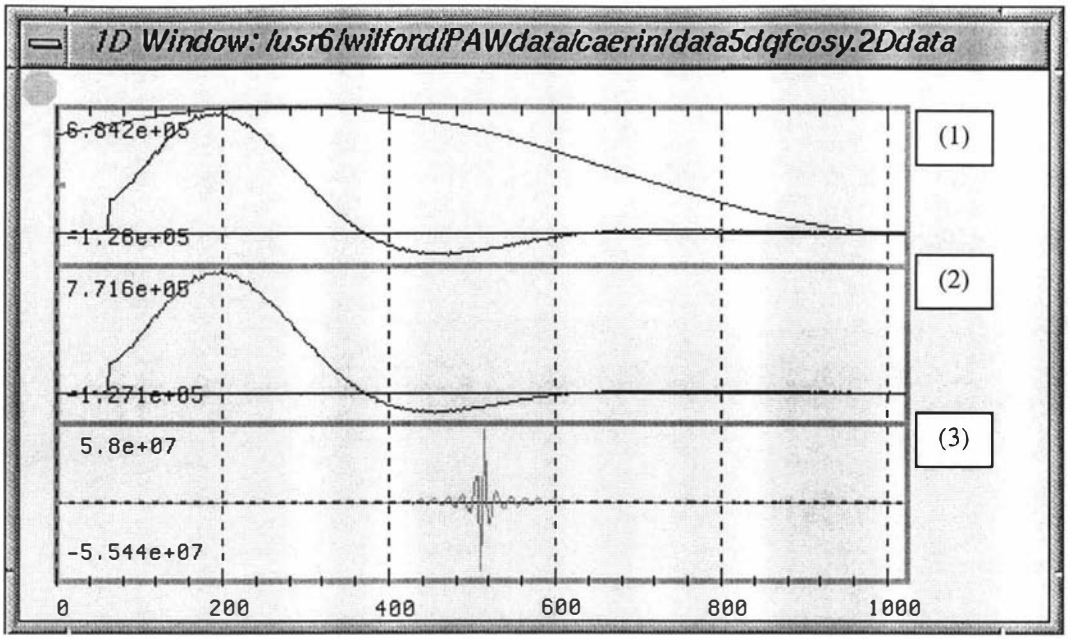


Figure 8.26 Row 100 of the Caerin 4.1 DQF-COSY and two intermediate processed results. Plot 1 is the raw data and the filter. Plot 2 is the filtered data. Plot 3 is the spectrum before phase correction.

Figure 8.27 shows a vertically scaled 1D NMR spectrum of Row 100.

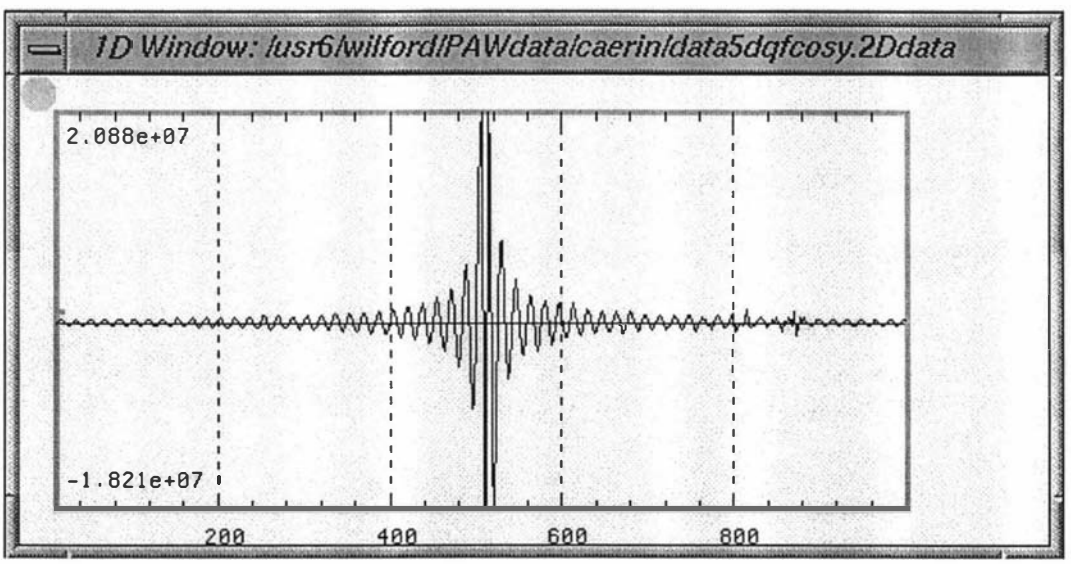
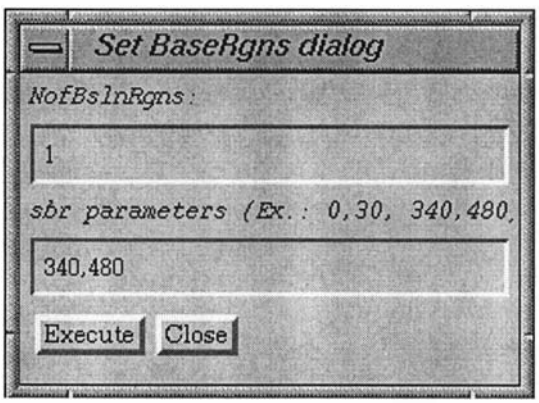


Figure 8.27 The vertically scaled 1D NMR spectrum of Row 100 of the Caerin 4.1 DQF-COSY before phasing. (See Chapter 6 on the baseline correction and Chapter 8 of Volume 1 for some comments.)

The following operations set a base region that is used to define base levels for PAW's auto-phase routines.

- Choose [Phase] in the *1D-processing Toolbox* to open the *Phasing Toolbox*. The popup-message window will re-appear to remind the user to set a base level for the auto-phasing operations.
- Choose [OK] in the popup-message window to close it.
- Choose [ResetParms] in the *Phasing Toolbox* to reset all phasing parameters.
- Type sbs to open the *SetBslnSegm Dialog* (Figure 8.28).



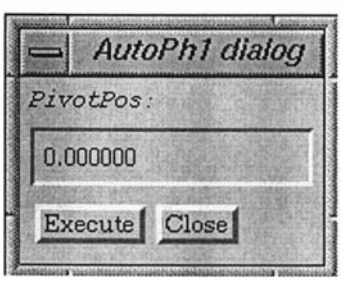
- Enter 1 for the *NofBslnSegms*, and 340,480 for the *sbs parameters*. (This is the largest baseline segment found in the processed 1D and 2D spectra, as described in the last subsection.)
- Choose [Execute].

Figure 8.28 The *SetBslnSegm Dialog*.

An alternative way to set a baseline segment is as follows:

- Choose [Set1DBase] to set a base level for 1D processing. The message "Use button 1 to set a segment for calculating base-level." will appear on the Unix-shell window.
- Select a segment from points 340 to 480 by drawing a rubber rectangle in the 1D draw-window using the click-and-drag technique with MsBtn#1.

The following operations perform an auto-phase correction.



- Choose [AutoPh1(S)] in the *Phasing Toolbox* to open the *AutoPh1 Dialog* (Figure 8.31).
- With the pivot set to 0, choose [Execute] to start the single-loop auto-phase process. (It takes about 100 seconds using a 133 MHz SGI Indy computer.)

Figure 8.29 The AutoPh1 Dialog.

The message "Best Ph0, Ph1 = 110, −22021" are shown in the Unix shell, indicating that the best parameters found are 110° for the zeroth-order phase correction and

−22021° for the first-order phase correction.[1] The resulting spectrum contains isolated in-phase peaks (Figure 8.30), which is what the auto-phase routine is designed to do.

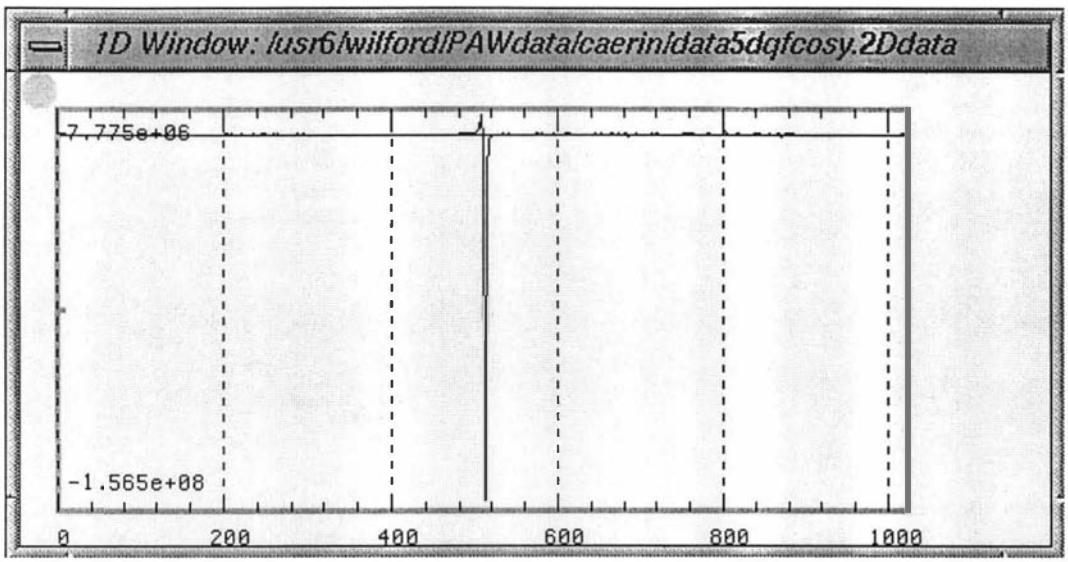


Figure 8.30 The auto-phased result of Row 100 of the Caerin 4.1 DQF-COSY.

An expanded view of the first 200 points shows two isolated peaks centred at around points 150 and 169 (Figure 8.31).

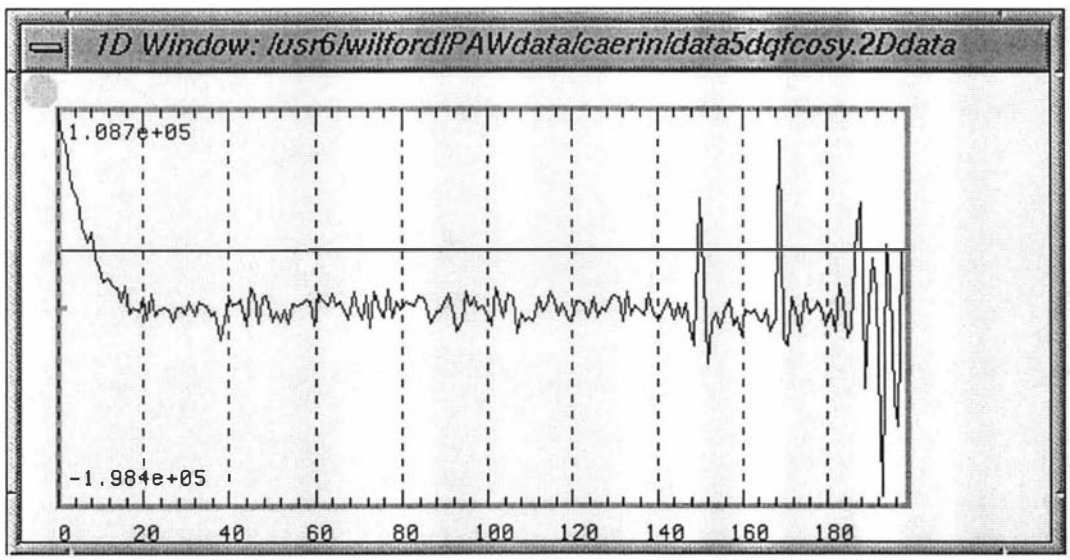


Figure 8.31 An expanded view of the first 200 points of Row 100 after auto-phase correction.

The following operation uses the auto-phase results as starting points to obtaining proper anti-phase correction parameters. The following operations provide two zeroth-order phase values that can both produce anti-phase spectra.

[1] Selecting the multi-loop auto-phase option [SetPh1(M)] gives no significant improvement, with the results being 111.5 and −22024.

- Choose [WrtBuf0→1] in the *1D-MiscProc Toolbox* to save the result in the Buffer 1.
- Zoom the 1D plot into the region between points 0 to 200.
- Choose [ResetParms] and then [SetPh0(S)] in the *Phasing Toolbox* to open the *SetPh0 Dialog*.
- Adjust the slider to phase the peak centred at point 150 to an anti-phase shape. This should either be 66 or −104 degrees, as shown in Plot 1 and 2 of Figure 8.32.

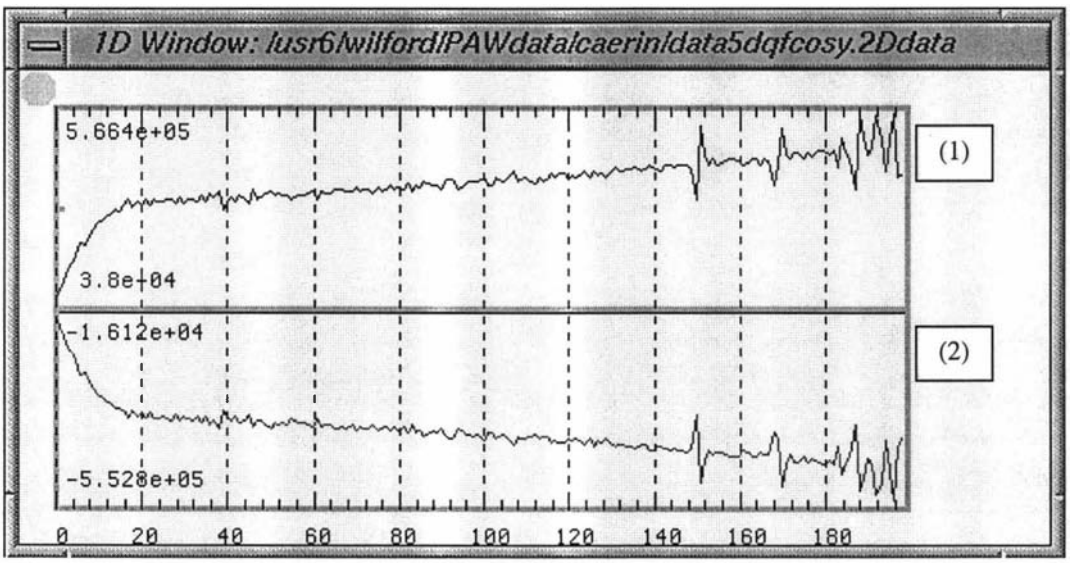


Figure 8.32 Two possible zeroth-order phase corrections for Row 100 of the Caerin 4.1 DQF-COSY. Top: −104-degree correction. Bottom: 66-degree correction.

Therefore, the total Ph0 value from the manual and the auto-phase operation is either +176 or −4 degrees, namely 110+66 or 110−104 degrees.

It is necessary to find a new first-order phase-correction value using one of the total zeroth-phase values. The operations are as follows:

- Re-load Row 100.
- Apply the enhanced sine-bell filter with the four parameters being 1024, 70, 2 and −100.
- Type fft.
- Choose [ResetParms].
- Choose [SetPh0(S)] in the *Phasing Toolbox* to open the *SetPh0 Dialog*.
- Apply +176 degrees of zeroth-order phase correction.
- Choose [SetPivot], and set the pivot to 0.0.
- Choose [SetPh1(L)] in the *Phasing Toolbox* to open the large-range *SetPh1 Dialog*.

- Adjust the slider to produce an anti-phase water peak, preferably with an expanded view. The result will be –21939 degrees, as shown in Figure 8.33. (Here, producing an anti-phase water peak is merely a loose criterion. The relatively small difference from the value obtained previously (i.e., –22021 degrees) is not that critical. Any value in between the two, say, –21972, will produce an equally satisfactory spectrum.)

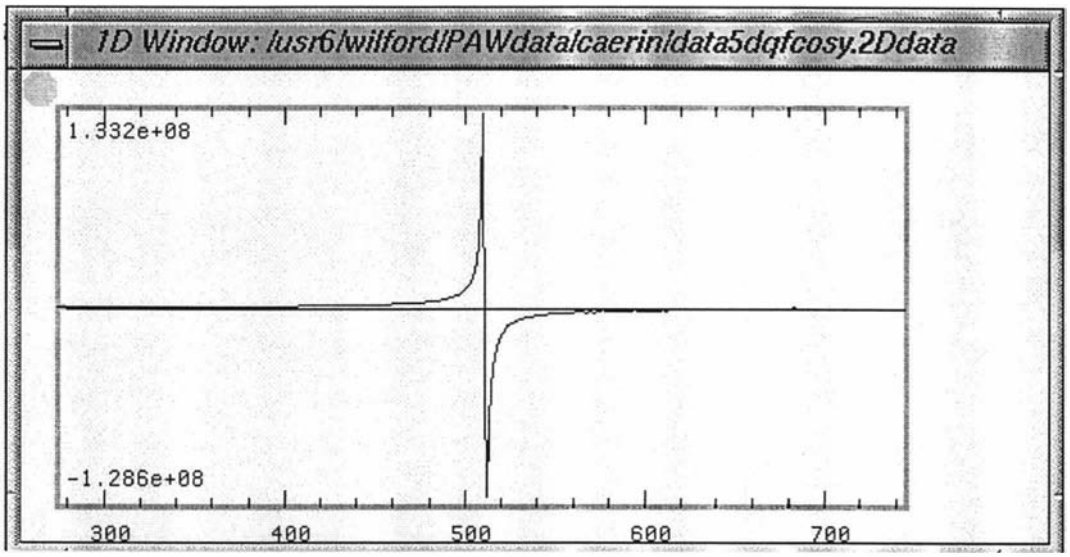


Figure 8.33 An expanded view of the water peak in the final processing result for Row 100.

The final row-processing operations above can be summarised into the following three macro commands:

```
esb (1024,70,2,-100)      # Enhanced sine-bell
fft                       # Fast Fourier transform
ph  (176,-21939,0)        # phase
```

These commands have been embedded into **CaerinDQF_COSY.2Dproc** as follows:

```
# (CaerinDQF_COSY.2Dproc)
MD1  = 1024          # Matrix D1
MD2  = 1024          # Matrix D2

lpCom1    = "n"                          # D1 linear prediction
lpCom2    = "lp (100,412,20,288, r,t)"# D2 linear prediction

fltCom1   = "esb (1024,70,2,-100)"     # D1 filtering
fltCom2   = "esb (800,  0,2,-100)"     # D2 filtering

phCom1    = "ph (-184,-21972,0)"       # D1 phasing
phCom2    = "ph (-88, 0, 0)"           # D2 phasing

bcCom1    = "n"     # D1 baseline correction
bcCom2    = "n"     # D2 baseline correction

DimNoToTrans = b    # Dimension to transform
ExprmtType   = t    # Experiment type

sbs(4, 340,480, 44,140, 900,930, 960,1020) # Set baseline segments
```

Processing the DQF-COSY in D1 with **CaerinDQF_COSY.2Dproc** by setting the *DimNoToTrans* to 1 results in the intermediate data shown in Figure 8.34.

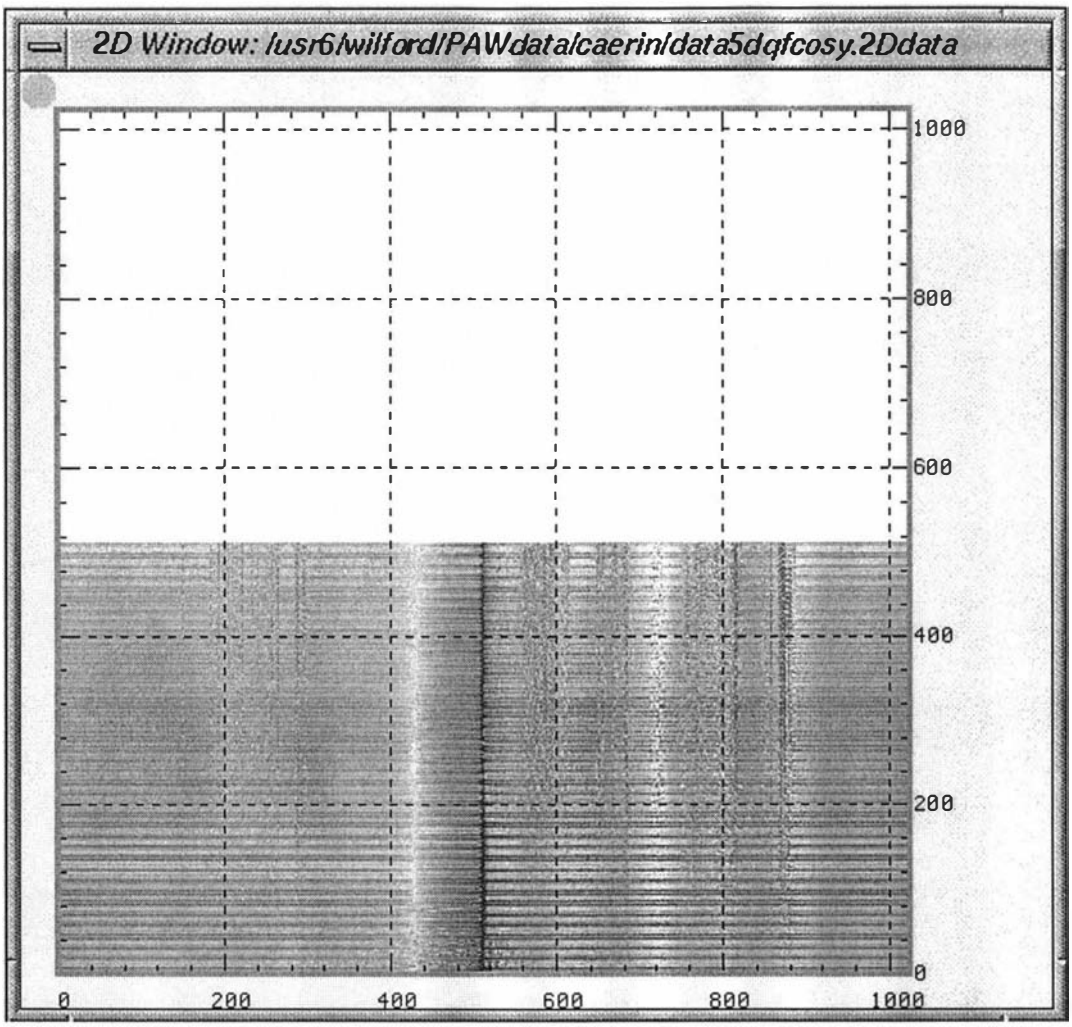


Figure 8.34 The 2D Caerin 4.1 DQF-COSY spectrum after processing in D1.

See also Chapter 8 of Volume I for some comments.

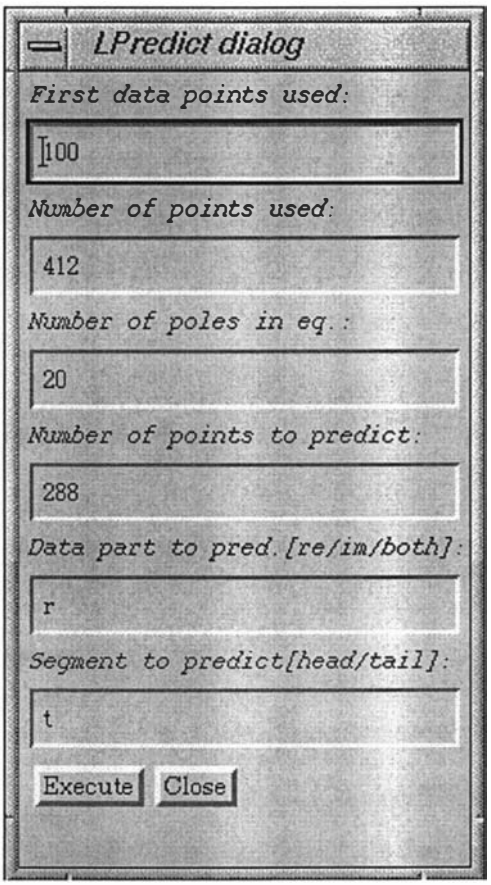
➢ Processing in D2

Processing the DQF-COSY in D2 is similar to that in D1, except for the additional linear-prediction operation.

This subsection describes the high-resolution processing in D2 for the DQF-COSY spectrum. It involves constructing and processing a combined column with the application of linear prediction, an enhanced sine-bell filter, Fourier transforming, and phasing. The process provides the command parameters used in D2-processing, as shown in **CaerinDQF_COSY.2Dproc**. The resulting 2D spectrum is then processed further with a baseline correction macro.

The following operations combine two columns from the intermediate data set and the application of linear prediction.

- Follow the instructions in the last subsection to combine columns 38 and 885. (By default, all results of 1D processing operations are applied to Buffer 0.)
- Choose [LPredict] in the *1D-processing Toolbox* One to open the *LPredict Dialog* (Figure 8.35).



- Fill in the dialog with the data shown in the figure. Then choose [Execute] in the dialog.
- Apply the enhanced sine-bell filter with the four parameters being 800, 0, 2 and −100.
- Type rft.
- Phase the data with ph(−88, 0, 0).

Figure 8.35 The LPredict Dialog.

The LPredict operation used 412 points starting from point 100 to predict 288 points of the real part with a 20^{th}-order linear prediction model. This resulted in a total of 800 significant data points for each column of data. (The reason for predicting 288 points but not 512 was to avoid severe noise induced by incorrect prediction.)

Figure 8.36 shows the results of the above operations.

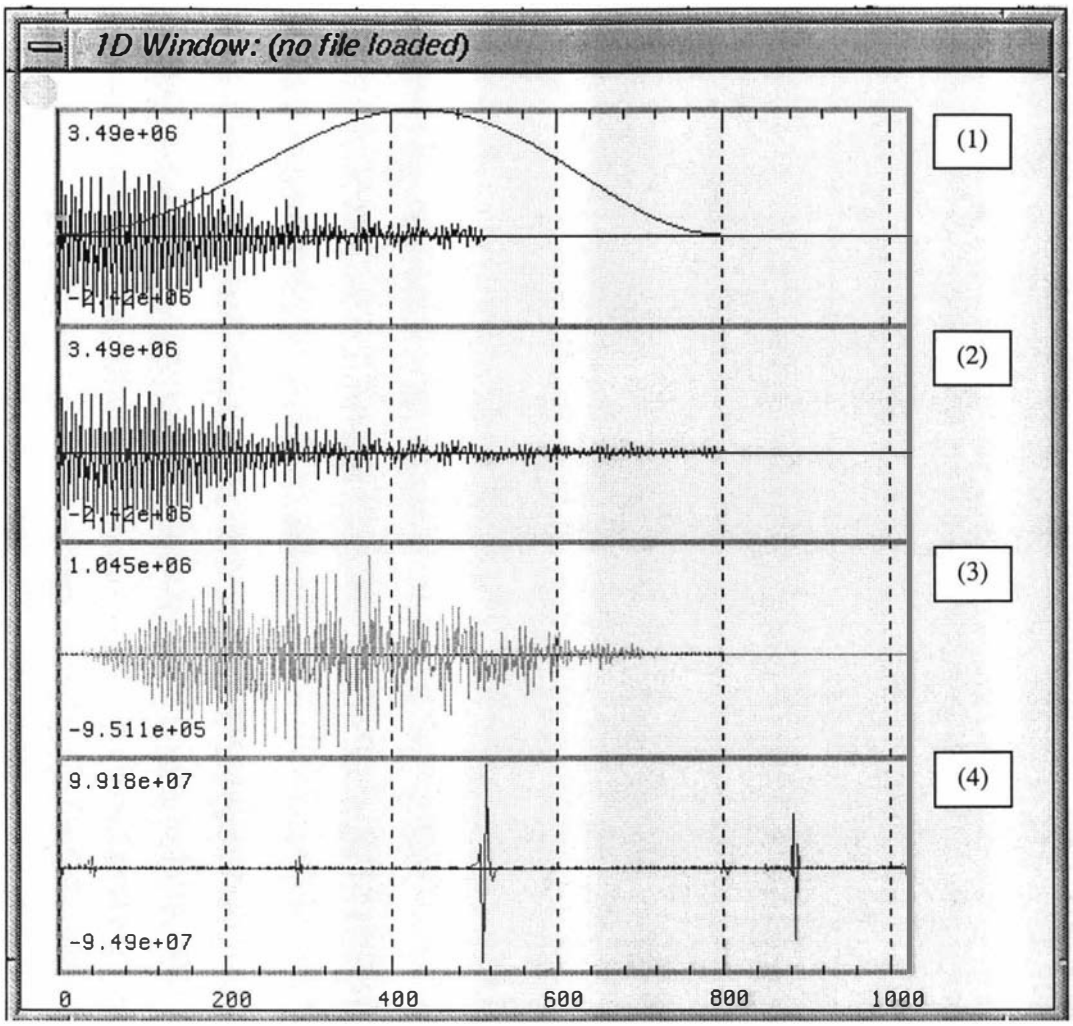


Figure 8.36 The processed results of a combined column of the intermediate data set. Plot 1 is the raw data and the filter. Plot 2 is the predicted data. Plot 3 is the filtered data. Plot 4 is the transformed and phased data.

The parameters obtained from the above operations have been embedded into **CaerinDQF_COSY.2Dproc** (presented in the last subsection), where the t for the ExprmtType defines the use of RFT for column processing.

Loading **CaerinDQF_COSY.2Dproc** for processing in D2 produces a 2D DQF-COSY spectrum, as shown in Figure 8.37.

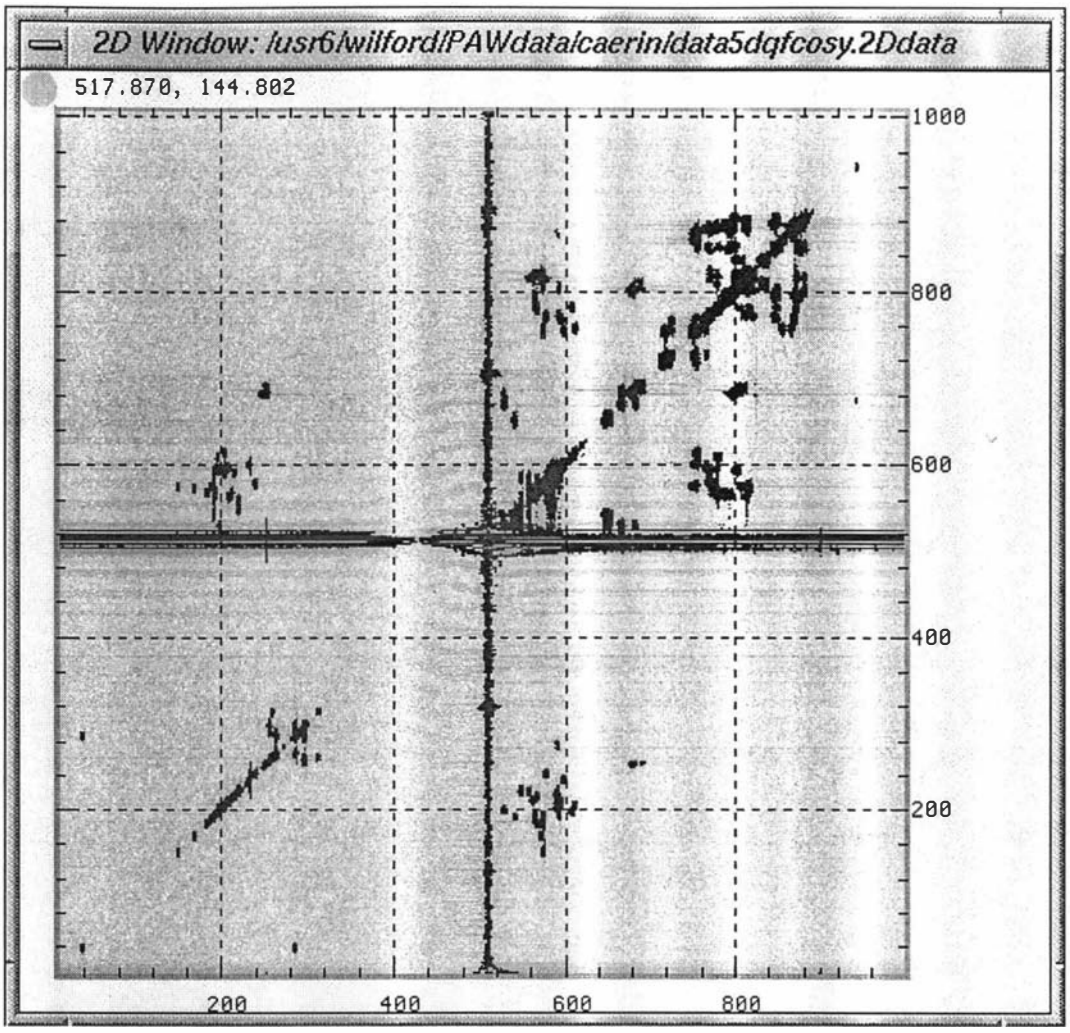


Figure 8.37 A full view of the Caerin 4.1 DQF-COSY spectrum after D2 -processing with high-resolution method.

Ignore the five Bad-LP error messages reported on the Unix shell, because none of these five columns has caused any serious problem. This can be checked by inspecting the small 2D regions containing them.

The baseline correction macro below is written to suppress horizontal streaks in the spectrum. It is in the macro directory, and can be run by typing run. (See chapter 4.)

```
# (CorrectBaseline.mcr)
D1    = 1024                # Dimension 2
D2    = 1024                # Dimension 1
D1Struct  = r               # D1 structure
D2Struct  = r               # D2 structure
DrawOn    = F               # Not to draw
sbs (4, 340,480, 44,140, 900,936, 960,1024) #Set baseline segments
for (%I=0, 1023, 1)         # Start the for-loop
  ldr (%I, r)               # Load row %I as real data
  bc (s, 5)                 # Correct baseline
  wrr (%I, r)               # Save row %I as real data
next                  # Continue the for-loop
DrawOn = T                  # Draw now
dr1                   # Draw the 1D plot
```

```
dr2                # Draw the 2d plot
```

Figure 8.38 shows the DQF-COSY spectrum after baseline correction.

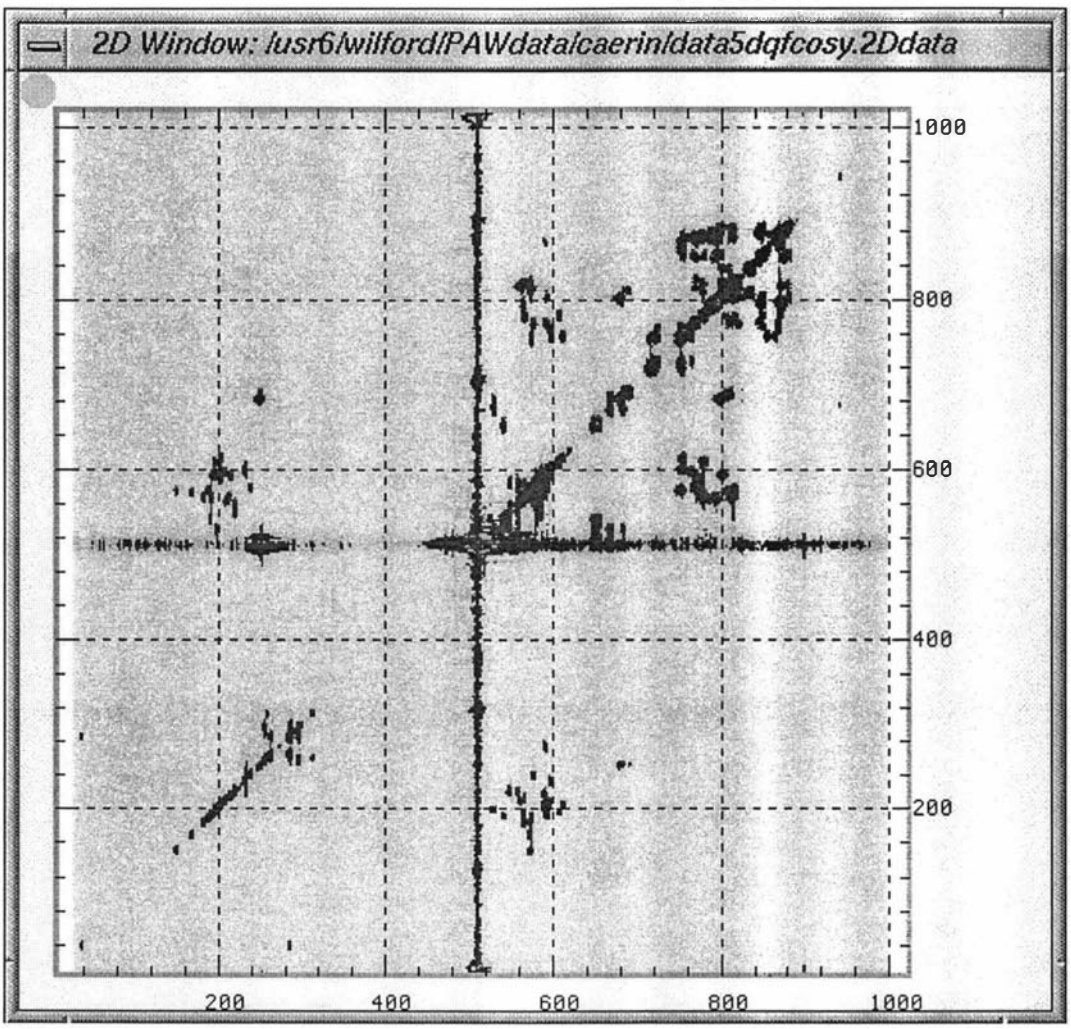


Figure 8.38 The Caerin 4.1 DQF-COSY spectrum after baseline correction.

The expanded views of the significant regions can be found in Chapter 8, Volume I.

Note that, to show the intensity background, all the contour plots above are deliberately presented with a relatively low first level of 25 in PAW's threshold units for 2D contour and intensity plots. Figure 8.39 shows a cleaner plot produced by setting the *First-level Threshold* for the contour to 30. (See Chapter 7 for the operations required.)

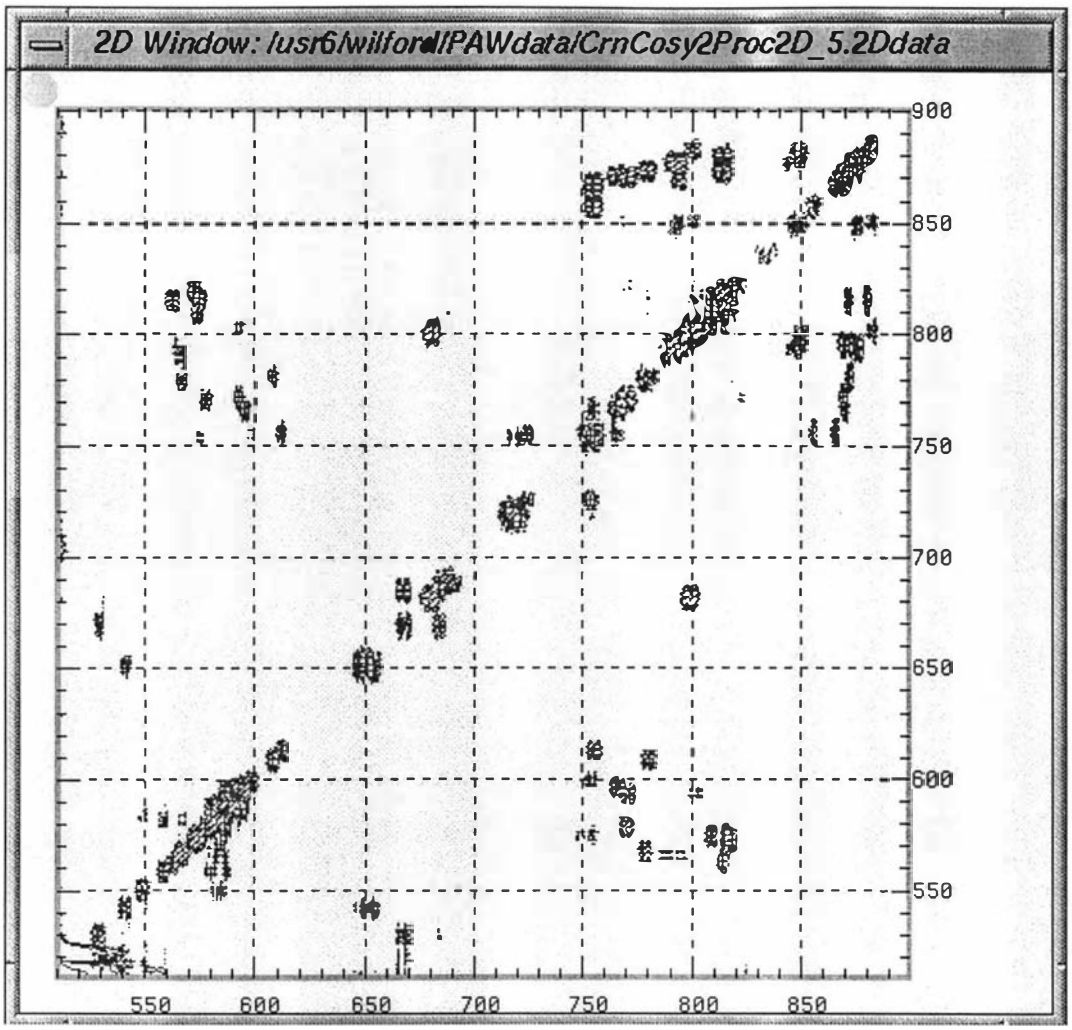


Figure 8.39 A cleaner plot of the upper-right region (aliphatic) of the processed Caerin 4.1 DQF-COSY spectrum.

➢ Spectral calibration

The following operations calibrate the 2D spectrum with the water peak centred at (510.3, 511.5)[2].

- Type zs to zoom into a small region containing the water peak (Figure 8.40).

[2] In points.

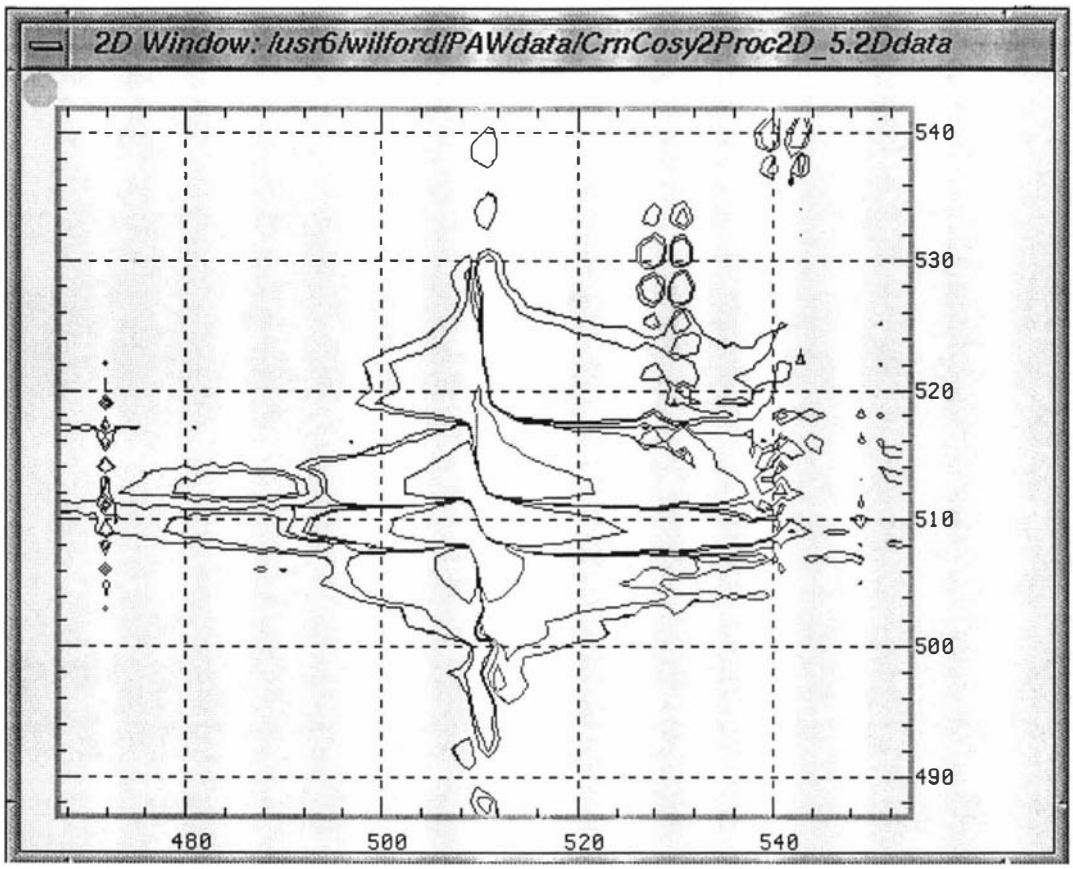


Figure 8.40 The contour plot of the small region containing the water peak.

- Select ppm from the *ComDspUnit Radio-box* of the *Common-display Toolbox* (See Chapter 7).
- Choose [Calibration] from the *Display Menu* in the *Menu-bar* to open the *Spectrum Calibration Dialog* (Figure 8.41).

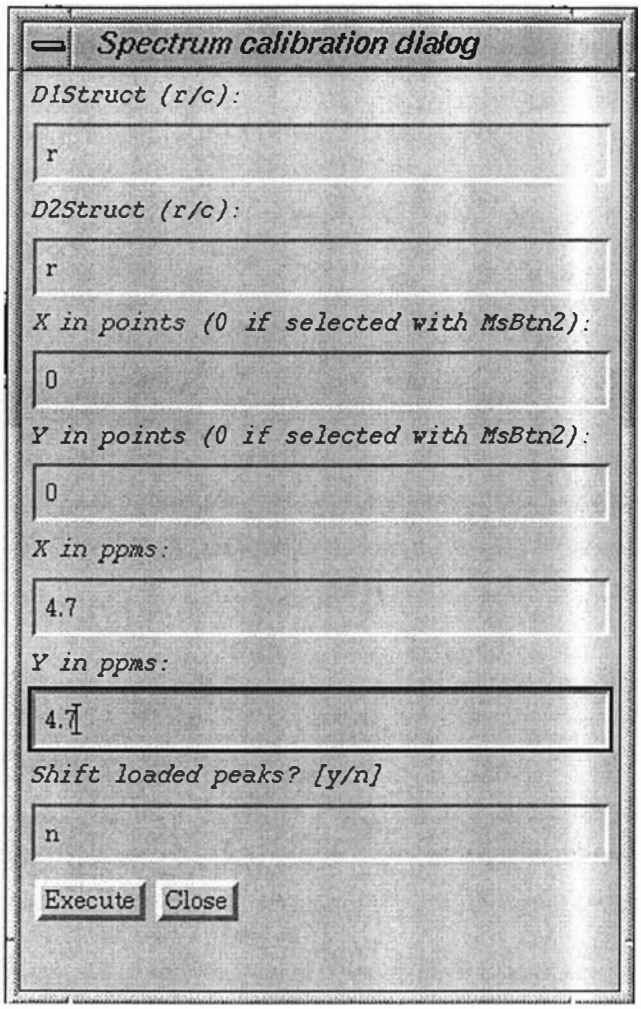


Figure 8.41 The Spectrum Calibration Dialog.

- Fill in the dialog with the data shown in the figure.
- Select the water peak centre by clicking and dragging a crosshair using MsBtn#2 (Figure 8.42).

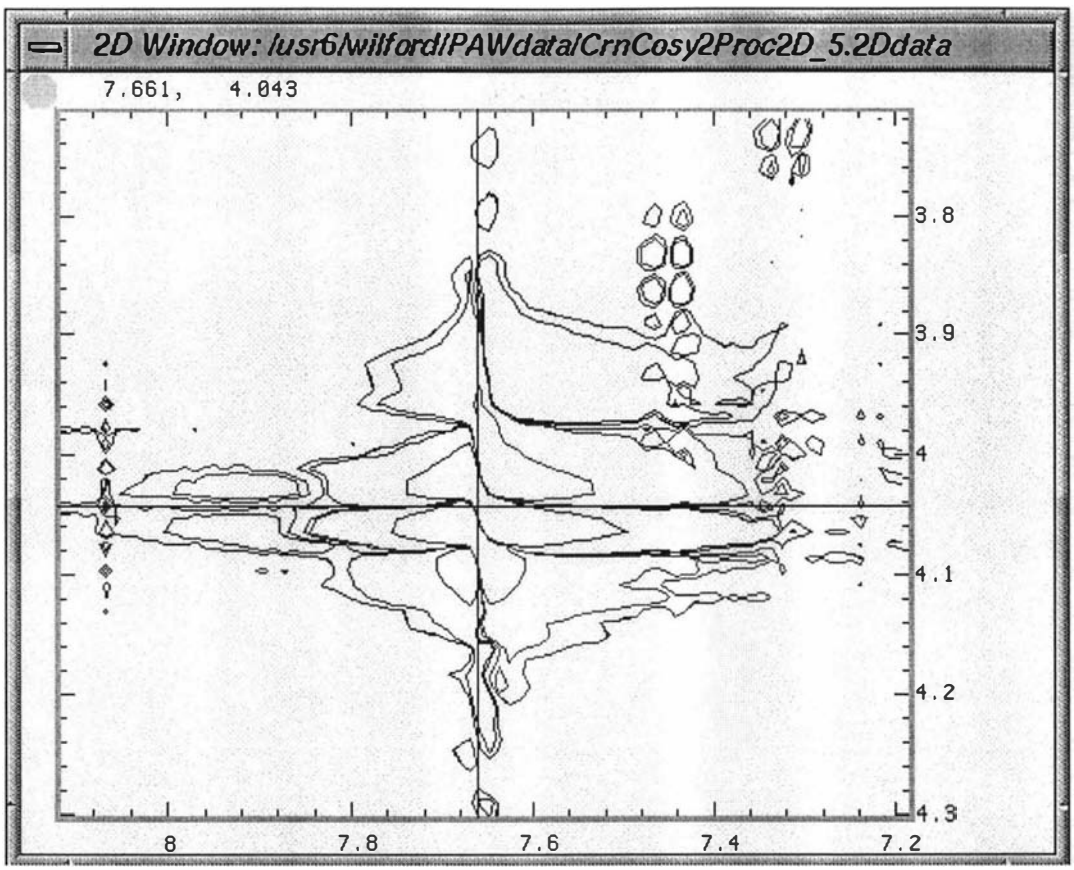


Figure 8.42 Referencing the water-peak centre by clicking and dragging a crosshair using MsBtn#2.

- Choose [Execute] in the dialog. The plot will then be re-drawn with the calibrated scales shown on the axes. (Figure 8.43).

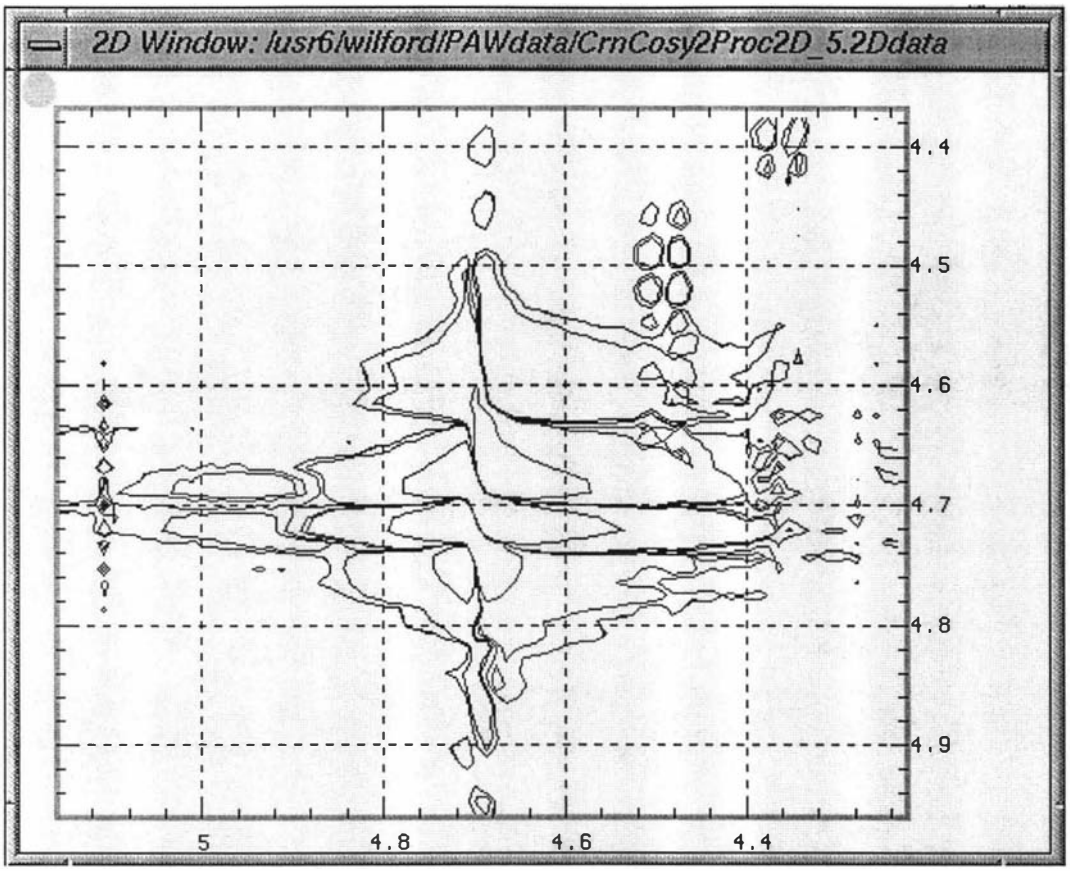


Figure 8.43 The calibrated plot of the water peak.

8.3 Processing the Caerin 4.1 TOCSY and NOESY NMR Data

The operations to process the Caerin 4.1 TOCSY and NOESY NMR data are similar to those presented for the high-resolution processing method for the DQF-COSY, apart from the few differences described below.

1) The NMR data files involved are **data6tocsy070.2Ddata** and **data7noesy150.2Ddata**, where the numbers 070 and 150 refer to the mixing times used in the experiments. The time-domain data will be named the Caerin 4.1 TOCSY070 and NOESY150. The corresponding frequency-domain data set will be called the Caerin 4.1 TOCSY070 or NOESY150 spectrum.

2) The auto-phase operations are largely simplified in comparison with those for the DQF-COSY NMR data, because there is no need to search for anti-phase parameters. Only minor adjustments are required to improve the auto-phase results, and the criteria are the refined shapes of the small 1D peak centred at about point 38.

3) The macro for processing the Caerin 4.1 TOCSY070 NMR data is as follows:

```
# (data6tocsy070.2Dproc)
```

```
MD1  = 1024          # Matrix D1
MD2  = 1024          # Matrix D2

lpCom1    = "n"                  # D1 linear prediction
lpCom2    = "lp (100,412,20,288, r,t )" # D2 linear prediction

fltCom1   = "esb (1024, 80, 2, -100)"    # D1 filtering
fltCom2   = "esb (800,  80, 2, 0)"       # D2 filtering

phCom1    = "ph (16, -21988, 0)"  # D1 phasing
phCom2    = "ph (64, 62, 0)"      # D2 phasing

bcCom1    = "n"     # D1 baseline correction
bcCom2    = "n"     # D2 baseline correction

DimNoToTrans = b    # Dimension to transform
ExprmtType   = t    # Experiment type

sbs(4, 340,480, 44,140, 900,930, 960,1020) # Set baseline segments
```

4) The macro for processing the Caerin 4.1 NOESY150 NMR data is as follows:

```
# (data7noesy150.2Dproc)
MD1  = 1024          # Matrix D1
MD2  = 1024          # Matrix D2

lpCom1    = "n"                  # D1 linear prediction
lpCom2    = "lp (100,412,20,288, r,t )" # D2 linear prediction

fltCom1   = "esb (1024, 80, 2, -100)"  # D1 filtering
fltCom2   = "esb (800,  80, 2, 0)"     # D2 filtering

phCom1    = "ph (-78, -21978, 0)"      # D1 phasing
phCom2    = "ph (-12, 27, 37.8)"  # D2 phasing

bcCom1    = "n"          # D1 baseline correction
bcCom2    = "n"          # D2 baseline correction

DimNoToTrans = b         # Dimension to transform
ExprmtType   = t         # Experiment type

sbs(4, 340,480, 44,140, 900,930, 960,1020) # Set baseline segments
```

5) The baseline correction macro that corrects the vertical and horizontal streaks in both 2D spectra is as follows:

```
# (CorrectBaseline.mcr)
D1   = 1024          # Dimension 2
D2   = 1024          # Dimension 1
D1Struct  = r        # D1 structure
D2Struct  = r        # D2 structure
DrawOn    = F        # Not to draw
sbs (4, 340,480, 44,140, 900,936, 960,1024)# Set baseline segments
for (%I=0, 1023, 1)      # Start the for-loop
  ldc (%I, r)        # Load column %I as real data
  bc (s, 5)          # Correct baseline
  wrc (%I, r)        # Save column %I as real data
next           # Continue the for-loop
for (%I=0, 1023, 1)      # Start the for-loop
  ldr (%I, r)        # Load row %I as real data
  bc (s, 5)          # Correct baseline
```

```
  wrr (%I, r)       # Save row %I as real data
next              # Continue the for-loop
DrawOn = T            # Draw now
dr1               # Draw the 1D plot
dr2               # Draw the 2d plot
```

Both macros can be found in the data directory.

Chapter 9:

Peak-picking Process for the Caerin 4.1 NMR Spectra

9.1 Introduction

This chapter describes the peak-picking operations to create the peak lists used for the NMR spectral assignment of Caerin 4.1. All the three high-resolution spectra obtained in Chapter 8 were used during the peak picking.

➢ The *Peak-picking Toolbox*

Figure 9.1 The Peak-display Toolbox.

The command buttons that are related to peak-picking operations are collected into the *Peak-picking Toolbox*, as shown in Figure 9.1.

To open the toolbox, either type opp or choose [PkPickTBox] from the *Process Menu.*

To close the toolbox, either type xpp or double-click on the window-control button at its top-left corner.

To avoid unexpected errors, it is recommended that PAW is restarted and the workbench used in Chapter 7 is again loaded.

9.2 Preliminary

- Start PAW.
- Choose [MyWorkbench] from the *Window Menu.* (See Chapter 7.)
- In the *General-display Toolbox*, turn on the [ppm] and [DspInLog] buttons.

➢ **Loading spectra for peak-picking**

- Load the spectra **CaerinNoesy150C**, **CaerinTocsy150C** and **CaerinCosyC** as described in Chapter 7.
- Change to *Colour Scheme 2* as described in Chapter 7.

Figure 9.2 shows the **CaerinNoesy150C** spectrum displayed in *Colour Scheme 2*.

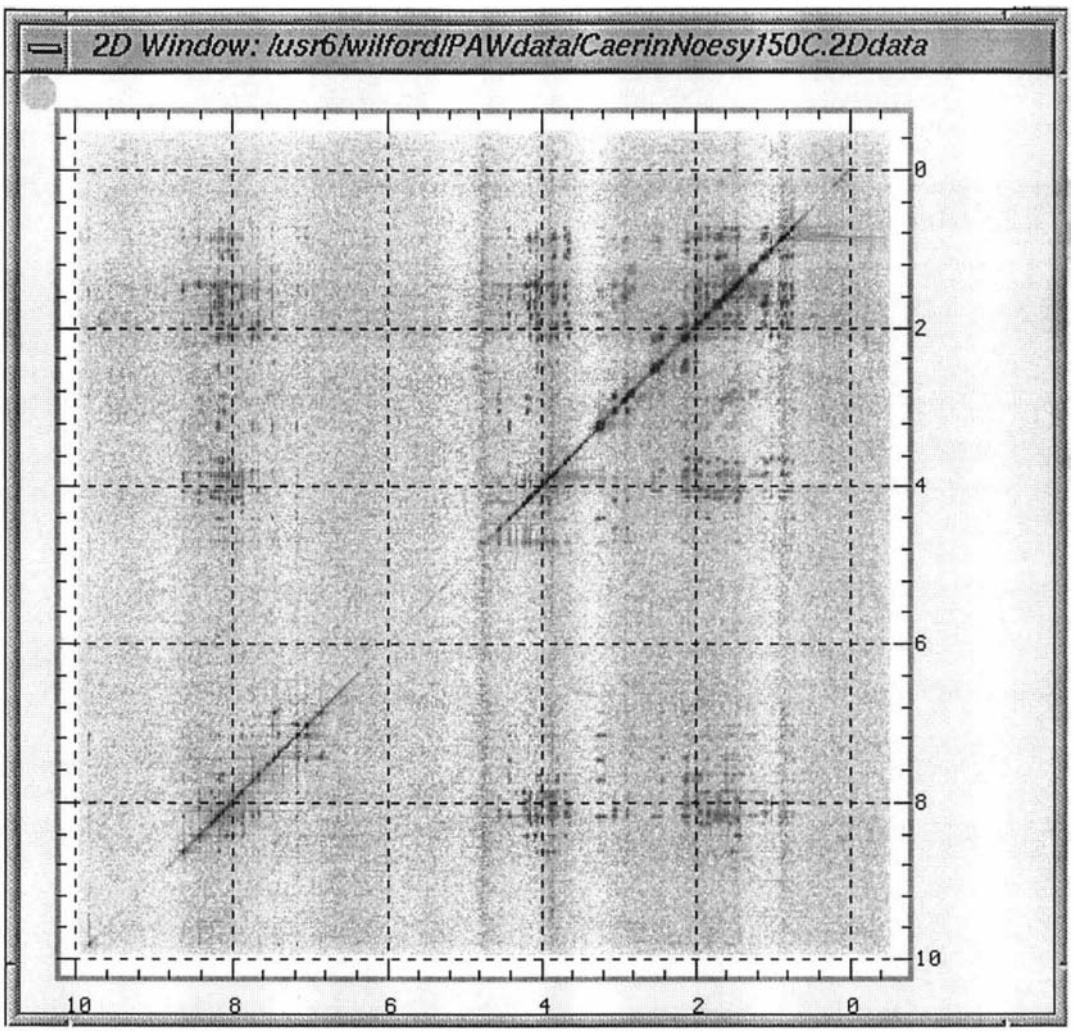


Figure 9.2 The CaerinNoesy150C spectrum displayed in *Colour Scheme 2*.

➢ Defining contour parameters for peak picking

The following operations turn on the intensity-and-contour display mode, and define proper contour parameters for displaying the **CaerinNoesy150C** spectrum.

- Click in the window for the **CaerinNoesy150C** spectrum.
- Turn on [Intn.+Contour] in the *2D-display Toolbox*.
- Choose [DefContrs] in the *2D-display Toolbox* to open a *Contour-plot Dialog* and set the *First-level Threshold* to 28, as described in Chapter 7.
- Repeat the same operations for the **CaerinTocsy070C** spectrum.
- Repeat the same operations for the **CaerinCosyC** spectrum, with the *First-level Threshold* set to 30 and the entry for the Sign set to b.

➢ Defining numbers to zoomed-patterns for peak picking

To simplify the operations and description for the picking operations, seven zoomed patterns were defined as follows.

- Click in the window for the **CaerinNoesy150C** spectrum.
- Zoom into the multi-region pattern shown in Figure 9.3, as described in Chapter7.

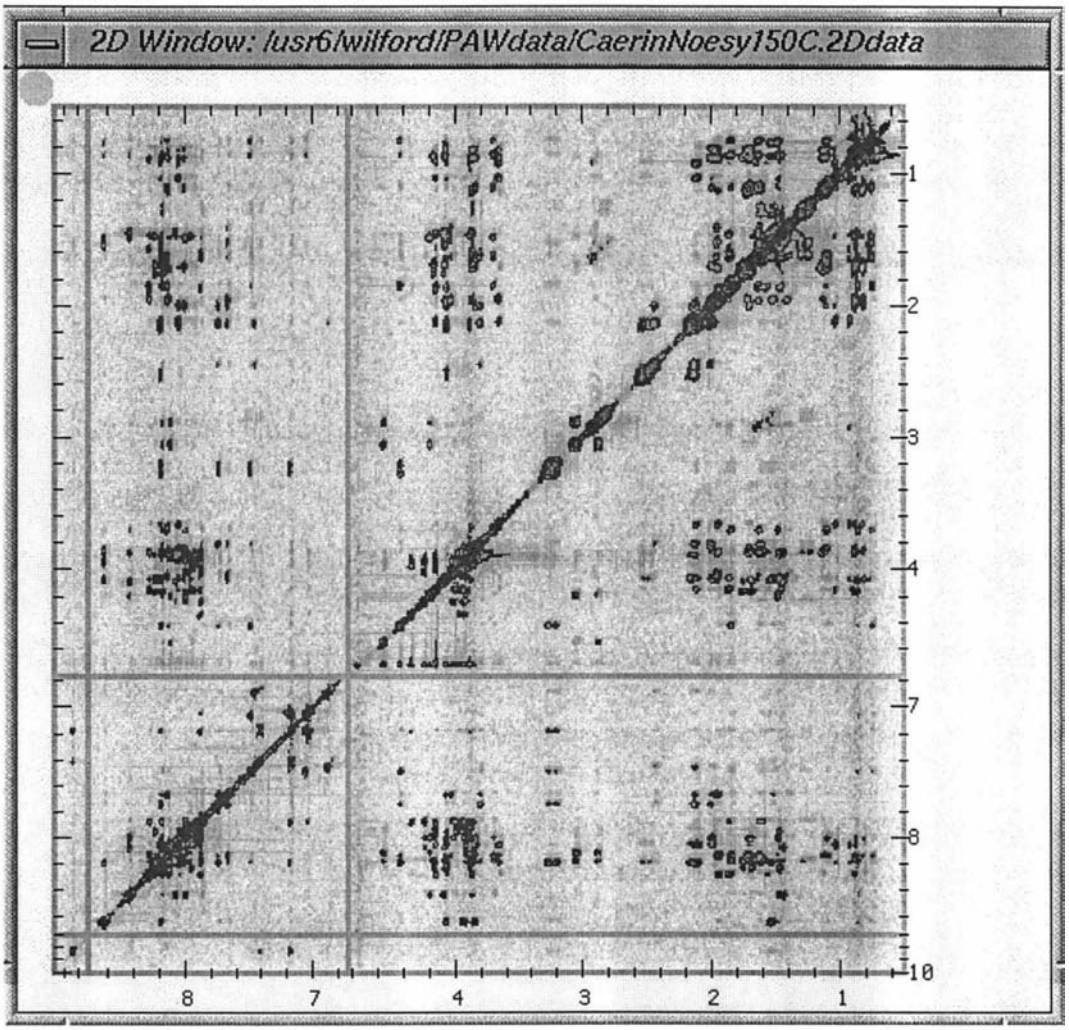


Figure 9.3 The multi-region display used in peak picking.

- Use the dz# command to define this multi-region pattern as Zm#8, as described in Chapter 7.
- Use the zs command to zoom into six other regions as shown in Figure 9.4, and define them as z1 to z6.

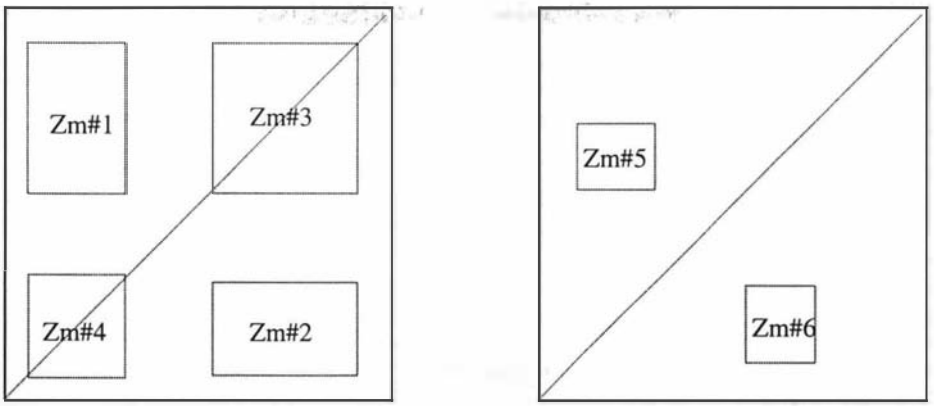


Figure 9.4 The six zooming patterns used in peak picking.

These definitions allow the commands z1, z2, ... z6 and z8 to be used in the following sections.

9.3 Picking Raw Peaks

➢ **Picking raw peaks in the upper-left region (Zm#1)**

- Choose [Set2DBsLv] in the *Peak-picking Toolbox*. The message "*Use MsBtn#1 to set a region for calculating 2DBaseLev*." will appear above the plot.
- Use the left mouse-button to select a small region that encloses a small peak at around [6.828,2.078], which is the strongest peak on the right. The base-level parameters, of which the Mean+SD is around 1.08×10^6, will be calculated and displayed on the Unix shell.
- Choose [PickRawPeaks] in the *Peak-picking Toolbox*. The message '*Use MsBtn#1 to set a region for picking raw peaks*.' will be displayed on top of the draw-window.
- Select the entire zoomed region. The *Peak-picking dialog* (Figure 9.5) will appear. Retain all default values for the entries and make sure the BaseLev is set to 0 so that the calculated base level will be used.

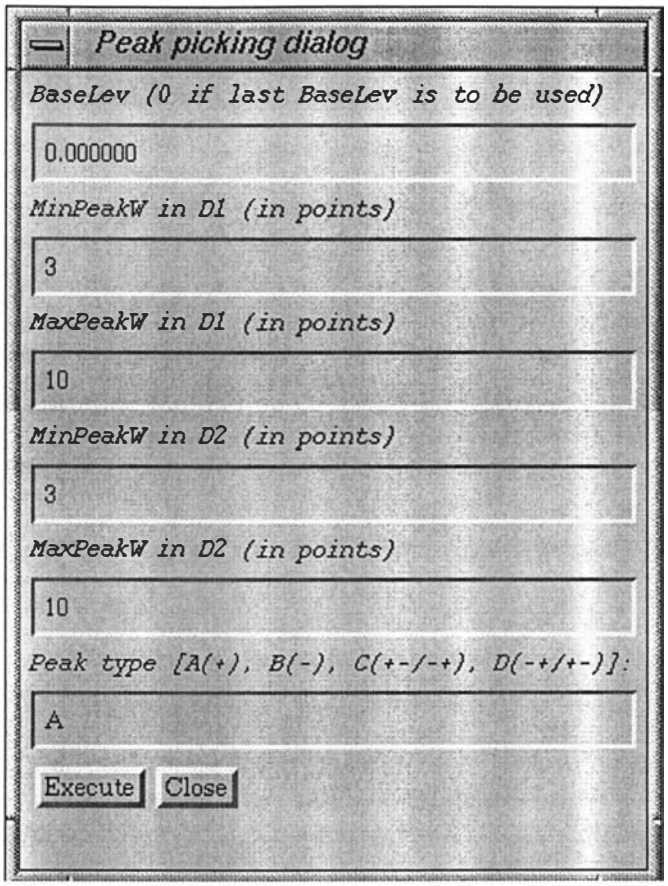


Figure 9.5 The *peak-picking Dialog* for picking raw-peaks.

- Choose [Execute]. The picked peaks will then be displayed on the plot, as shown in Figure 9.6.

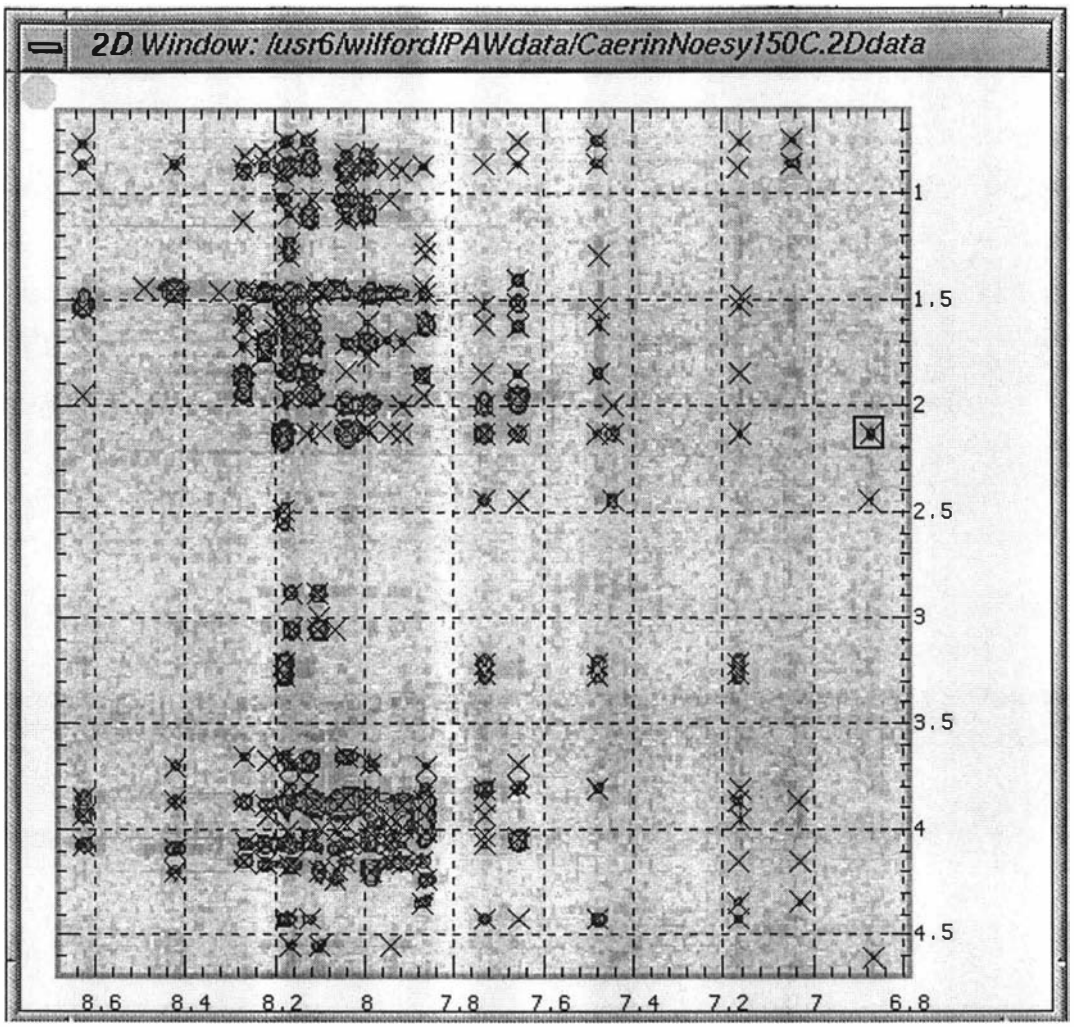


Figure 9.6 The raw peaks in the upper-left region of the **CaerinNoesy150C**. They are picked by setting a base level calculated from a small region at around [6.828,2.078].

- If the result is not satisfactory, remove all peaks in the region by choosing [RmveRPks] in the *peak-picking Toolbox* and then selecting the entire region. After removing the peaks in the region, repeat the procedure above to reset a base level and pick another lot of raw peaks.

Note:

- To preserve potentially useful information in the spectral assignment process, it is often better to keep more raw peaks in the list than less.

- Peaks symbols are only temporarily plotted on top of a spectrum in order to have the flexibility of displaying peak symbols onto any spectrum, including the DQF-COSY or TOCSY. When the X Window manager refreshes a window after any window-expose event, the peak symbols disappear, allowing a quick display of a clean spectrum without having to switch off the peak display. To redisplay the peaks, simply click in the draw-window with MsBtn#3.

➢ Saving the raw-peak list

- Choose [SavePkLists] in the *Peak-assignment Toolbox*. A small window (Figure 9.7) with the question '*Save the peak list?*' will popup to ensure that existing peak lists will not be overwritten by mistake.

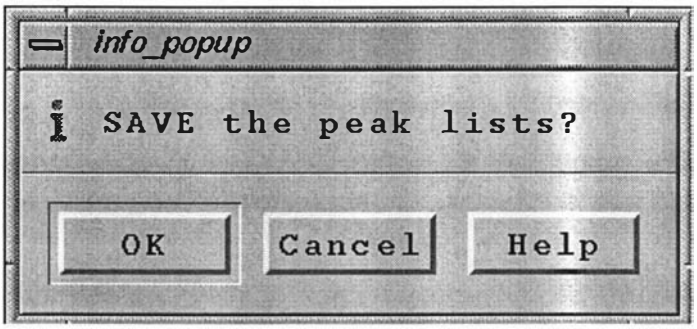


Figure 9.7 The popup message for saving peak lists.

- Choose [OK] and you see a *file-selection dialog* as follows:

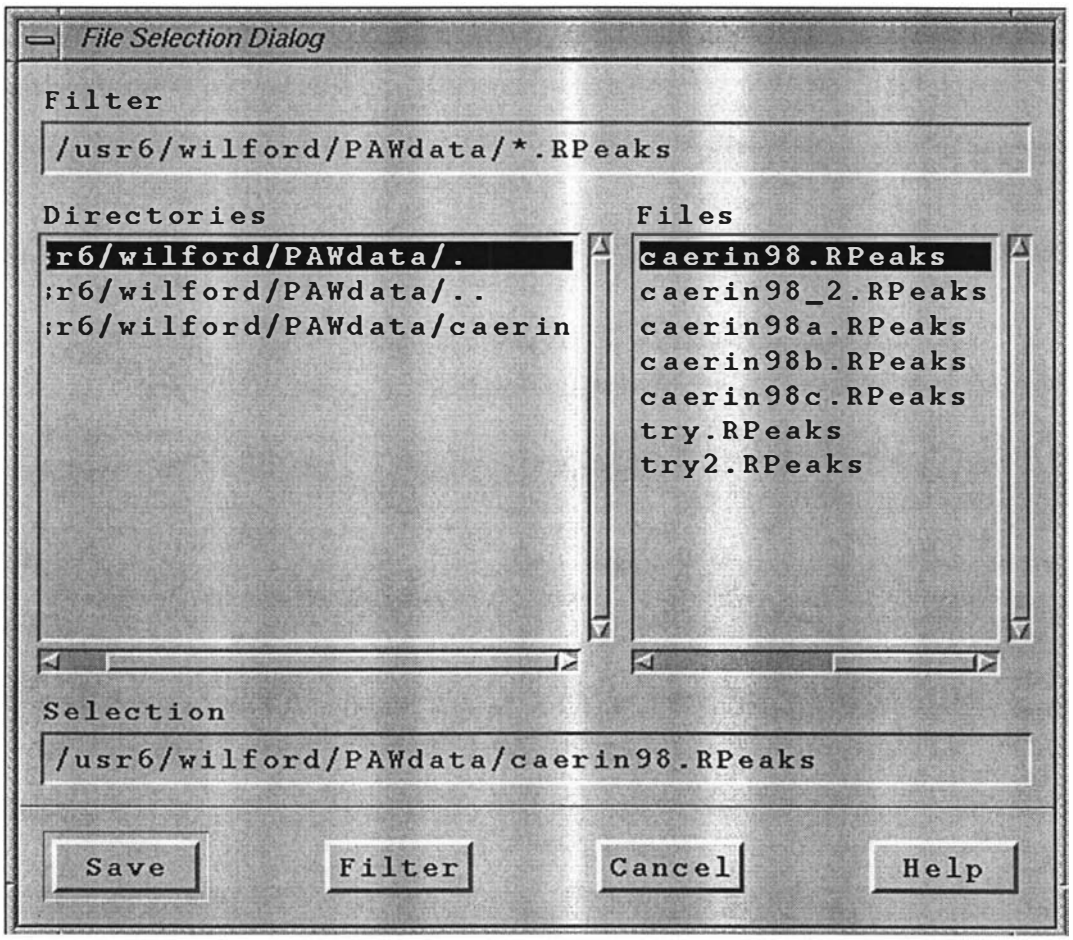


Figure 9.8 The file-selection window for saving peak lists.

- Choose **Caerin98.RPeaks** and change it to **Caerin99.RPeaks** or similar to avoid overwriting the list **Caerin98.RPeaks**.
- Choose [Save] in the dialog.

Note: that when saving peaks, PAW always save all peaks in one operation, including all diagonal peaks and cross peaks, if any. Although only raw peaks were picked so far, this operation also created two other empty files for the diagonal peaks and cross peaks.

➢ Picking raw peaks in the transposed region (Zm#2)

Because the base level is about the same as that set for Zm#1, there is no need to reset the base level for picking peaks in this region. The operation can be as simple as follows:

- Choose [PickRawPeaks] in the *Peak-picking Toolbox* to open the *Peak-picking dialog*. Select the entire region and keep the BaseLev displayed in the entry box. Choose [Execute]. The display will be refreshed with the newly picked peaks as shown in Figure 9.9.

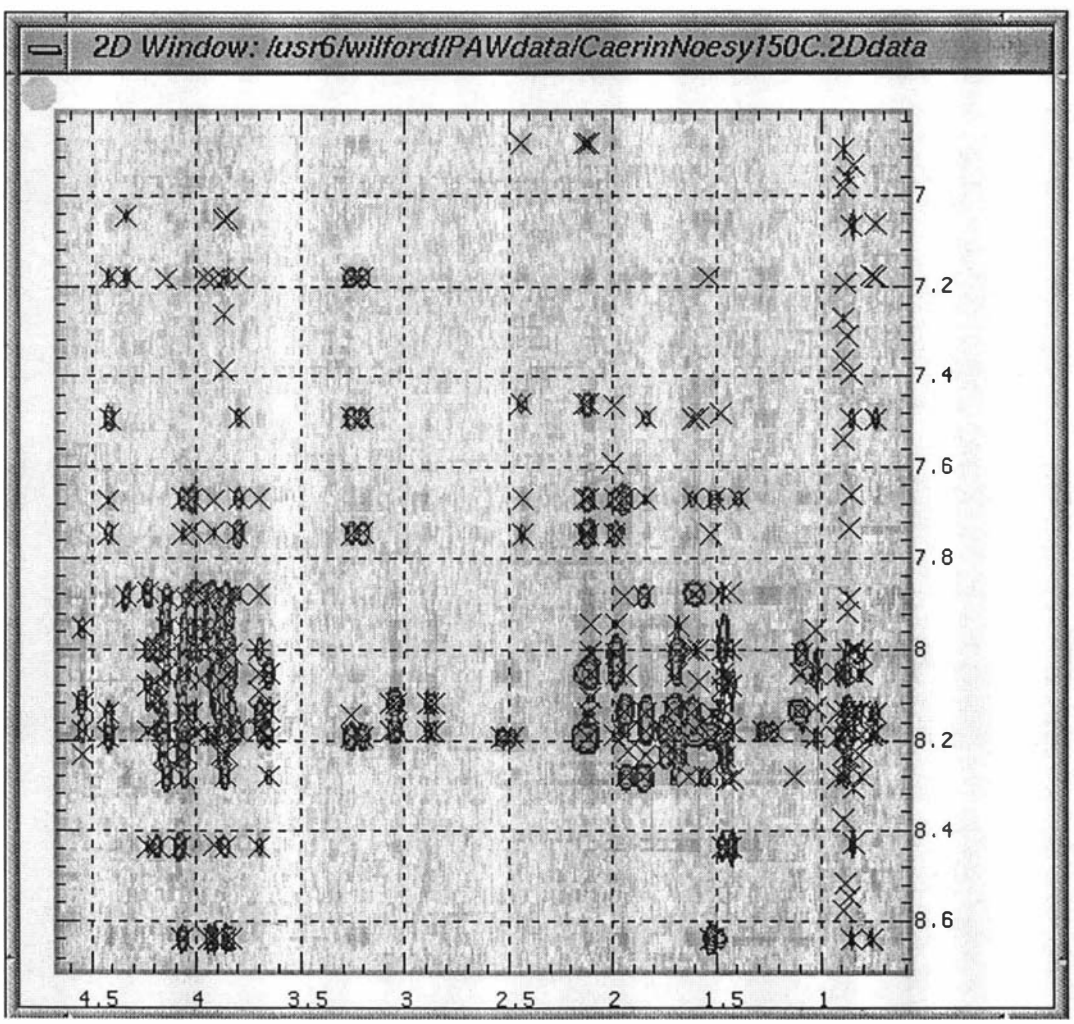


Figure 9.9 The raw peaks at the lower-right region of the **CaerinNoesy150C**. They are picked with the same base level calculated for the upper-left region.

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

➢ Picking raw peaks in the upper-right region (Zm#3)

Because of the higher level of noise, picking peaks using the base level chosen for regions Zm#1 and Zm#2 will result in too many peaks being picked in the upper-right region. A different base level must therefore be set, preferably with the first-level threshold for the contour plot set to 28.

- Choose [Set2DBsLv] in the *Peak-picking Toolbox.*
- Use the left mouse-button to select a small region that encloses a peak at around [4.368,0.696], which is the most intense peak at the top-left corner. The base-level parameters, of which the Mean+SD is around 3.66×10^6, will be calculated and displayed on the Unix shell.
- Choose [PickRawPeaks] in the *Peak-picking Toolbox.*
- Select the entire zoomed region. The *Peak-picking dialog* will appear. Retain all the default values in the entry boxes and make sure the entry for the base level is 0 so that the value calculated previously will be used.
- Choose [Execute]. The picked peaks will be displayed on the plot as shown in Figure 9.10.

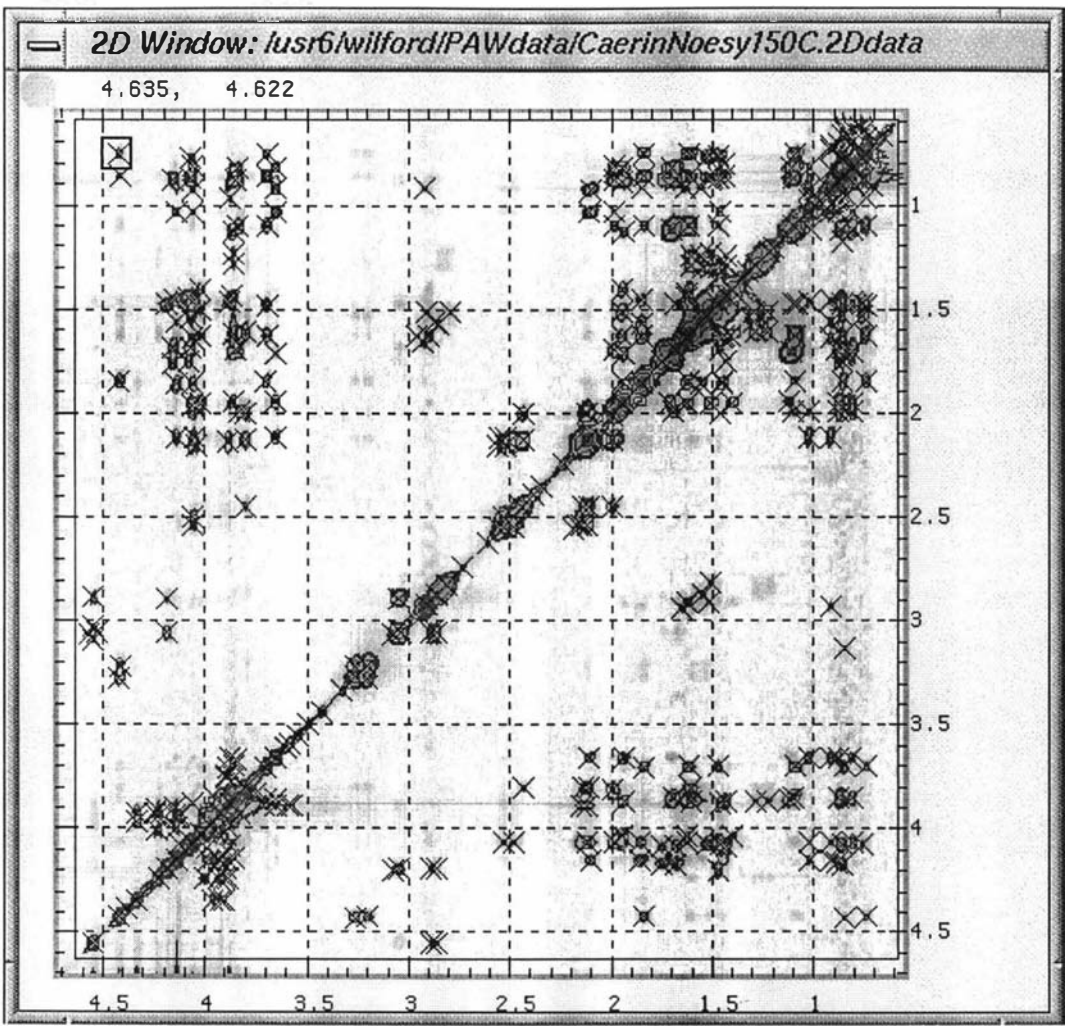


Figure 9.10 The raw peaks at the upper-right region of the **CaerinNoesy150C** spectrum. They are picked by setting a base level calculated from a small region at around [4.368,0.696].

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

Note:

- After the X Window manager will not be re-displayed the peaks symbols after removing the file-selection dialog. To re-display the symbols, click in the window with MsBtn#3.
- It can be seen that some weak peaks were not picked with this base level. To avoid complication, these peaks can be considered at a later stage of the spectral assignment.

➢ Picking raw peaks in the lower-left region (Zm#4)

- Type z4.
- Choose [Set2DBsLv] in the *Peak-picking Toolbox*.
- Use the left mouse-button to select a small region that encloses a peak at around [8.134,7.124], as shown by the rectangle in Figure 9.11. The base-level parameters will be calculated and displayed on the Unix shell. (The Mean+SD is around 2.17×10^6).

- Choose [PickRawPeaks] in the *Peak-picking Toolbox*.
- Select the entire region to pick the raw peaks. The *Peak-picking dialog* will appear. Retain all the default values in the entry boxes and make sure the base level is 0 so that the value calculated previously will be used.
- Choose [Execute]. The picked peaks will be displayed on the plot, as shown in Figure 9.11.

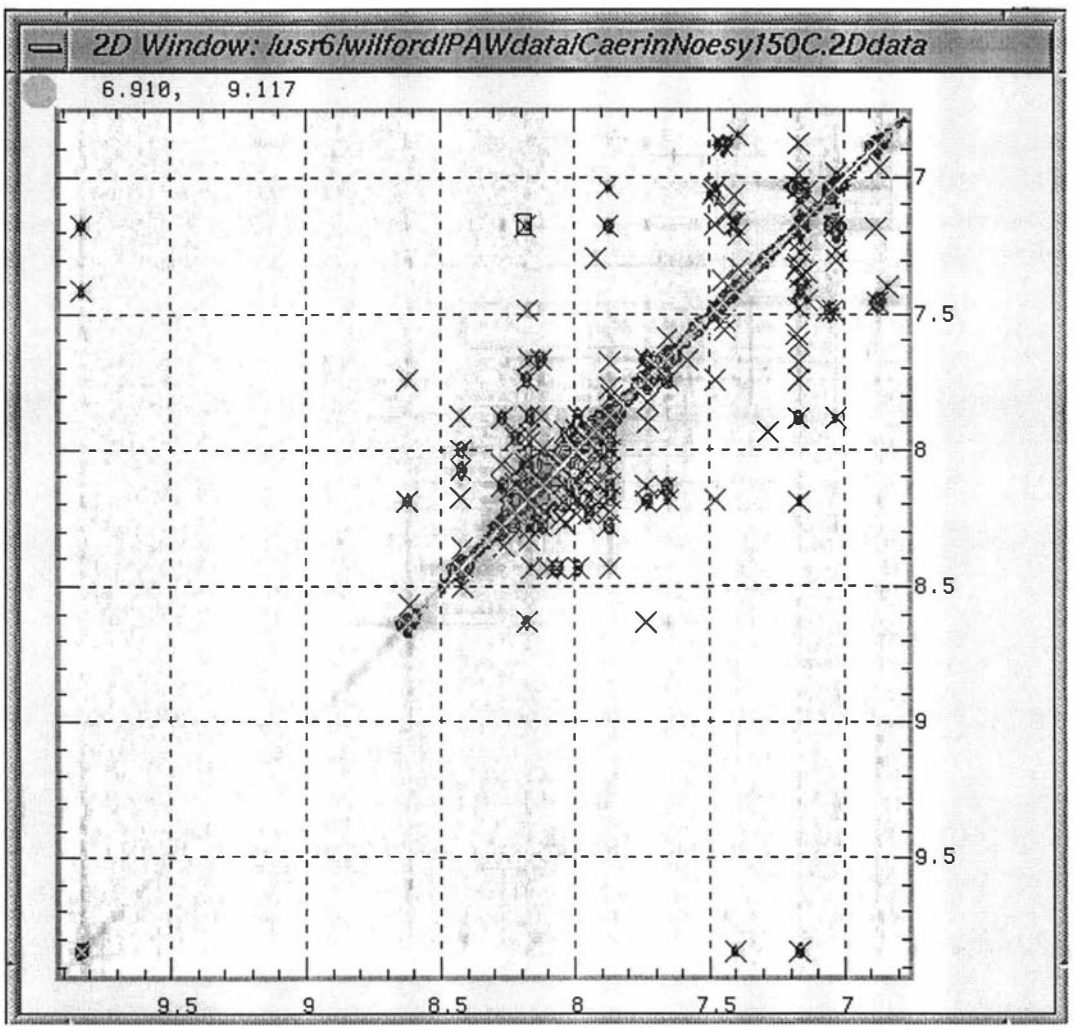


Figure 9.11 The raw peaks at the lower-left region of the **CaerinNoesy150C** spectrum. They are picked by setting a base level calculated from a small region at around [8.134,7.124].

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

9.4 Picking Diagonal Peaks

➢ Picking the diagonal peaks in the lower-left region

- Type z4.
- Type dr or click in the plot with MsBtn#3 to re-display the peak symbols if the X Window manager has just refreshed the draw-window.
- Choose [PickDiagPeaks] in the *Peak-picking Toolbox*. The message "*Use MsBtn#2 to select two terminal diagonal peaks for peak-picking.*" will appear above the plot.
- Use the middle mouse-button to select the two peaks at the two ends of the displayed diagonal segment. All the diagonal peaks in the plot will be picked and displayed in different colours, as shown in a multi-region plot in Figure 9.12. The diagonal peaks in the entire region had been closely checked and refinement was not found to be necessary.

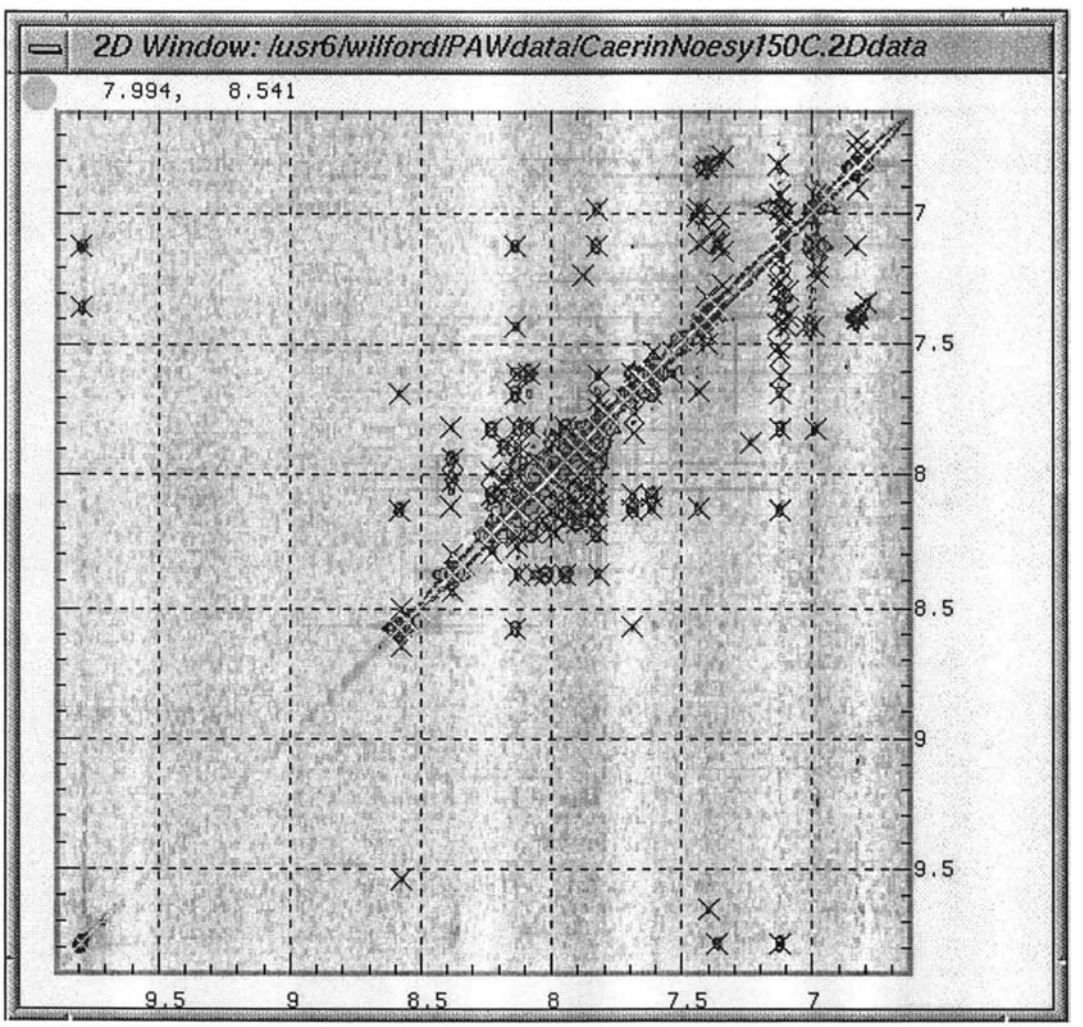


Figure 9.12 The diagonal peaks at the lower-left region of the **CaerinNoesy150C**. They are picked by selecting the two peaks at the two ends of the displayed diagonal segment

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

➢ Picking diagonal peaks in the upper-right region

The procedure to pick the diagonal peaks in this region is as follows:

- Type z3.
- Type dr to re-display the peak symbols if the X Window manager has just refreshed the draw-window.
- Choose [PickDiagPeaks] in the *Peak-picking Toolbox*. The message "*Use MsBtn#2 to select two terminal diagonal peaks for peak-picking*." will appear above the plot.
- Use the right mouse-button to select the two peaks that locate the two ends of the displayed diagonal segment. Since there are no serious Bloch-Siegert shifts in the spectrum, all the diagonal peaks in the plot will be picked and displayed (Figure 9.13).

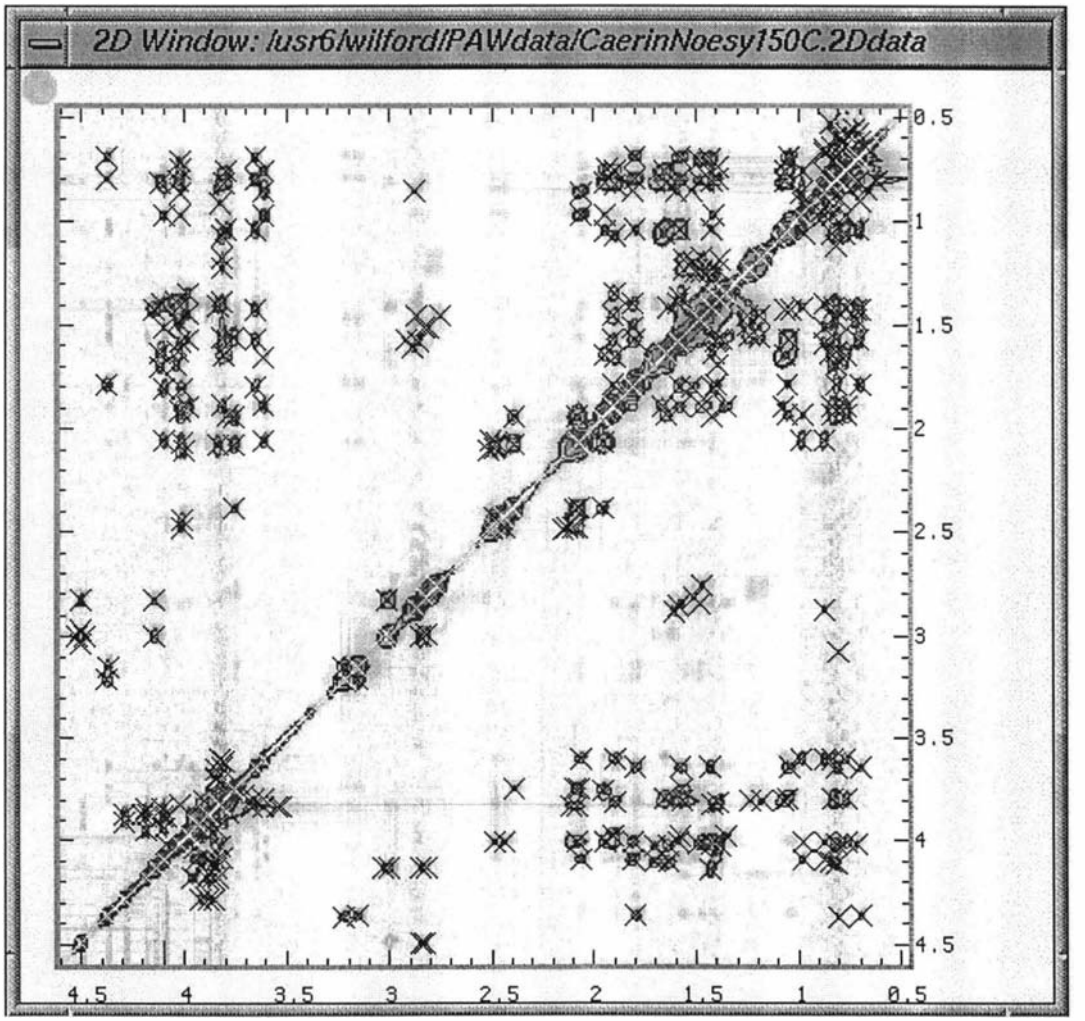


Figure 9.13 The diagonal peaks at the upper-right region of the **CaerinNoesy150C** spectrum. They are picked by selecting the two peaks at the two ends of the displayed diagonal segment

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

Note:

- When picking diagonal peaks, the operation will not be performed unless the two terminal peaks are quite accurately located. For this reason, it may be better to perform this operation with a larger draw-window. Repeat the process if the draw-window does not respond immediately after the second peak is selected. It may be that one of the locations selected was incorrect.

- The colour for the diagonal peaks was set to yellow for the distributed package. To change the colour, use any editor to open the file *PAW.colours* and change the colour setting for *DPeakColour* to any of the X Window predefined colours, which can be found in the file /usr/lib/X11/rgb.txt.

➢ Refining diagonal peaks in the upper-right region

Having all diagonal peaks properly picked is essential for cross-peak picking and any other operation that requires a correctly calculated transposed location. For this reason, it is recommended that the diagonal-peak list be refined as soon as a segment of diagonal peaks is picked. To do this, it is necessary to zoom into individual smaller regions in which all the diagonal peaks can be clearly seen. It is also necessary to draw diagonal lines in a region in order to assess the alignment of the diagonal peaks.

The following steps are required:

- Type zs and use *MsBtn#1* to select a region that covers about one-third of the diagonal peaks in *Zm#3*, as shown in Figure 9.14. (The diagonal peaks in the other regions were verified and found acceptable.)

- Choose [Dr / Line] in the *Drawing Toolbox*. Then, select the two diagonal peaks that are located at the two ends of the region. A diagonal line will be drawn, which connects almost all the diagonal peaks except four, as shown inside the two rectangles in Figure 9.14.

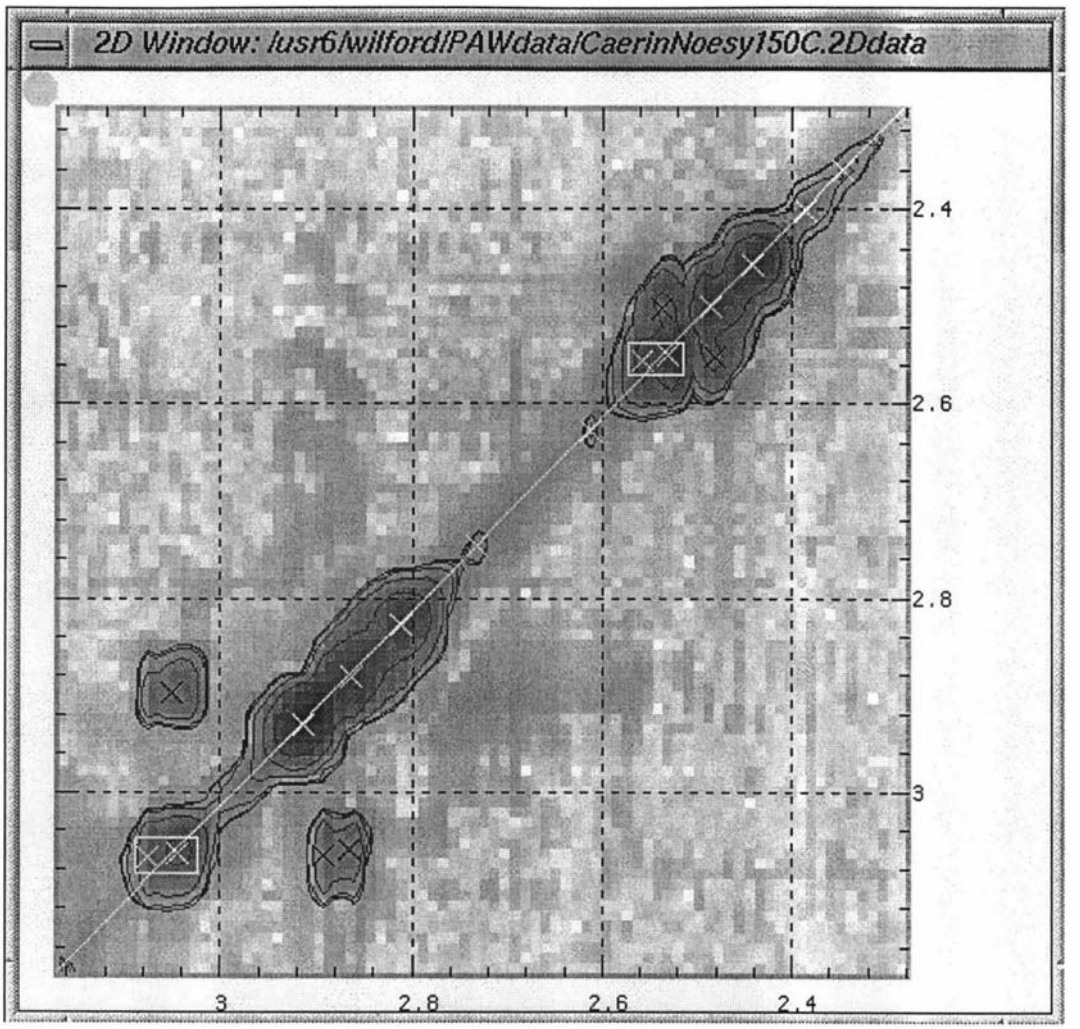


Figure 9.14 A diagonal line drawn for assessing the correctness of the diagonal peaks that were automatically picked. Nearly all of the diagonal peaks picked are properly located except the four enclosed in the two rectangles.

- Repeatedly choose [RmveDPks] and use *MsBtn#1* to select the rectangles to remove the four peaks. The colour of the peak symbols will be changed (Figure 9.15), indicating that they have been removed from the diagonal-peak list.

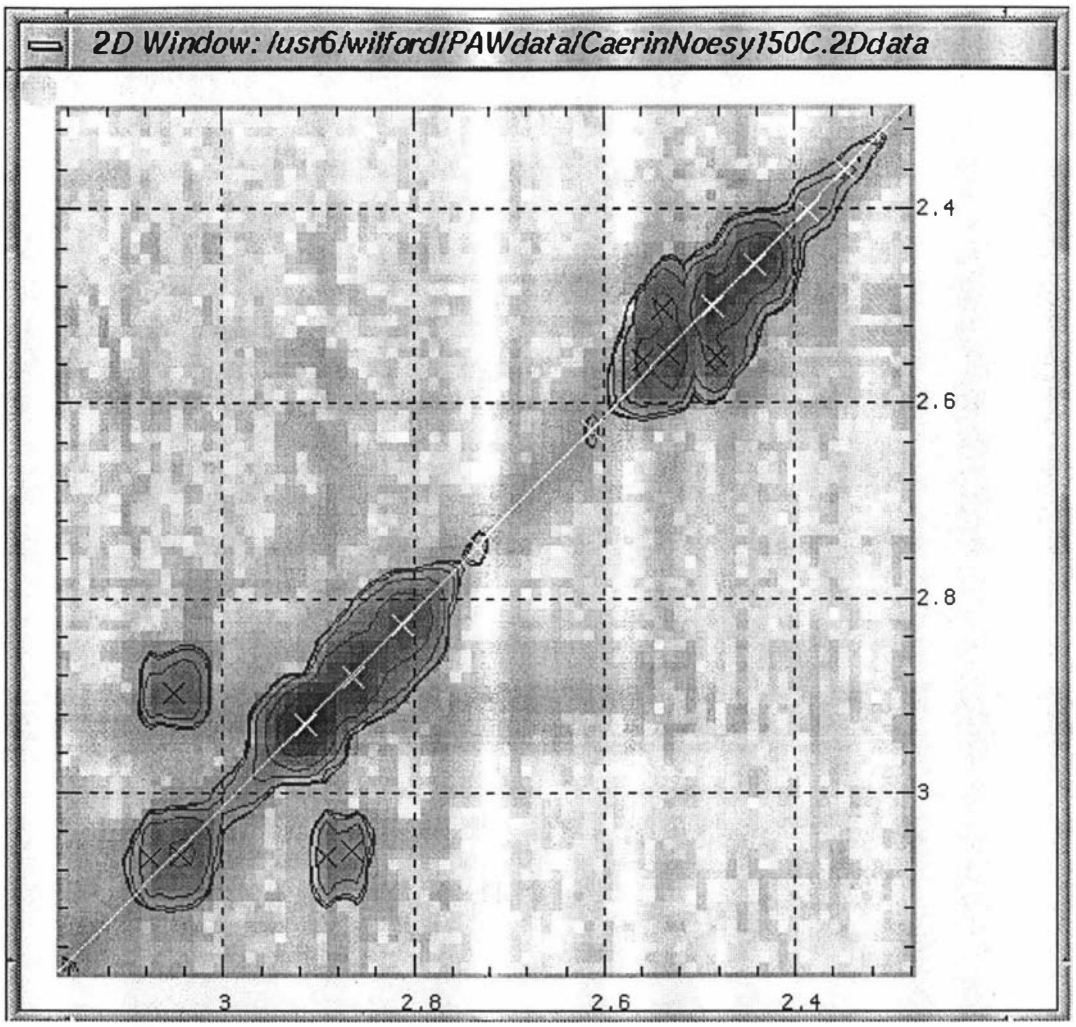


Figure 9.15 A set of carefully selected diagonal peaks after removing four peaks in the initial diagonal-peak list. The remaining diagonal peaks were sufficient to define the diagonal.

- Choose [RmAllLines] in the *Drawing Toolbox* to remove the line.
- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

Note:

- A badly located diagonal peak is often associated with heavy overlap, and must either be removed or re-located. Often, the bad diagonal peaks can be simply removed as long as the line that joins the adjacent two diagonal peaks passes the approximate location of them. Caution must only be taken when the diagonal curve is not straight due to the existence of severe Bloch-Siegert shifts.

9.5 Picking Cross-peaks

At this stage, all raw peaks and diagonal peaks have been picked and saved. The next task is to pick the cross peaks in the four regions, as described below. The process can be started from any region once all diagonal peaks have been picked.

➢ Picking cross peaks in the lower-left region

- Type z4.
- Choose [PickCrossPeaks] in the *Peak-picking Toolbox.*

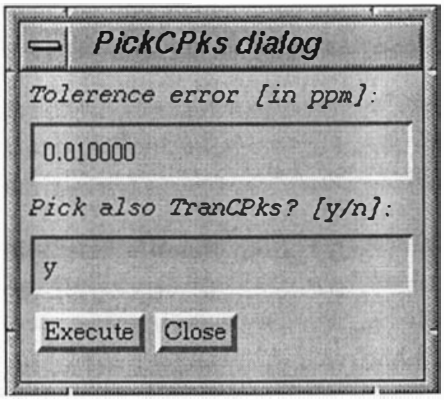


Figure 9.16 The PickCPks dialog.

- Use the left mouse-button to select the entire region. Make sure that the entry is y for the second item in the *PickCPks dialog* (Figure 9.16) so that only the peaks that have a transposed component in the raw-peak list are to be picked.
- Choose [Execute] to start the operation. It can be seen from Figure 9.17 that a significant number of the raw peaks are not picked. Most of them are noise.

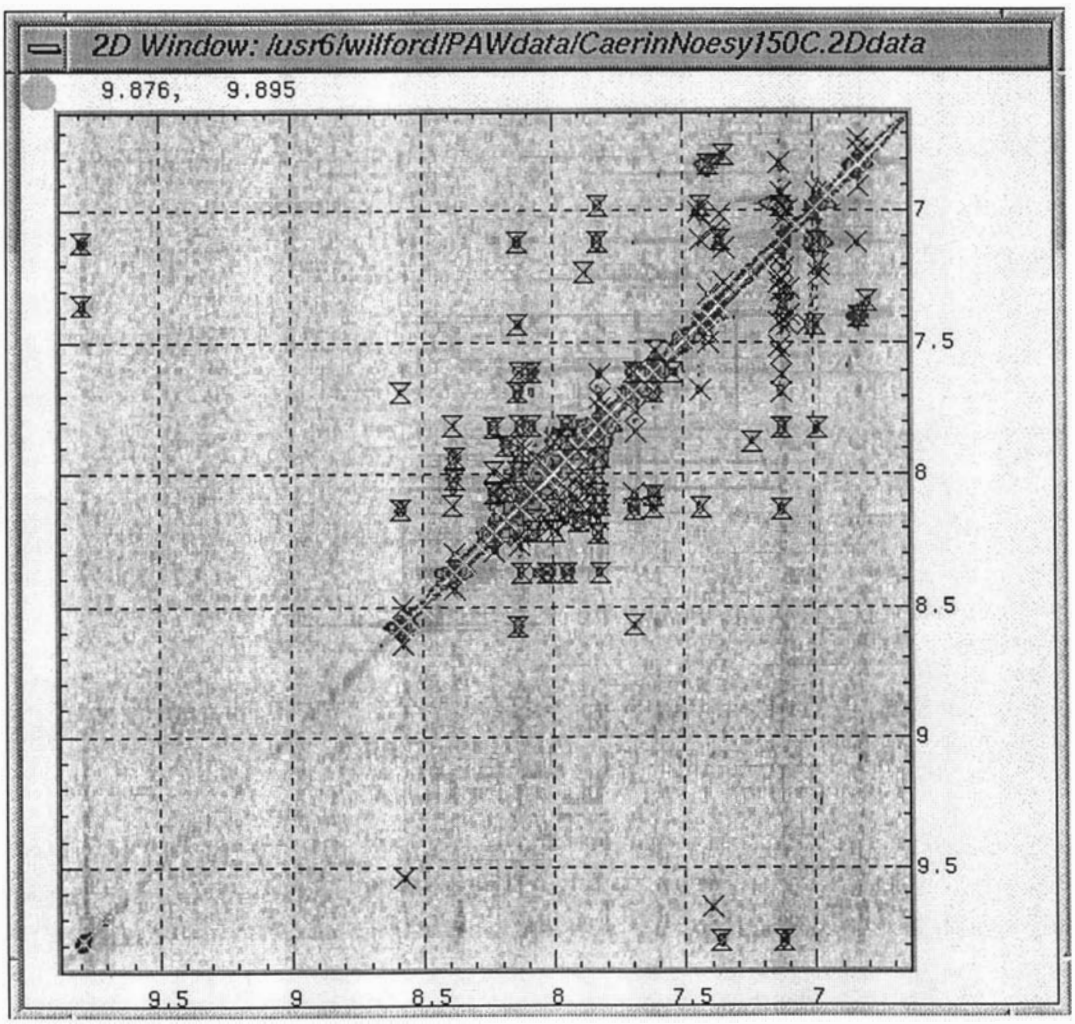


Figure 9.17 The cross-peaks in the lower-left region of the Caerin 4.1 NOESY150 spectrum. Each picked peak is represented by an hour-glass symbol.

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

➢ Picking cross peaks in the upper-right region

- Type z3 to zoom back to the upper-right region.
- Choose [PickCrossPeaks] in the *Peak-picking Toolbox*. The message '*Use MsBtn#1 to select a region for picking cross peaks*.' will appear above the plot.
- Use the left mouse-button to select the entire region and wait for the *PickCPks dialog* (Figure 9.18) to open. Make sure that the entry is y for the second item so that only cross peaks paired with transposed peaks will be picked.
- Choose [Execute] to start the operation. All paired cross peaks in the region will be picked and displayed with symbols that look like an hour-glass instead of a simple cross like ×, as shown in Figure 9.6. It can be seen that only a small fraction of the raw peaks are not picked, and most of these are noise.

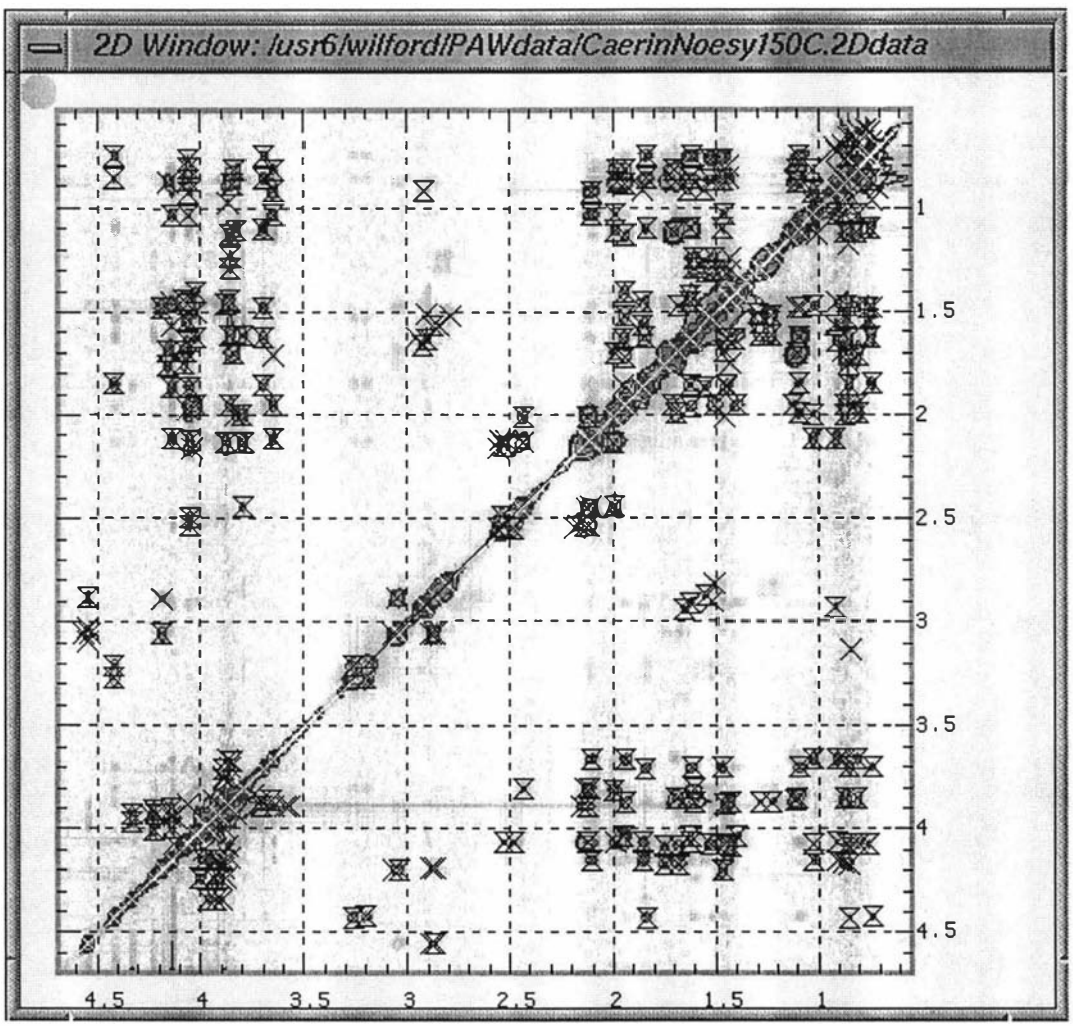


Figure 9.18 The cross-peaks in the upper-right region of the Caerin 4.1 NOESY150 spectrum.

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

➢ Picking cross peaks in the upper-left and lower-right region

This process is more easily done by cross-referencing to both regions. A typical process can be as follows:

- Type z1 to zoom into the upper-left region.
- Choose [PickCrossPeaks] in the *Peak-picking Toolbox*.
- Use the left mouse-button to select the entire region. Again, make sure that the entry is y for the second item in the *PickCPks dialog*.
- Choose [Execute] to start the operation.

All the paired cross peaks in the upper-left and lower-right region will be picked and displayed on the plot, as shown in Figure 9.19 and Figure 9.20.

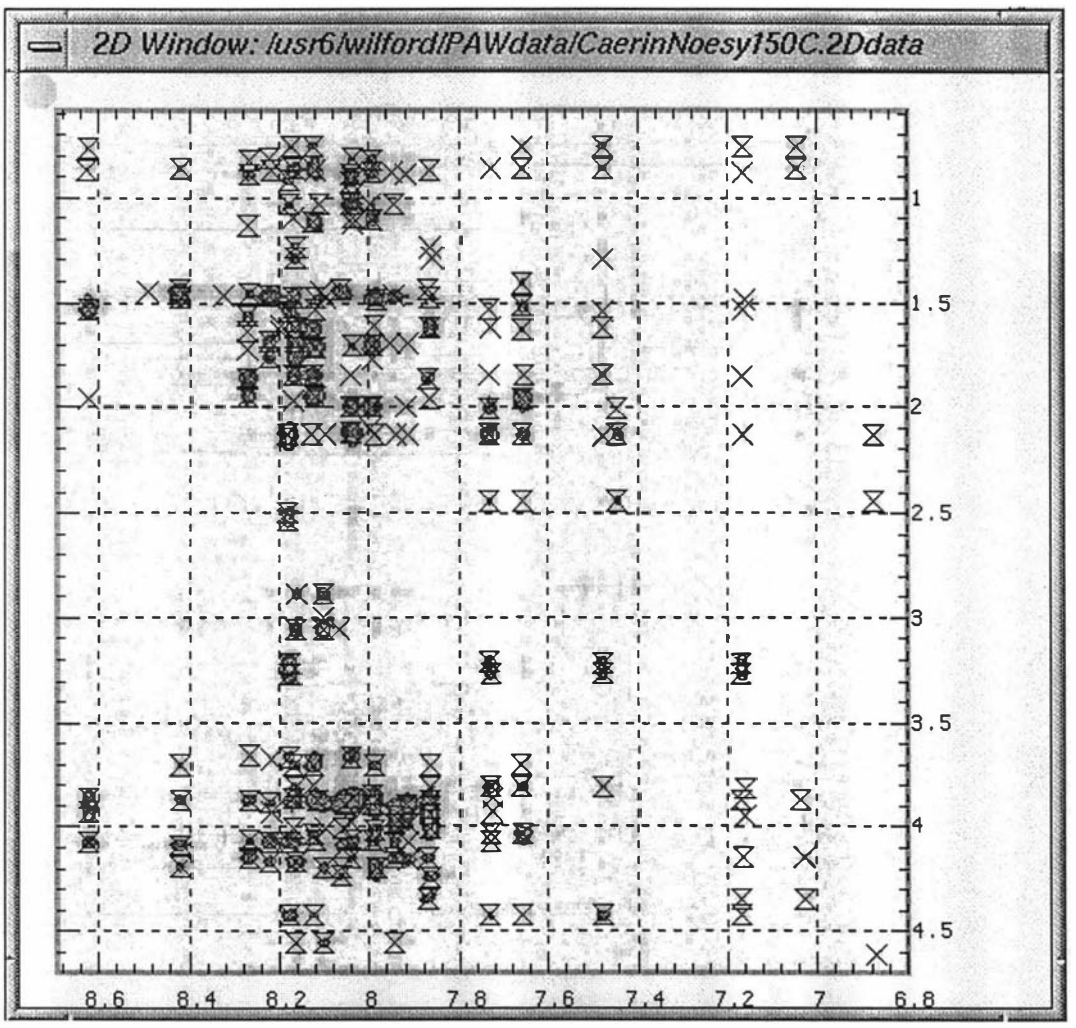


Figure 9.19 The cross-peaks in the upper-left region of the Caerin 4.1 NOESY150 spectrum.

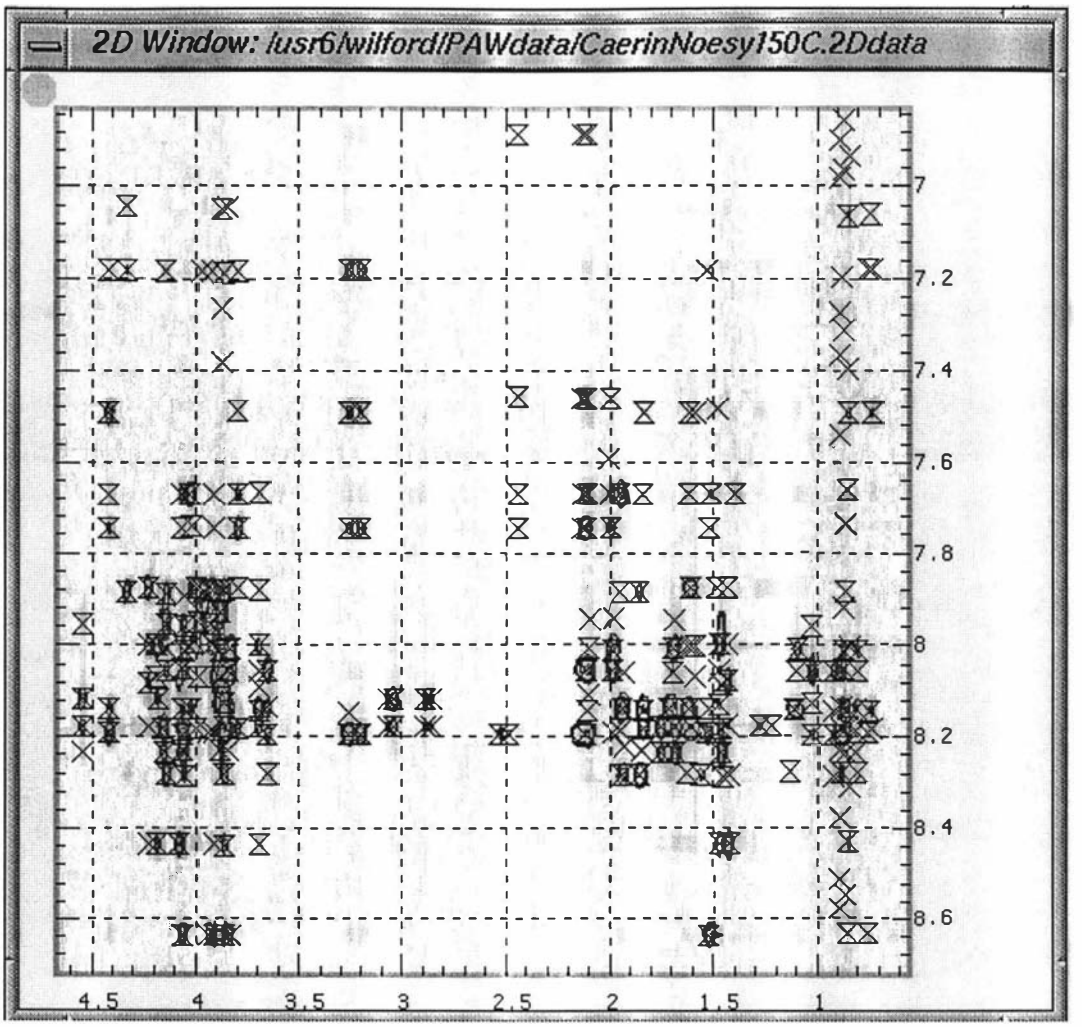


Figure 9.20 The cross-peaks in the lower-right region of the Caerin 4.1 NOESY150 spectrum.

- Choose [SavePkLists] in the *Peak-assignment Toolbox* to save the result again.

➢ Assessing missing transposed cross peaks

When picking cross peaks with an automatic operation, there are always a small number of unpaired raw peaks that are not picked. Some of them are real, and some of them are noise. Missing transposed cross peaks are produced for a number of reasons, including the significantly lower resolution in D2, improperly phased data in one dimension, improperly picked diagonal peaks, heavily overlapped peaks, and local noise. The time that is required for refining a cross-peak list depends on the number of transposed-peaks that are missing or not properly picked. This is usually the most tedious and time-consuming part of the entire picking process because each case has to be judged individually. A better option is perhaps to pick the unambiguous peaks at the start, and then consider the ambiguous peaks individually at a later stage of the spectral assignment.

In general, it is better to zoom into smaller regions so that all peaks can be observed properly. Cross-examination between transposed regions can be used as often as

necessary to find missing transposed-peaks. Each operation may also involve drawing transposed-rectangles to locate the corresponding transposed peaks. PAW provides a simple command to find and add a pair of symmetrically located cross peaks in one operation. This requires the diagonal peaks be picked properly, as explained previously.

The following example describes a process for picking missing pairs of peaks:

- Type zs to zoom into a region from approximately [8.70,3.600] to [7.60,4.60], and define the region to be Zm#5 (Figure 9.21). This region contains all HN:HA cross peaks in the **CaerinNoesy150C** spectrum.

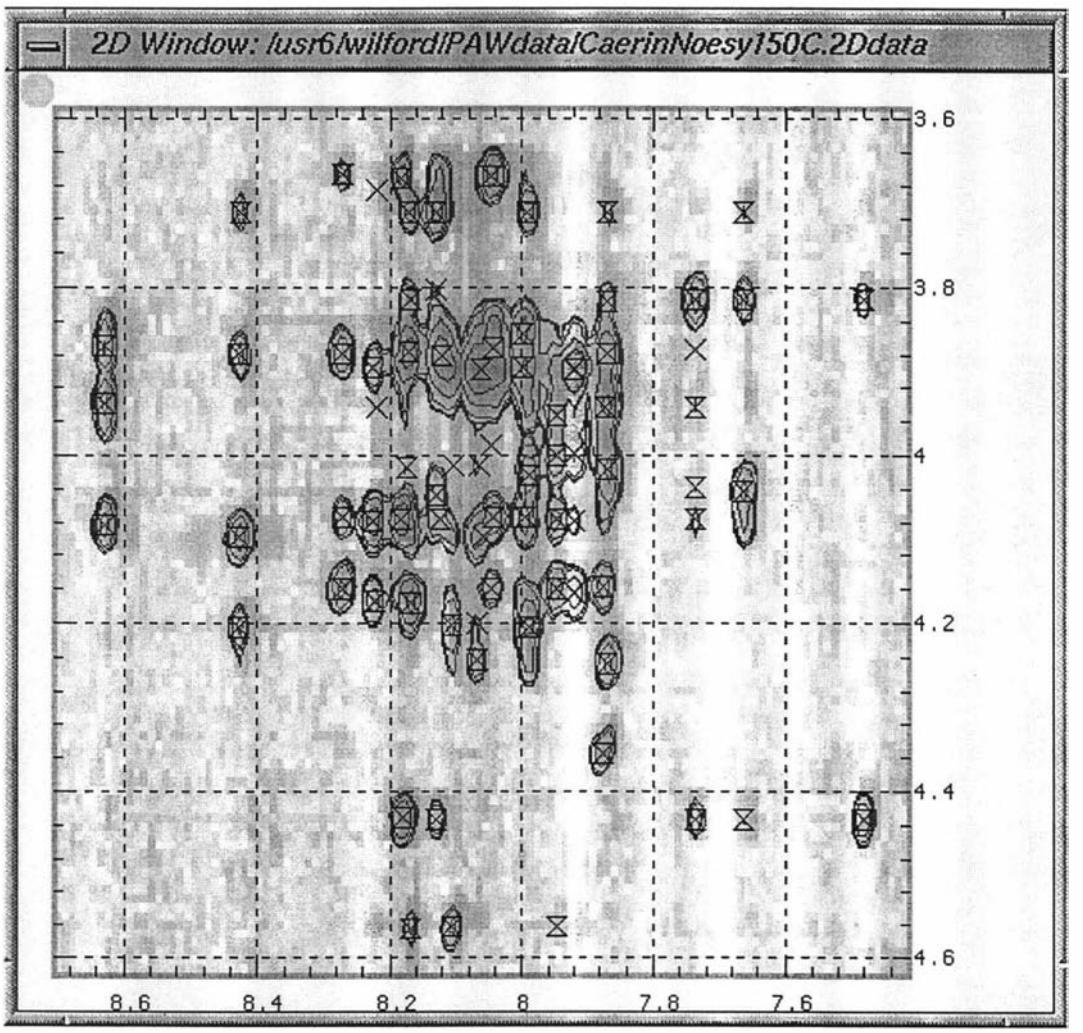


Figure 9.21 An expanded view of the region from approximately [8.70,3.600] to [7.60,4.60] that is defined as Zm#5.

- Type zt to zoom into its transposed region, and define the region to be Zm#6 (Figure 9.22).

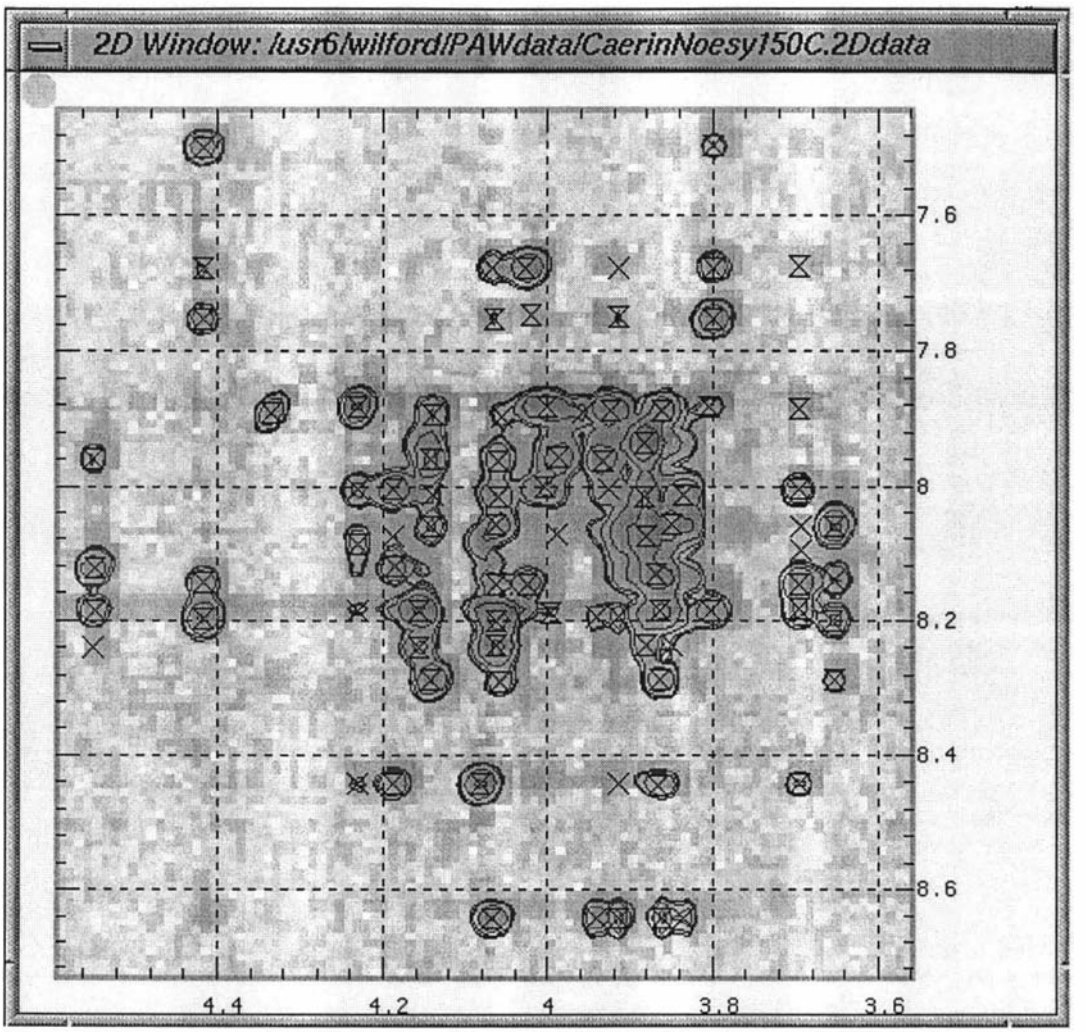


Figure 9.22 The transposed region of Zm#5, which is defined as Zm#6.

- While in Zm#6, type dtr (for drawing transposed rectangles) and use the middle mouse-button to click on a number of raw peaks that can be clearly identified as "un-picked" cross peaks. When finished, press the right mouse-button to stop the operation. For illustration, four transposed rectangles were drawn in Figure 9.23.

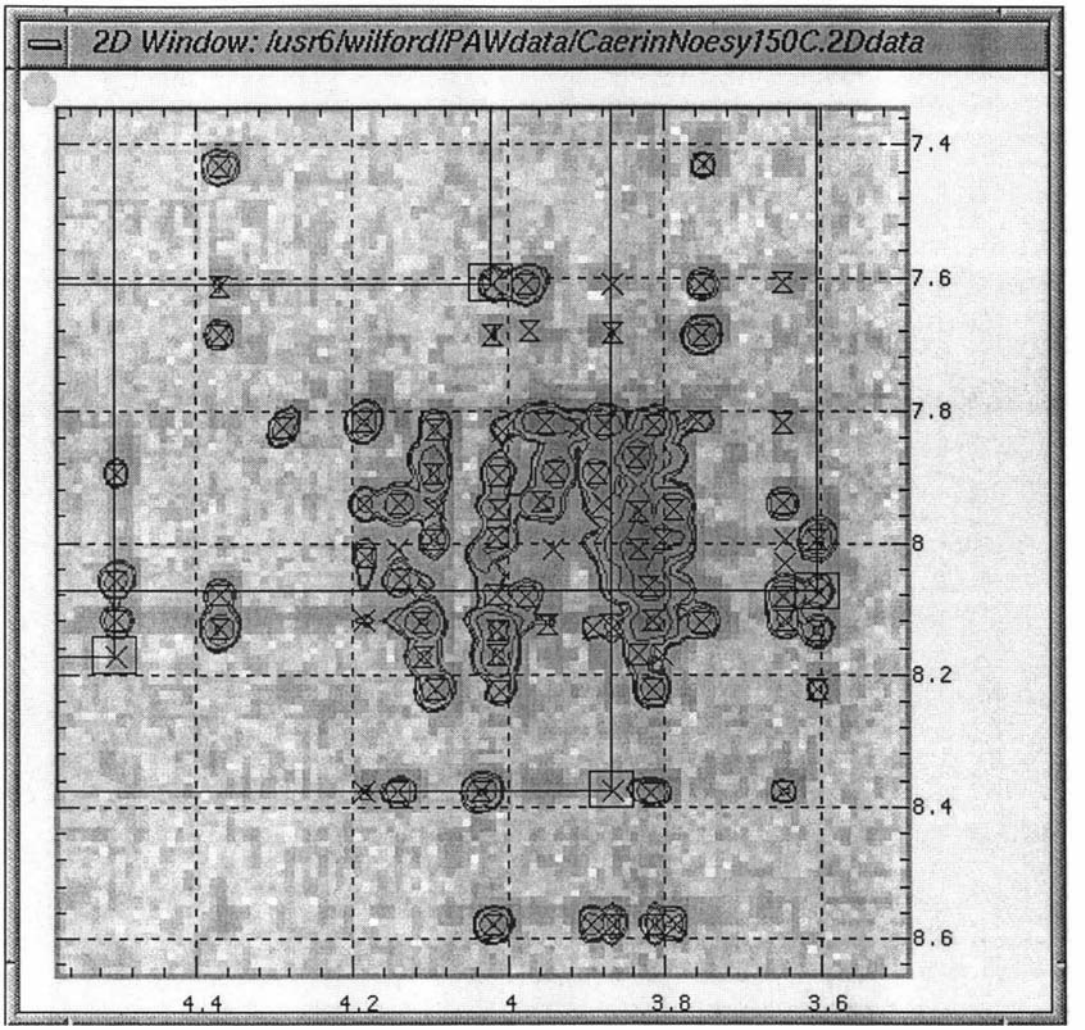


Figure 9.23 A smaller region with four transposed rectangles drawn for the assessment of possible missing transposed peaks.

- Type z5 to zoom into the transposed region (Figure 9.24). Look carefully at the corners of each rectangle (shown in small rectangles) to check if any peak also appears in the transposed region.

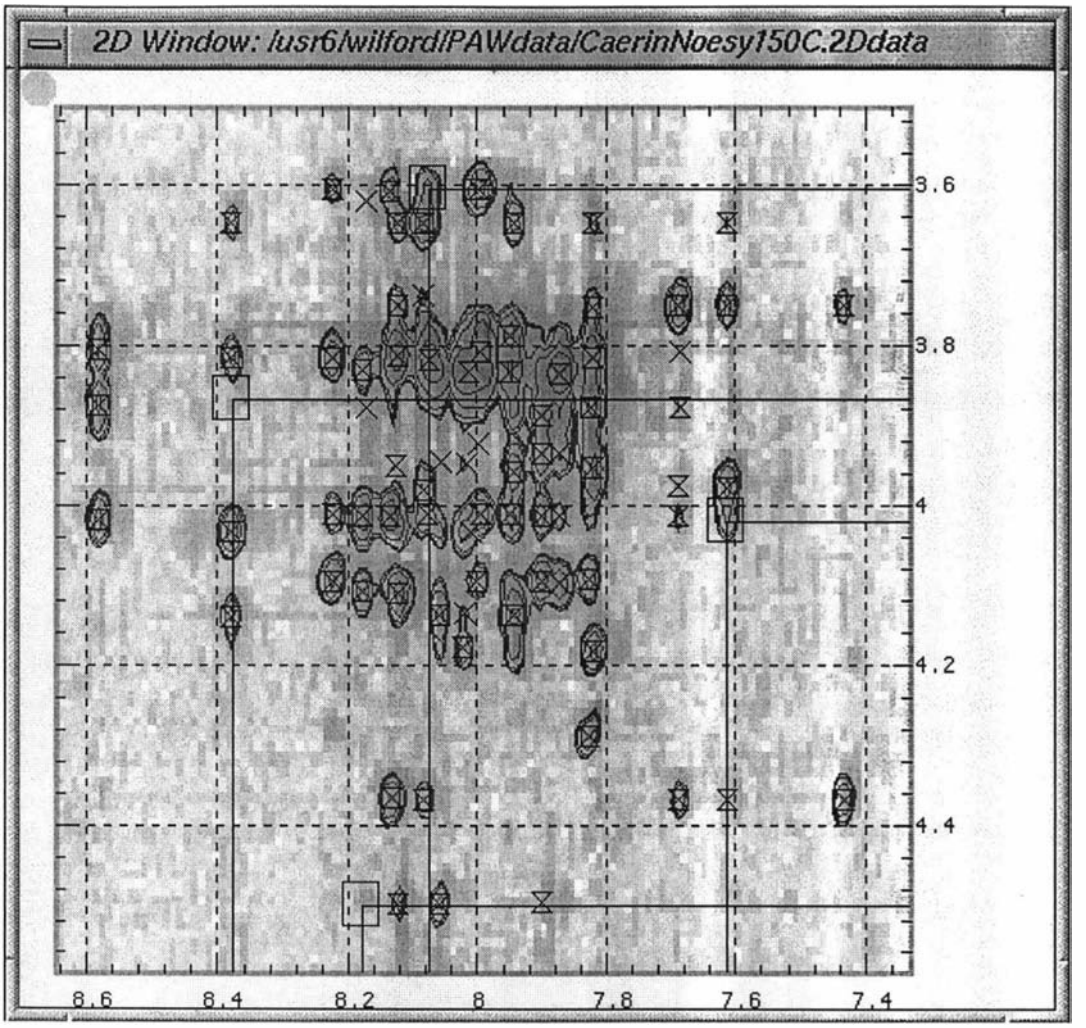


Figure 9.24 The transposed region of the last plot.

Often, additional evidence can be obtained by looking at a row or a column of data. For example, the 1D spectrum in Figure 9.25 convincingly confirmed the existence of the peak inside the small rectangle at the bottom of Figure 9.24. By contrast, the column at around 8.35 ppm does not shown any evidence of such a peak (Figure 9.26); therefore, the transposed rectangle can be removed.

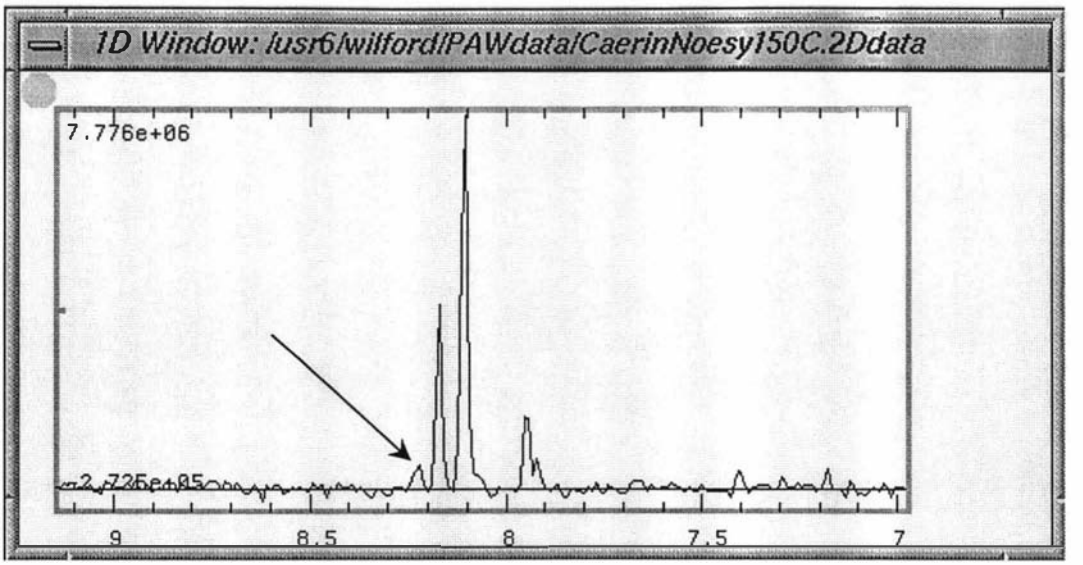


Figure 9.25 An expanded plot of a row drawn from Figure 9.16 at around 4.56 ppm. The arrow points to a peak that is inside the rectangle at the bottom of Figure 9.16.

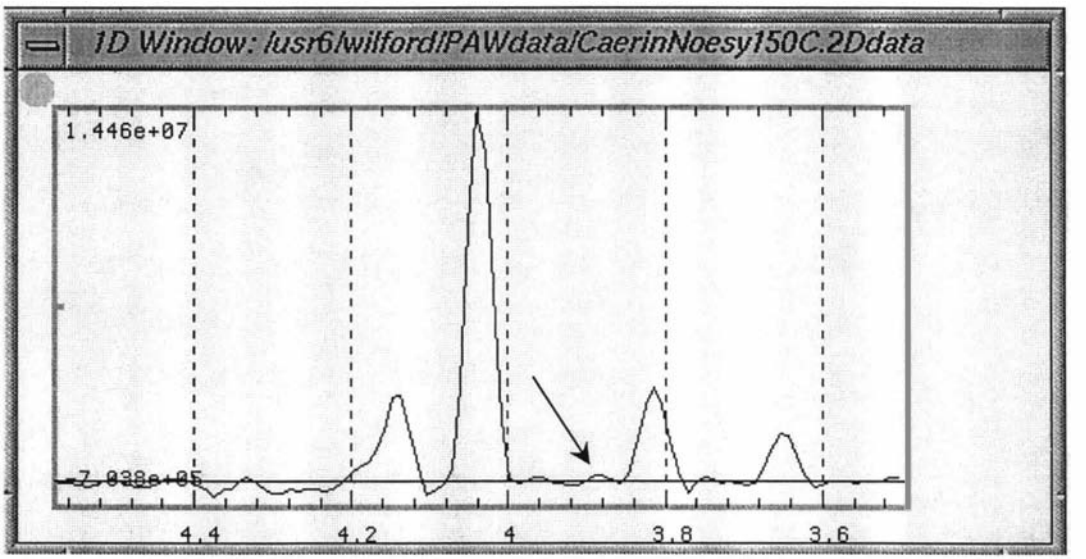


Figure 9.26 An expanded plot of a column drawn from Figure 9.16 at around 8.35 ppm. The arrow points to the insignificant peak that is inside the rectangle on the left of Figure 9.16.

- To remove the rectangle at 8.35 ppm, type r1r and click on one of its corners with *MsBtn#2*.

➢ Adding missing transposed cross peaks

- Type z6 (figure not shown). There should be only three rectangles left.
- Choose [FindTrnCpk] in the *Peak-picking Toolbox*. The message "*Use MsBtn#2 to set the peak position for picking a pair of transposed peaks*." will appear above the plot.
- Click on every peak at each corner of the remaining rectangles with the middle mouse-button.

- Type z5. There should be three cross peaks added, each of them at the corner of a transposed rectangle (Figure 9.27).

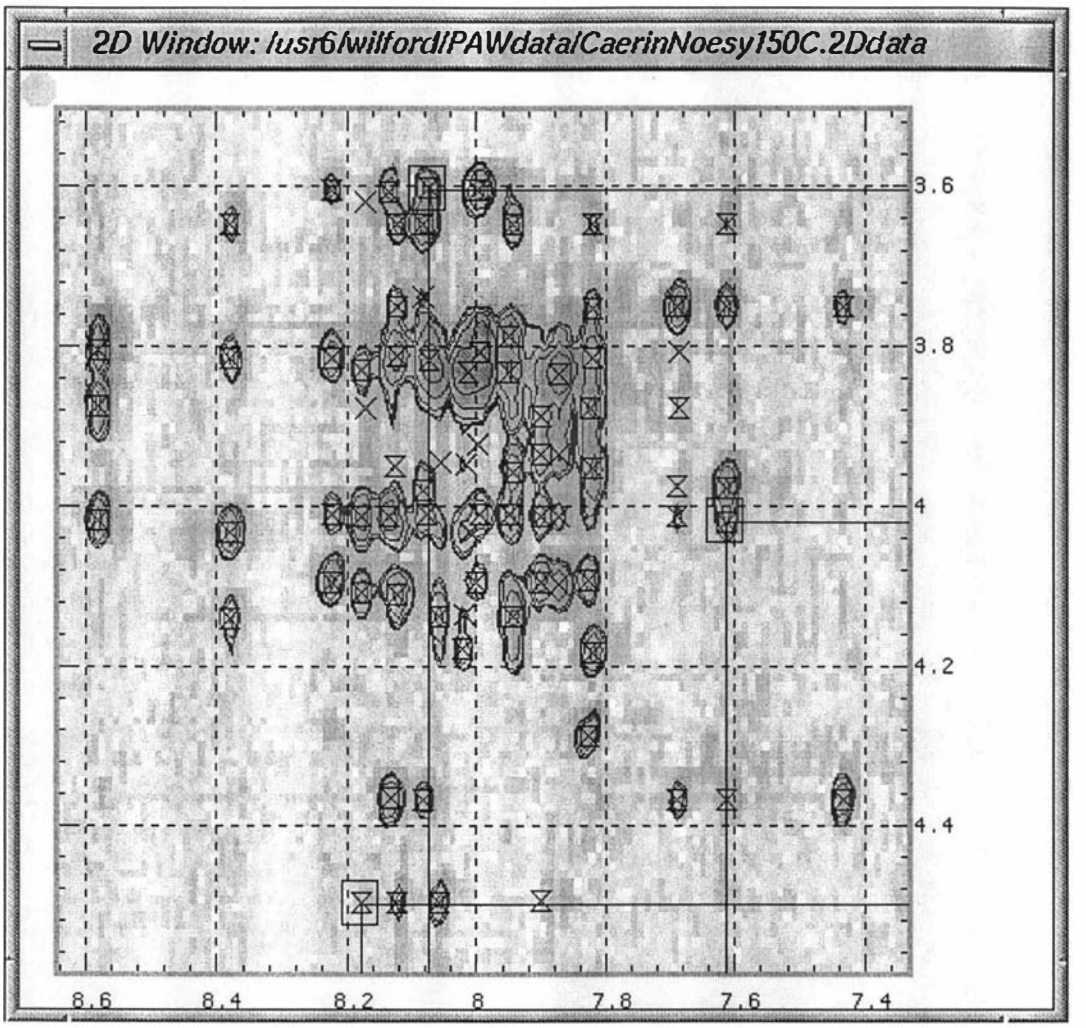


Figure 9.27 An example of three cross-peaks that were found and added.

- Choose [RmAllRects] in the *Drawing Toolbox* to remove all rectangles once the results are satisfactory.
- Starting from Zm#5, repeat the last six steps to find the missing transposed cross peaks in Zm#6.

Chapter 10:

Spin-system Identification Process for Caerin 4.1

10.1 Introduction

This chapter describes the spin-system identification operations for the Caerin 4.1 NMR spectra.

The high-resolution spectra and refined peak lists obtained by the methods described in chapters 8 and 9 were used in the process, namely:

- **CaerinNoesy150C.2Ddata**
- **CaerinTocsy150C.2Ddata**
- **CaerinCosyC.2Ddata** (optional)
- **Caerin98.RPeaks**
- **Caerin98.DPeaks**
- **Caerin98.CPeaks**

To avoid unexpected errors, it is again recommended that PAW is restarted, and the workbench used in Chapter 7 is again loaded.

➢ The Caerin 4.1 sequence

Getting familiar with the protein sequence of Caerin 4.1 is obviously useful for the spin-system identification and spectral assignment. It is:

GLWQKIKSAA GDLASGIVEG IKS-NH_2

where NH_2 is the terminator. Accordingly, the amino acid list is:

Amino acid codes	Number in the sequence	Number identified
G	4	
L	2	
W	1	
Q	1	
K	3	
I	3	
S	3	
A	3	
D	1	
V	1	
E	1	
Total	**23**	

As can be seen, five of the amino acids appear only once each in the sequence. These are W3, Q4, D12, V18 and E19.

10.2 Preliminary

➢ Loading spectra for spin-system identification

- Start PAW and load the spectra **CaerinNoesy150C**, **CaerinTocsy150C** and **CaerinCosyC** as described in Chapter 9.

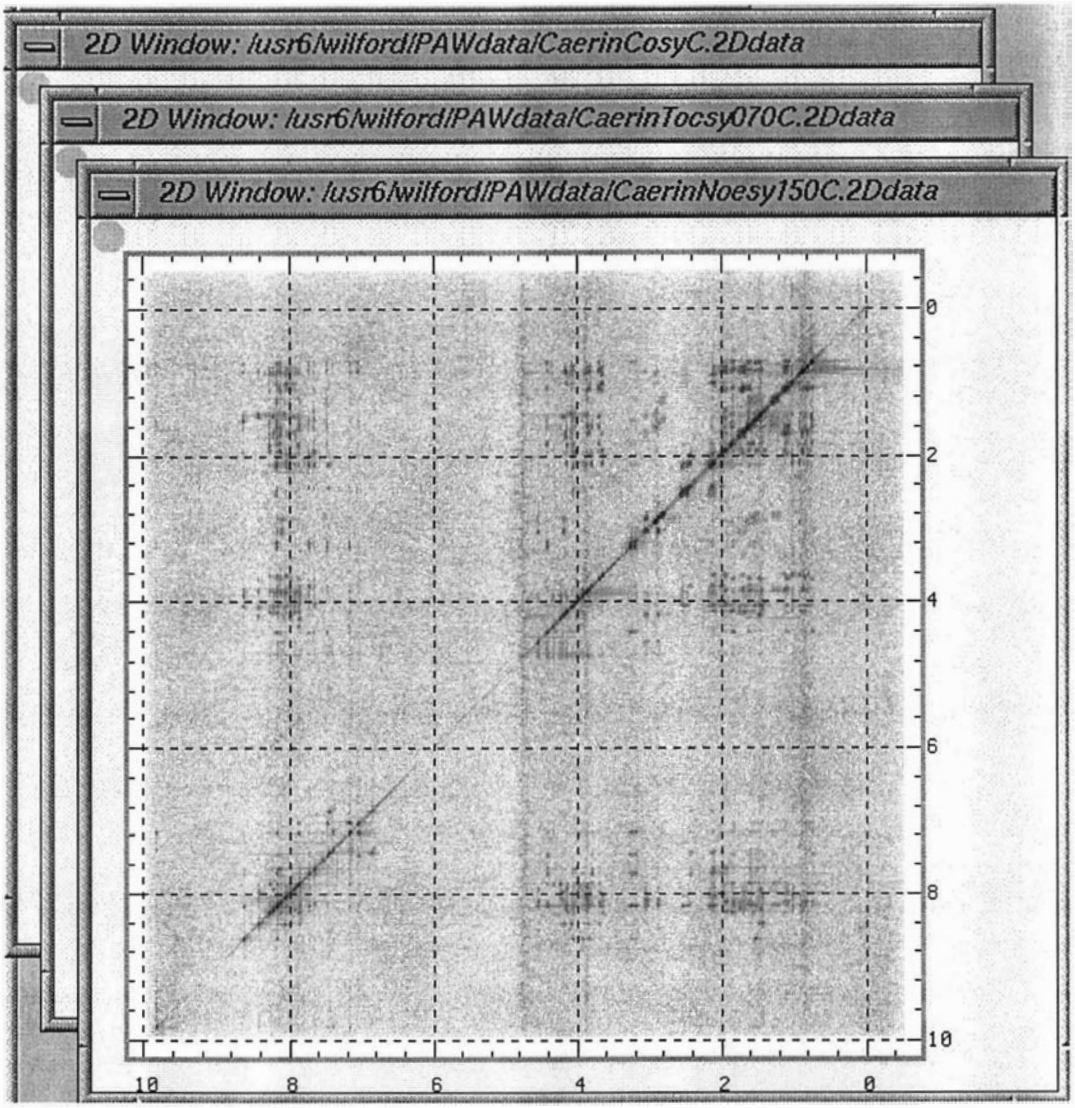


Figure 10.1 A full view of the **CaerinNoesy150C** spectrum, with the other two spectra hidden below.

➢ Defining zooming patterns for spin-system identification

- Define five zoomed-patterns for the spin-system identification operations, as shown in Figure 10.2.

The definitions will be useful during the spin-system identification process, especially Zm#5.

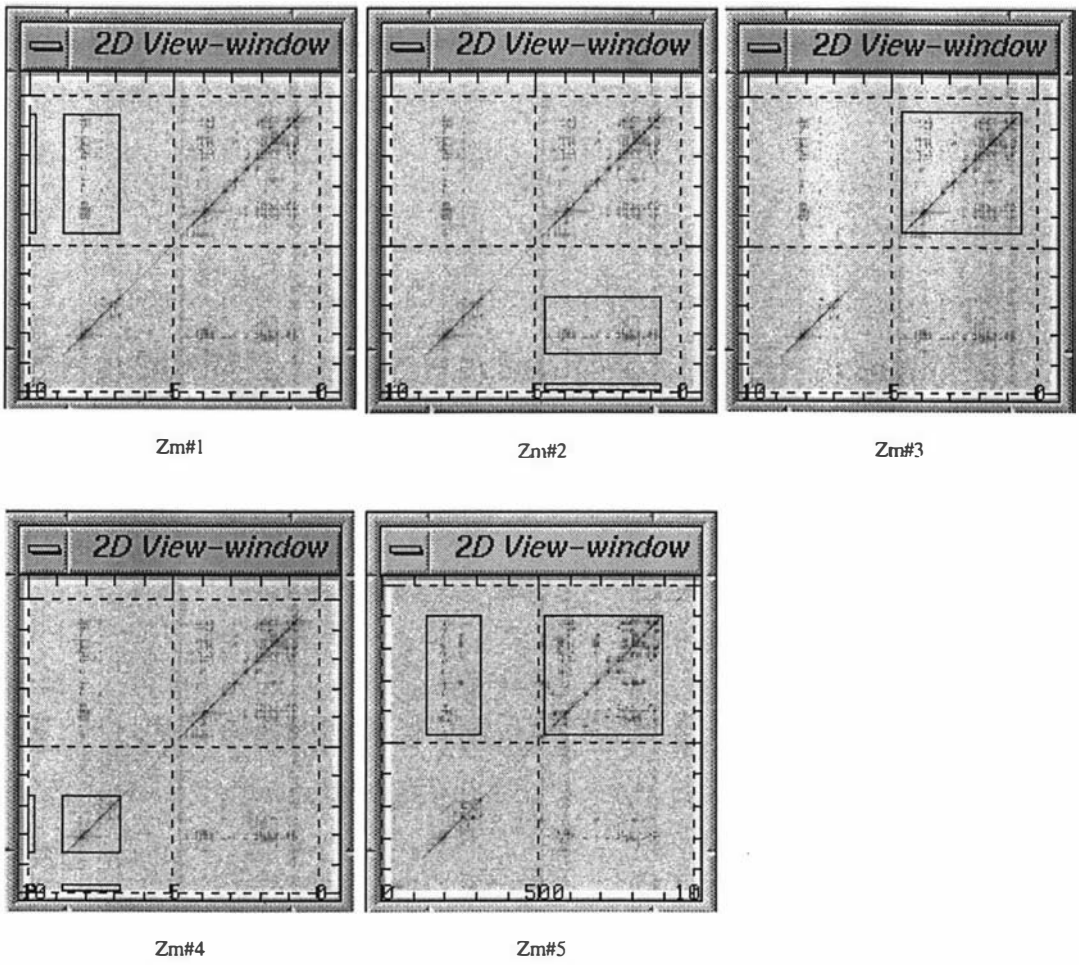


Figure 10.2 The Zm#1 to Zm#5 defined for the spin-system identification of Caerin 4.1.

The following steps describe the operations required to define the same pattern numbers for the **CaerinTocsy150C** window:

1. In the **CaerinNoesy070C** window, type zl and then zsp to zoom the **CaerinTocsy070C** window with the same pattern as that in the **CaerinNoesy070C** window.
2. On the **CaerinTocsy070C** window, click on the title-bar to bring it to the front, and then click in the plot area of the **CaerinTocsy070C** window.
3. Define this plot as Zm#1 for the **CaerinTocsy070C** window.
4. Repeat the last three steps to define Zm#2 to Zm#5 for the **CaerinTocsy070C** window.

Finally,

- Repeat similar operations above to define Zm#1 to Zm#5 for the **CaerinCosy150C** window.

After these operations, the commands z1, z2, z3, z4 and z5 can be applied to all three windows.

➢ Loading peak lists for spin-system identification

- Type zf.
- Type lpl to load a set of peak lists. This will open a file-selection dialog, as shown in Figure 10.3.

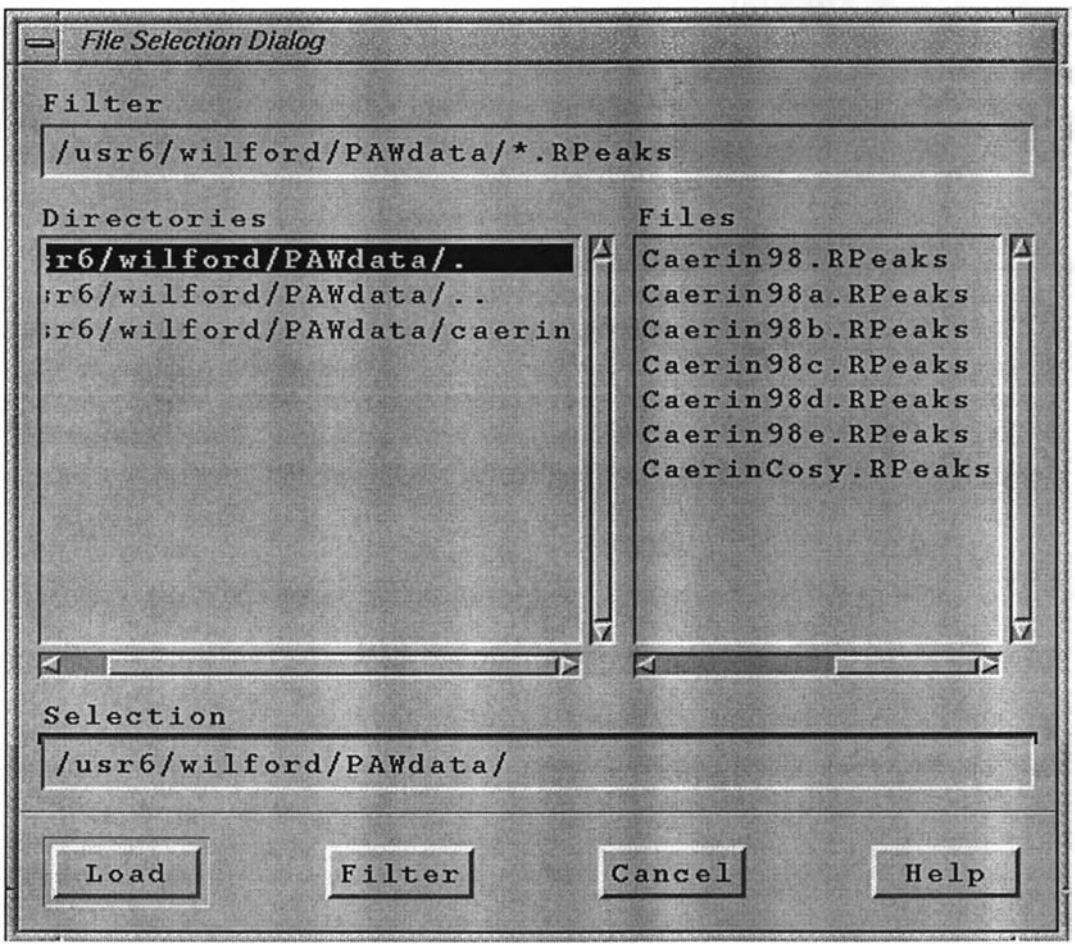


Figure 10.3 The file selection dialog for loading peak lists.

- Double-click on the name **Caerin98b.RPeaks** in the file list. The three peak lists mentioned before will be loaded simultaneously.
- Type dr to redraw the plot with the loaded peaks, as shown in Figure 10.4. (A quicker way to redraw peaks is to click in the draw-window with MsBtn#3.)

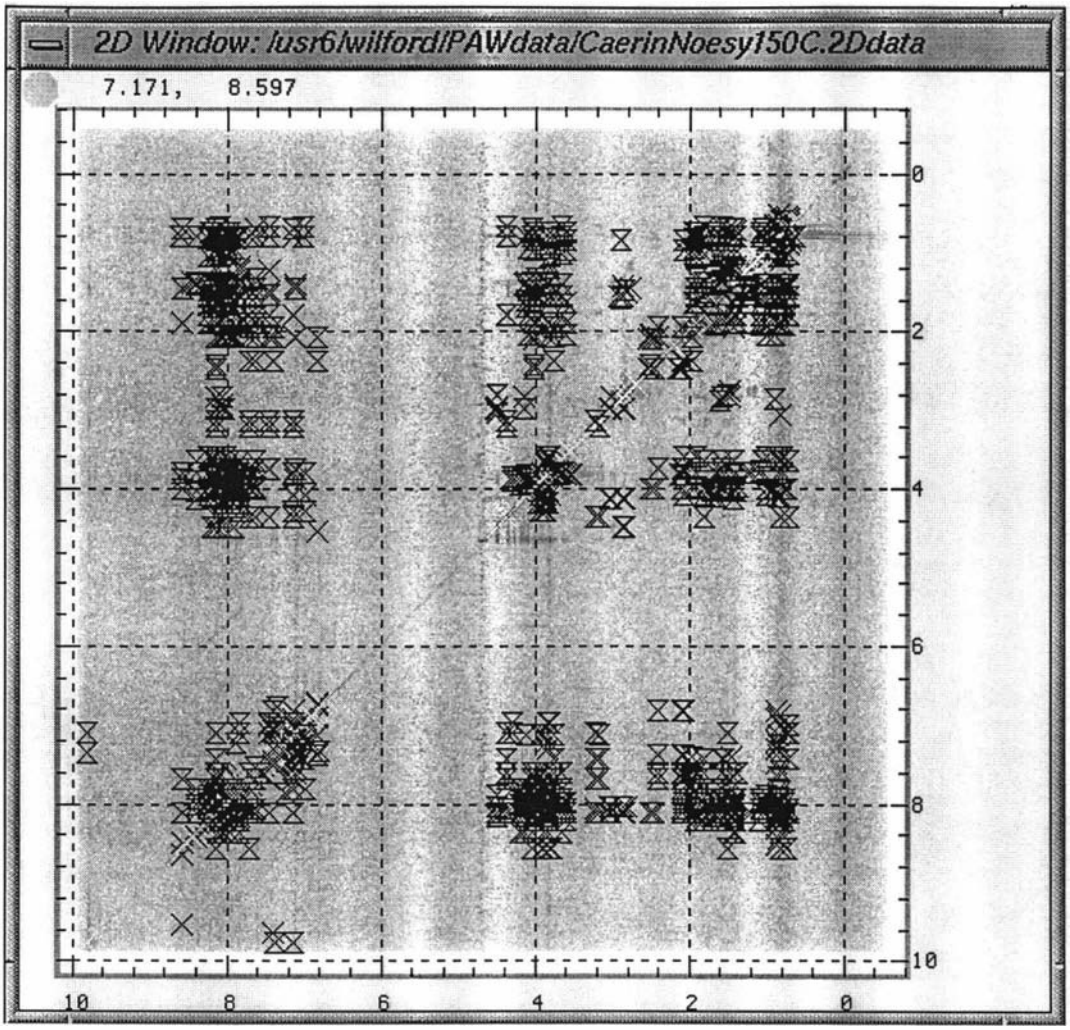


Figure 10.4 The **CaerinNoesy150C** with the loaded peak lists displayed on top.

Note that the peaks are often too crowded to be clearly seen in a small window. The following two useful operations for sizing a draw-window can be done as many times as needed in the identification and assignment process.

- To display a full-screen view of a plot, type Alt-F10.
- To restore a draw-window to its default size and position, click on the grey circle on the upper-left corner of the window.

10.3 Spin System Identification

This process is more easily done by reference to the **CaerinTocsy070C** spectrum, as it contains only cross-peaks that represent correlations between protons of the same spin system.

As mentioned in Chapter 10 of Volume I, during the spin-system identification process, tables 7.1, 7.2, 7.4 and Appendix 7.b in Volume I have been frequently referenced. They contain the basic, but useful, information on the primary-structures, spin-systems, average chemical-shifts and 2D cross-peak distributions of the 20 common amino-acids.

For the theory behind and comments on spin-system identification, see Chapter 10 of Volume I.

➢ Identification of Ala residues

- Click on the title-bar of the **CaerinTocsy070C** window to bring it forward, and then click on the plot.
- While the cursor is on the plot area, type z5 to recall Zm#5 (Figure 10.5).

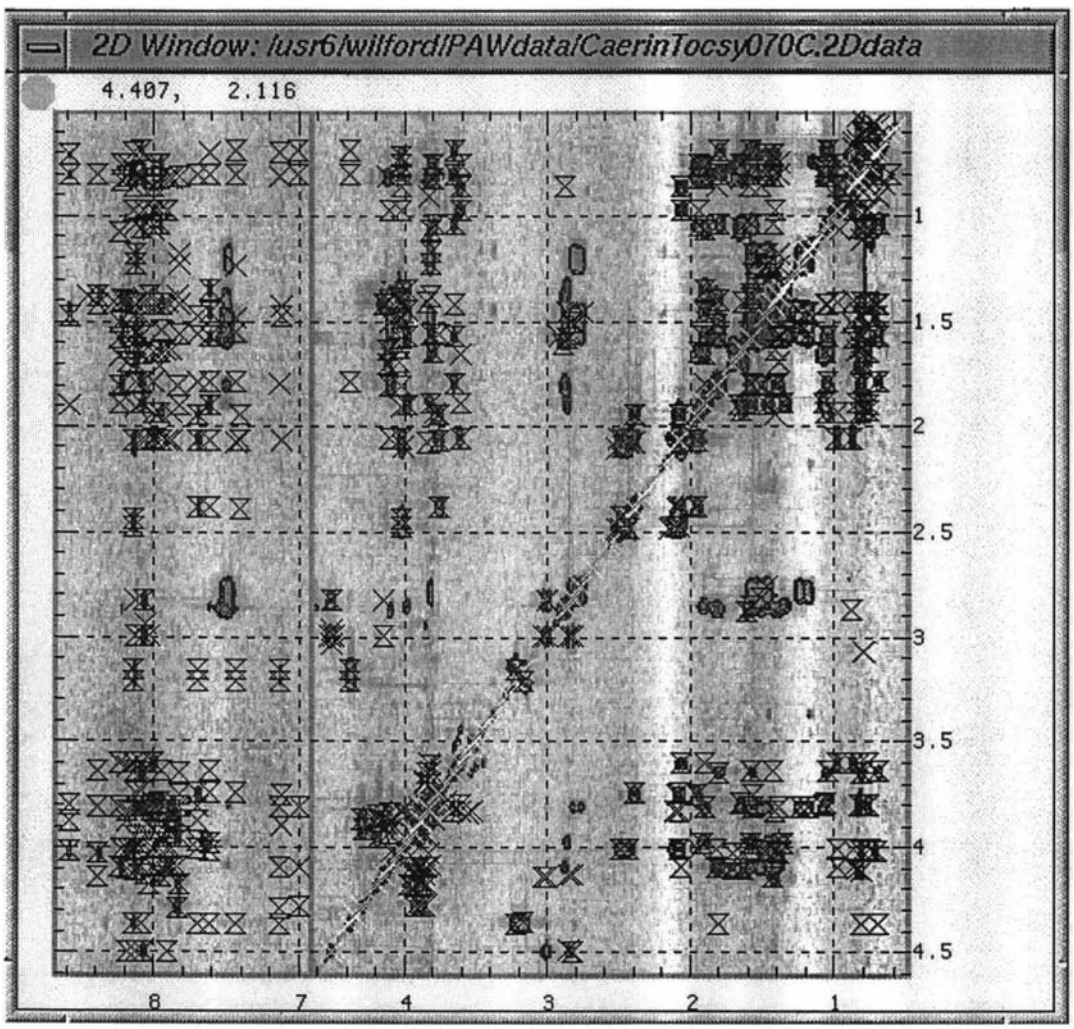


Figure 10.5 The expanded view of Zm#5 of the **CaerinTocsy070C** spectrum.

Type Alt-F10 to get a full-screen view of Zm#5.

- Click in the plot to display the peak symbols.
- Switch off the [Grid] button in the *Common-display Toolbox*.

The three Alanine spin-systems can be easily identified as follows:

- Type dch to draw crosshairs.
- Click on the peaks arrowed in Figure 10.6.

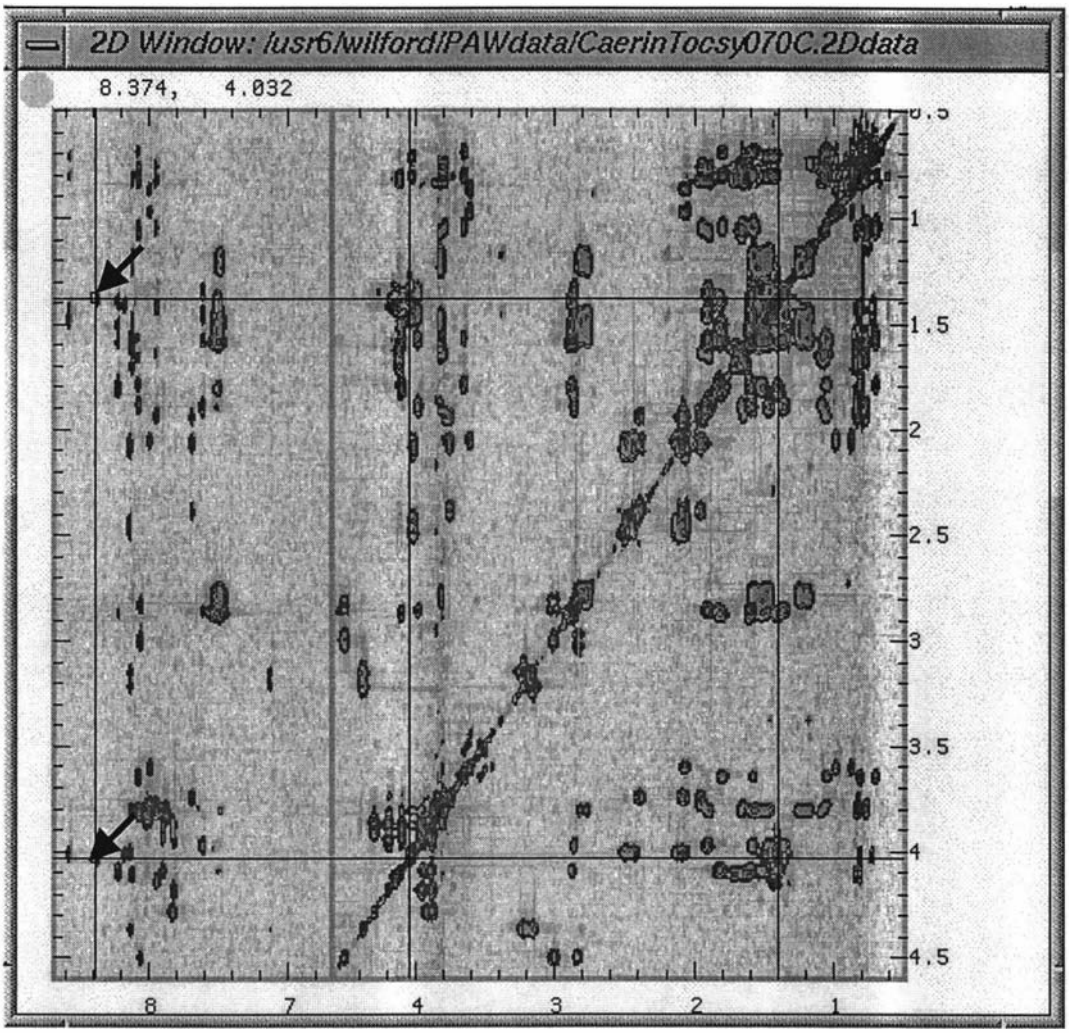


Figure 10.6 A multi-region plot that shows how one of the Ala spin-systems in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

If desired, the three TOCSY-related cross-peaks can be assigned and aligned immediately after the system is identified. For a systematic presentation, the procedures for assigning and aligning cross-peaks will be described in Chapter 11.

Once the cross-peaks are assigned, type ral to remove all lines in the plot and then continue doing the search for the other two Alanine spin systems in the same way as follows:

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in Figure 10.7.

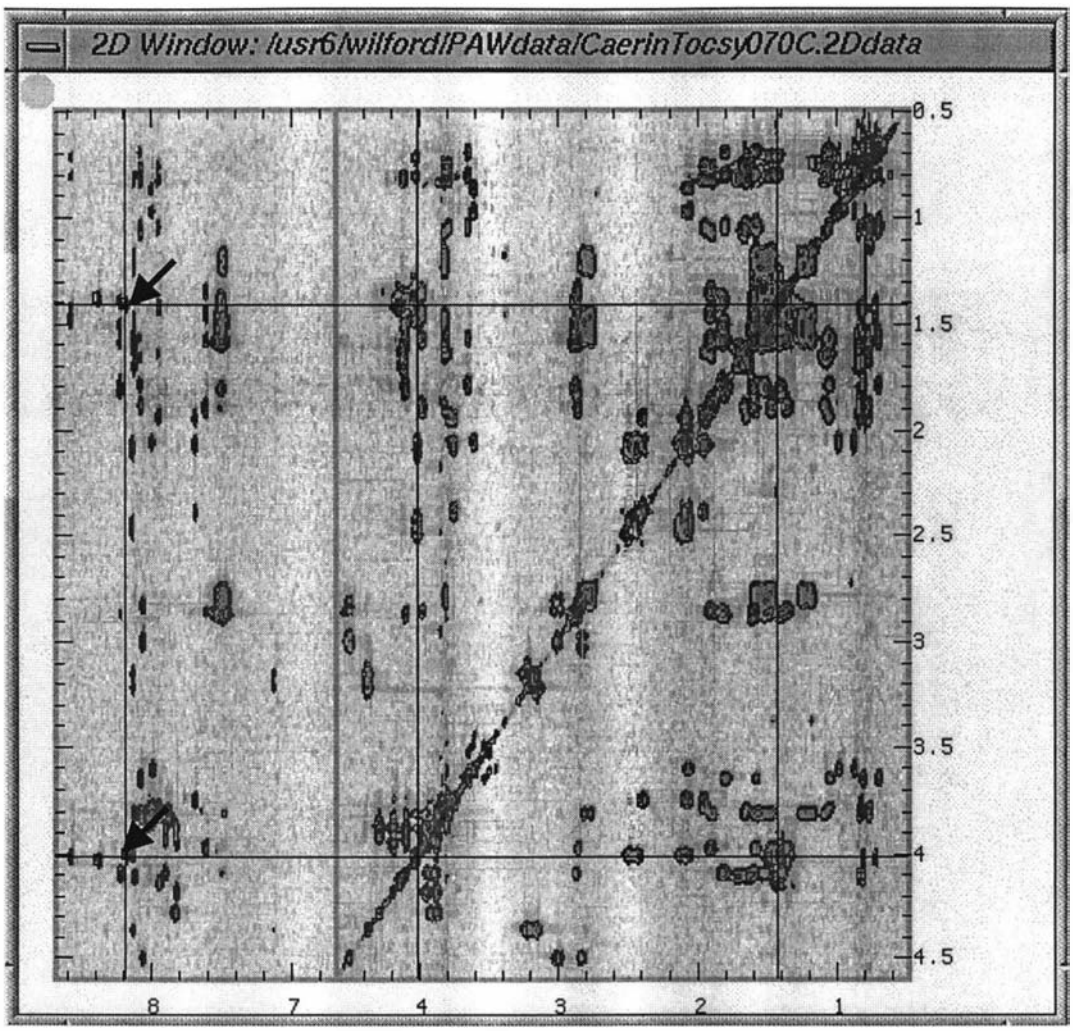


Figure 10.7 A multi-region plot that shows how the second Ala spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in Figure 10.8.

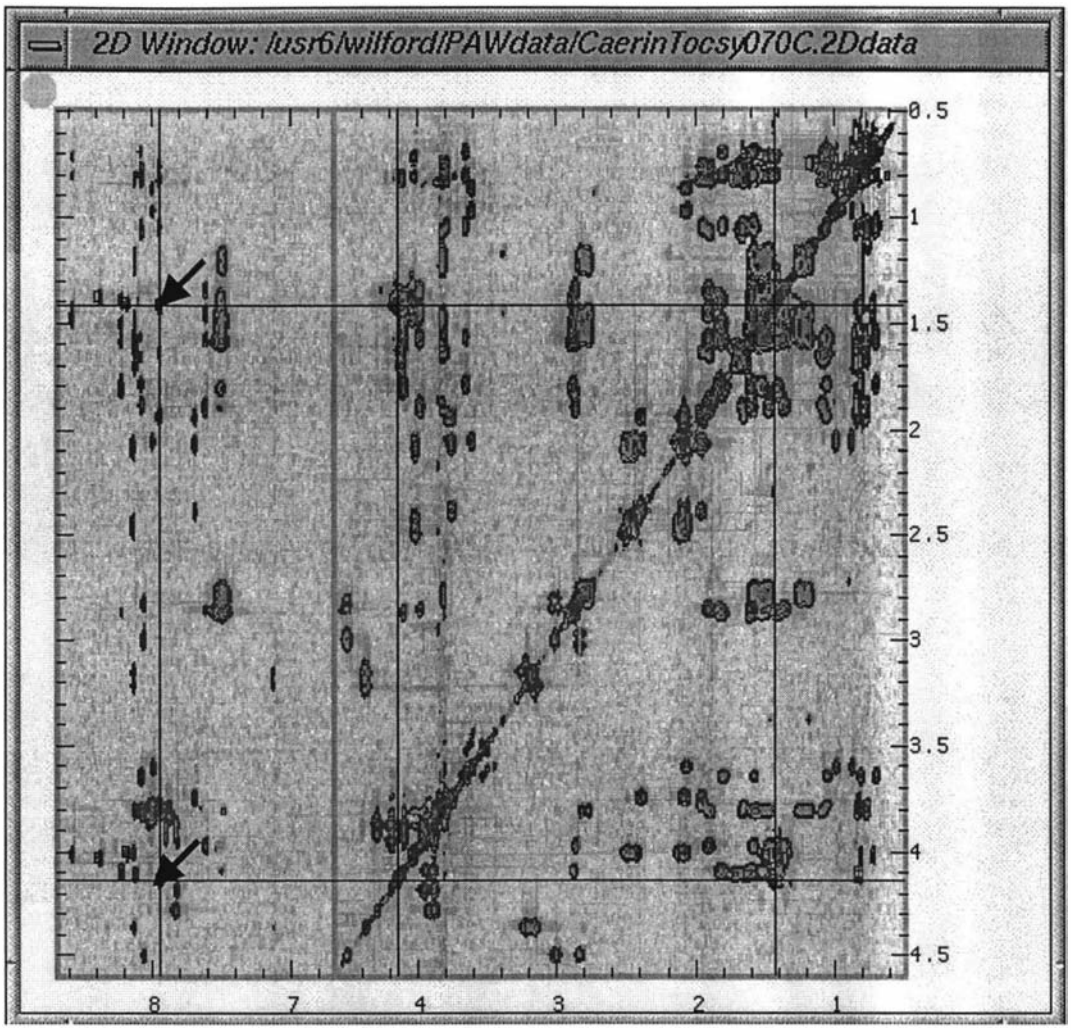


Figure 10.8 A multi-region plot that shows how the third Ala spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

The third Ala is only obvious after zooming into a much smaller region and comparing the 1D peak-shapes with the other five TOCSY-related peaks in the same column (Figure 10.9).

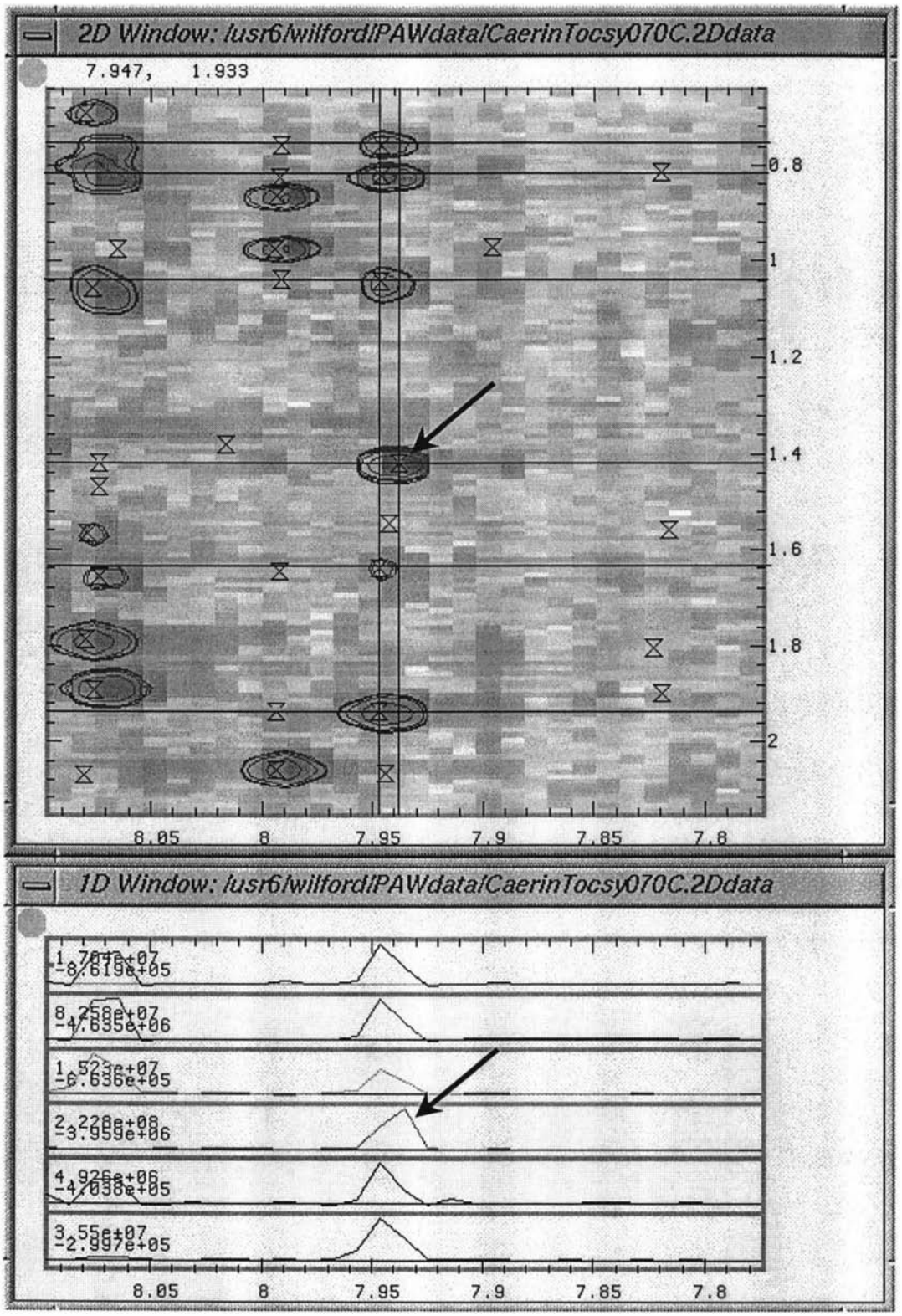
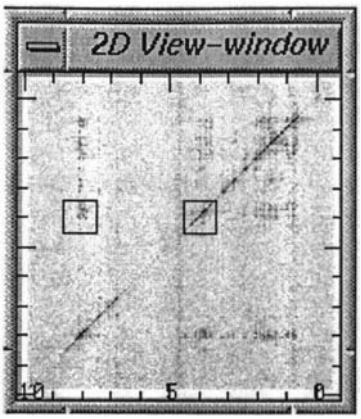


Figure 10.9 A expanded 2D and 1D view that shows clearly the HB cross-peak of an Ala residue, as arrowed in the plots. The six 1D plots show the six rows indicated by the horizontal lines in the 2D plot.

➢ Identification of Ser residues

The Serine-related cross-peaks are also easily identified as follows:



- Zoom into another multi-region pattern as shown in Figure 10.10.
- Type ral to remove all crosshairs.
- Type dch and click on the three diagonal peaks arrowed in Figure 10.11.

Figure 10.10 An overview of the next multi-region plot.

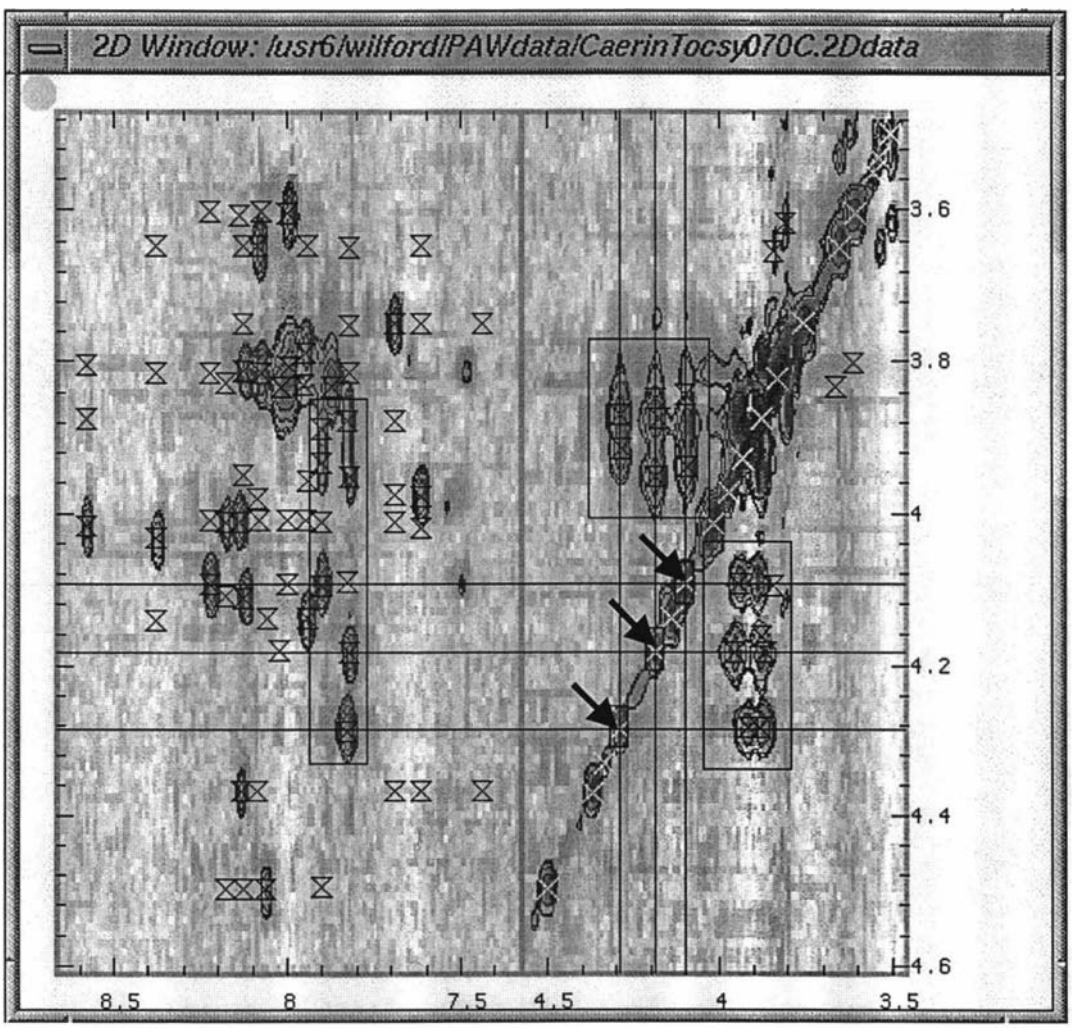


Figure 10.11 An expanded view used for the identification of Ser spin systems. Three sets of crosshairs were drawn from the arrowed diagonal peaks that are associated with the twelve HA-HB and HB-HA cross-peaks inside the two rectangles on the right. From these, the Serine-related cross-peaks in the HN-HA region were located, as enclosed by the rectangle on the left. Note that the peaks were picked from a NOESY spectrum, and some adjustments of the cross-peak locations were performed at a later stage.

The Serine-related cross-peaks in the HN-HA region are then found inside the rectangle on the left.

- Type rad to remove all lines and rectangles in the plot. Then, type dr to re-display the plot.

➢ Identification of Lys, Asp and Trp residues

The HE-related cross-peaks for the Lys residues and the HB-related cross-peaks for the Asp and Trp residues can be found within the two rectangle in Figure 10.12.

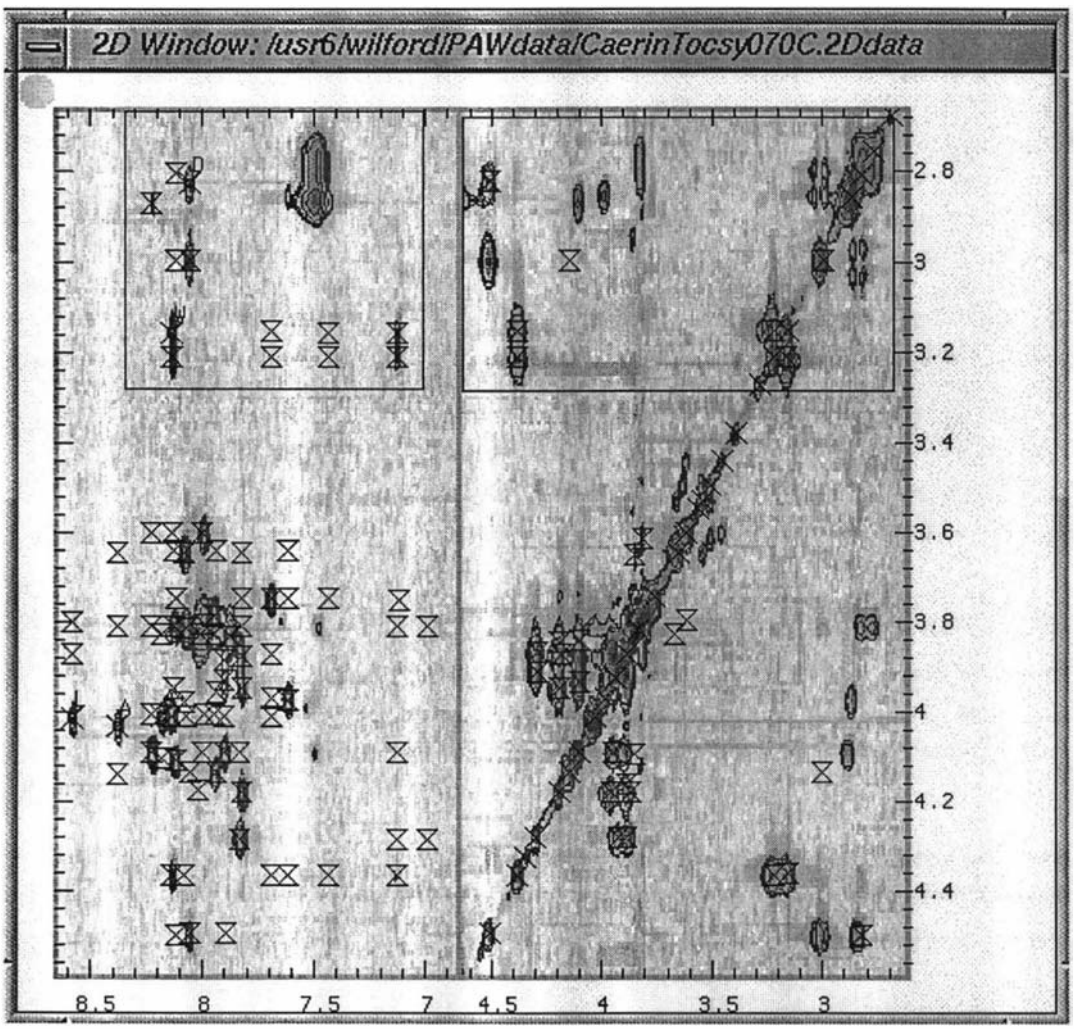


Figure 10.12 A multi-region plot that shows the HE-related cross-peaks for the Lys residues and the HB-related cross-peaks for the Asp and Trp residues.

The Trp-related cross-peaks are identified by their obvious connection to the aromatic cross-peaks, as shown in Figure 10.13.

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks centred at the three intersections on the left, as arrowed in Figure 10.13.

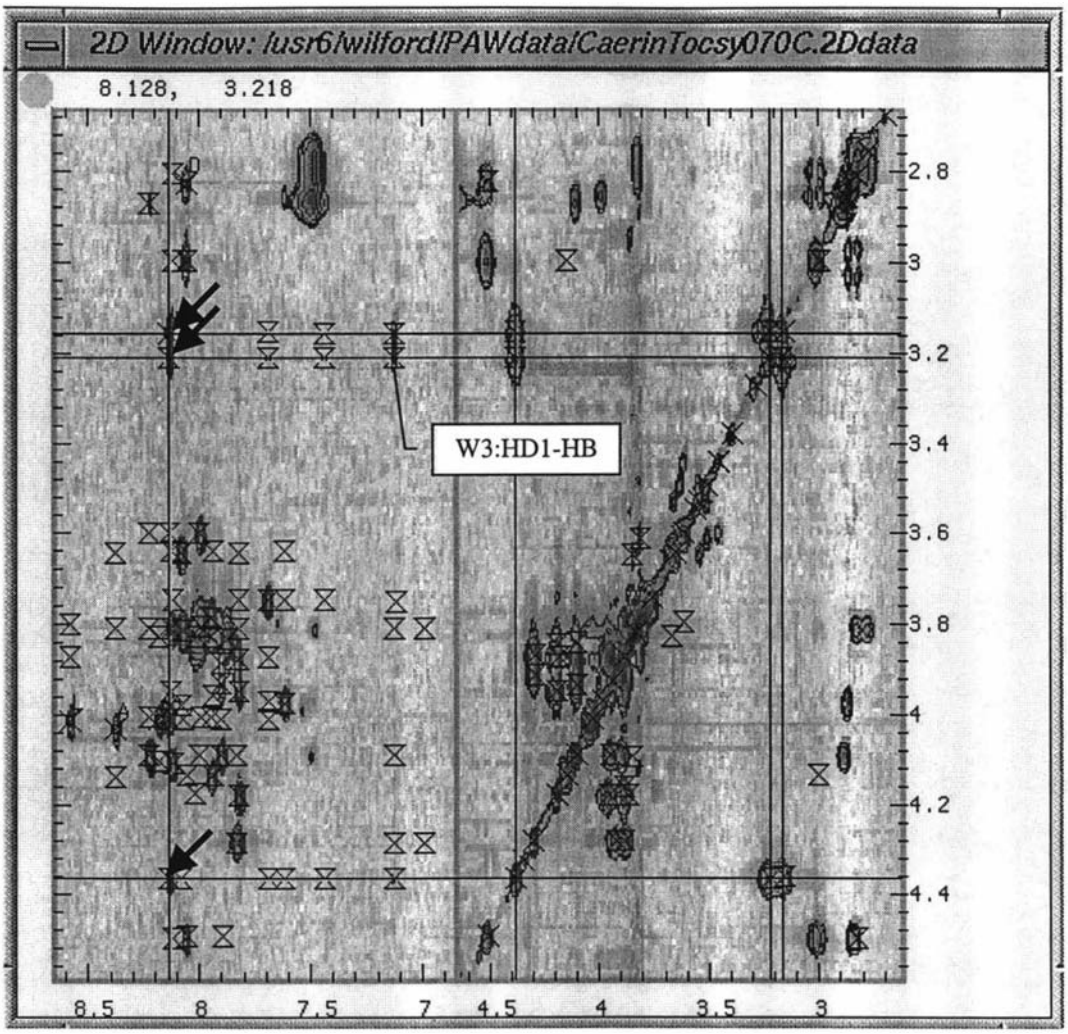


Figure 10.13 The same regions as the last plot. The crosshairs were drawn from the arrowed cross-peaks. The unique Trp residue was identified by the connection with the aromatic cross-peaks in Tryptophan.

The remaining two intense cross-peaks at the upper-left corner define the Asp-related cross-peaks, as shown in Figure 10.14, which was obtained as follows:

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in Figure 10.14.

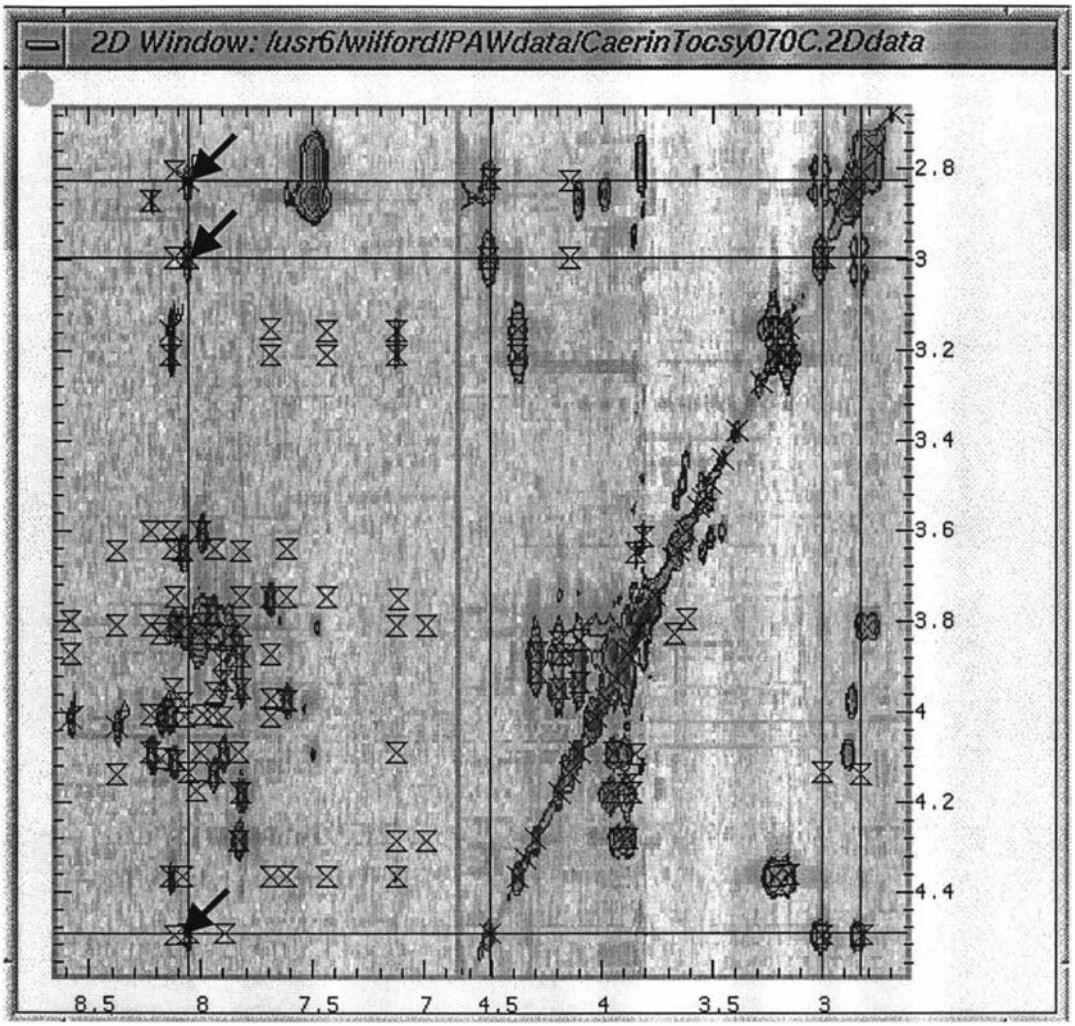


Figure 10.14 The same regions as the last plot. The crosshairs, which were drawn from the arrowed cross-peaks, enable the unique Asp residue to be identified, because the Asp cross peaks are much more intense than the Lys HN-HE cross-peaks and they have no connection with any aromatic cross-peaks.

The rest of the weak peaks in the upper area of the plot define three Lys residues, as shown in the next three figures, which were obtained as follows:

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in the figures.

Note that some cross-peaks have been picked by repeated use of the commands rp1 (for picking one raw peak) and tpf (for finding transposed-peak). The locations of the manually picked cross-peaks can easily be identified by drawing crosshairs from the strong peaks of the three spin-systems.

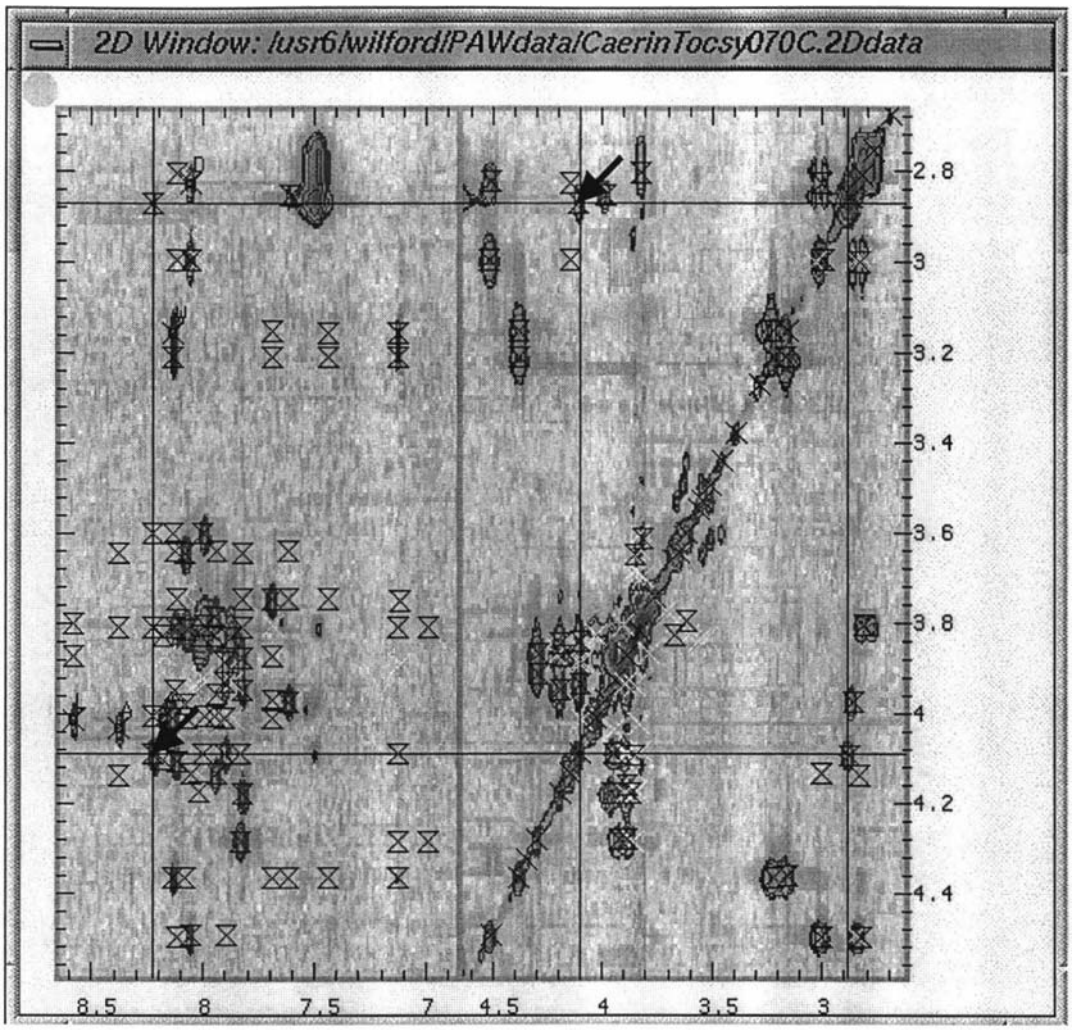


Figure 10.15 A multi-region plot that shows how one of the Lys spin-systems was identified. Some cross-peaks in the plots were picked manually, as described in Chapter 10 of Volume II. The crosshairs were drawn from the arrowed cross-peaks.

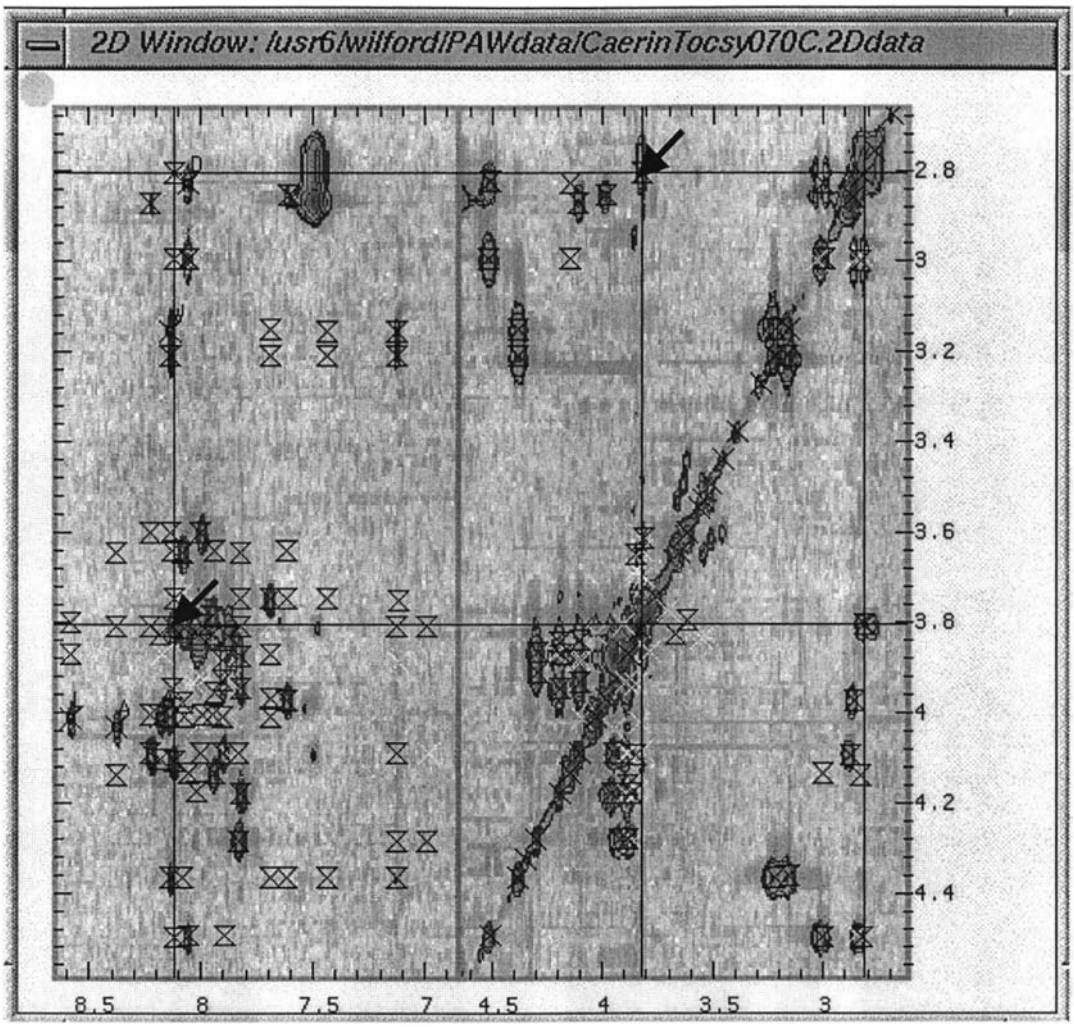


Figure 10.16 A multi-region plot that shows how the second Lys spin-system was identified. The crosshairs were drawn from the arrowed cross-peaks.

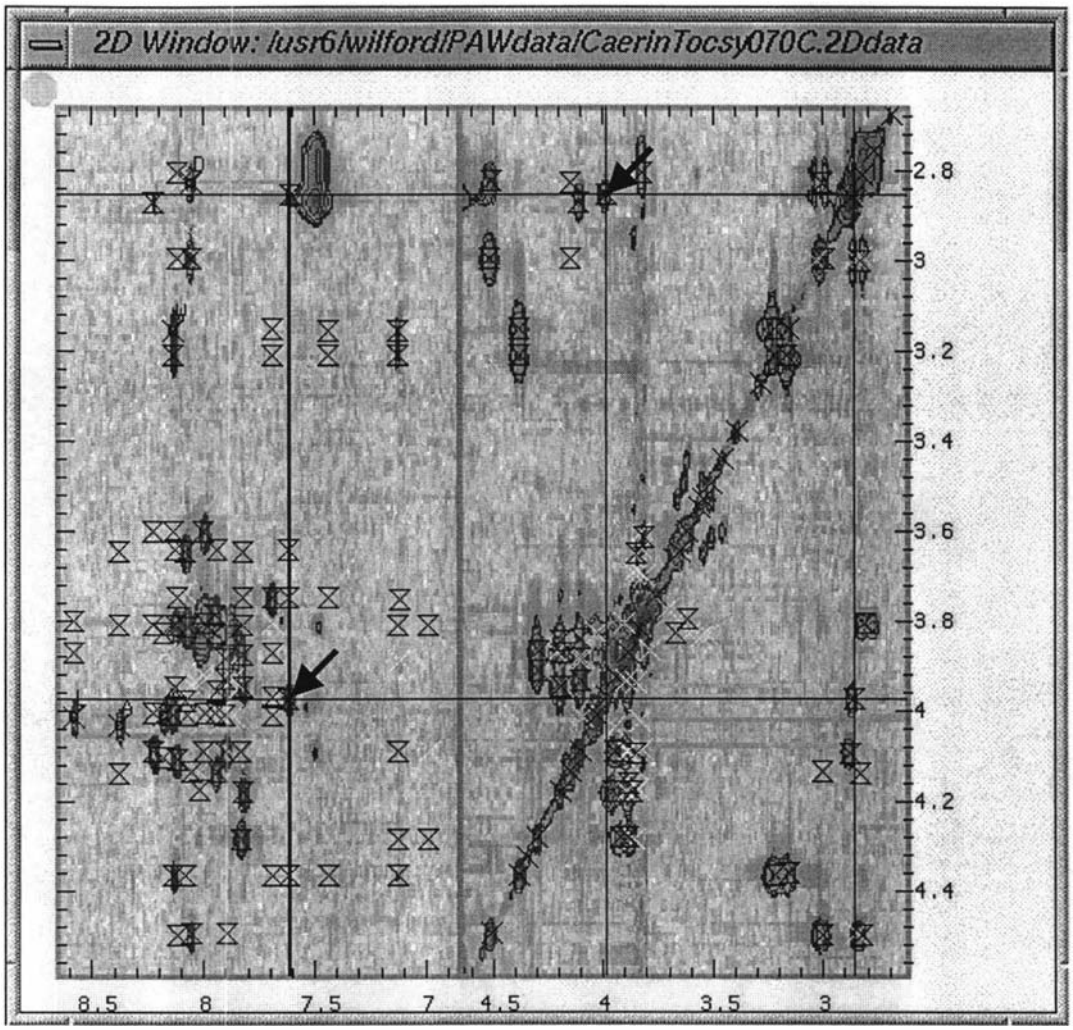


Figure 10.17 A multi-region plot that shows how the third Lys spin-system was identified. The crosshairs were drawn from the arrowed cross-peaks.

➢ Identification of Gly residues

Only two out of four Gly spin-systems can be unambiguously identified at this stage, as shown in the next figure, which was obtained as follows:

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in the figures.

No further correlation peaks can be found for the two HN-HA cross-peaks in the upper area of the fingerprint region, giving a very good identification of them. The HA1 and HA2 of all the Gly spin-systems, however, cannot be resolved in any of the spectra. (Identification of the remaining two Gly residues was only possible via the sequential-specific assignment.)

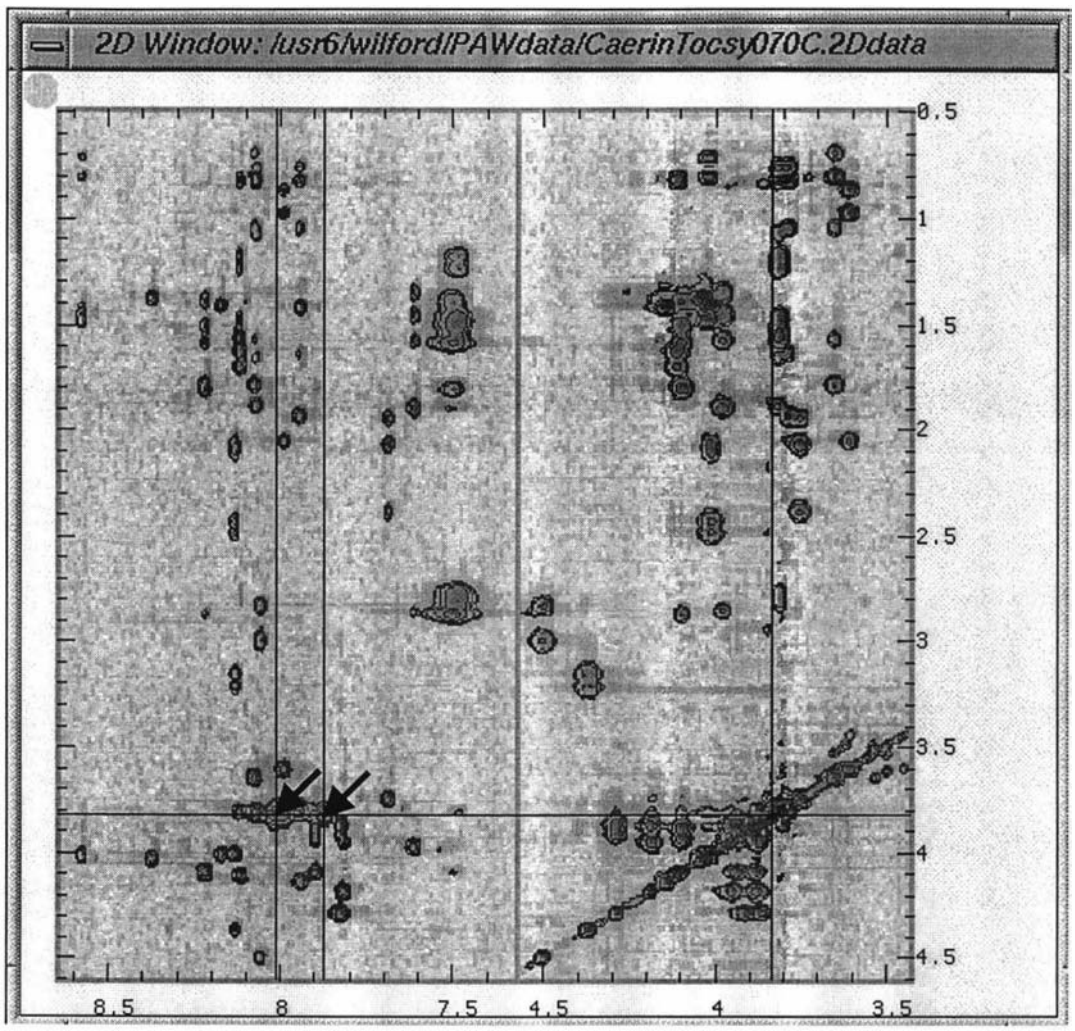


Figure 10.18 A multi-region plot that shows how two of the Gly spin-systems were identified. The crosshairs were drawn from the arrowed cross-peaks.

➢ Identification of Val, Glu and Gln residues

Each of these spin-systems only appears once in the Caerin 4.1 sequence. The Val spin-system is identified by its unique peak-distribution pattern in the TOCSY spectrum (Figure 10.19). The Gln system is identified by its connection to the unique HE2 protons (Figure 10.20). The Glu system is identified by its similar peak-distribution pattern with Gln (Figure 10.21). The next three figures were obtained as follows:

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in the figures.

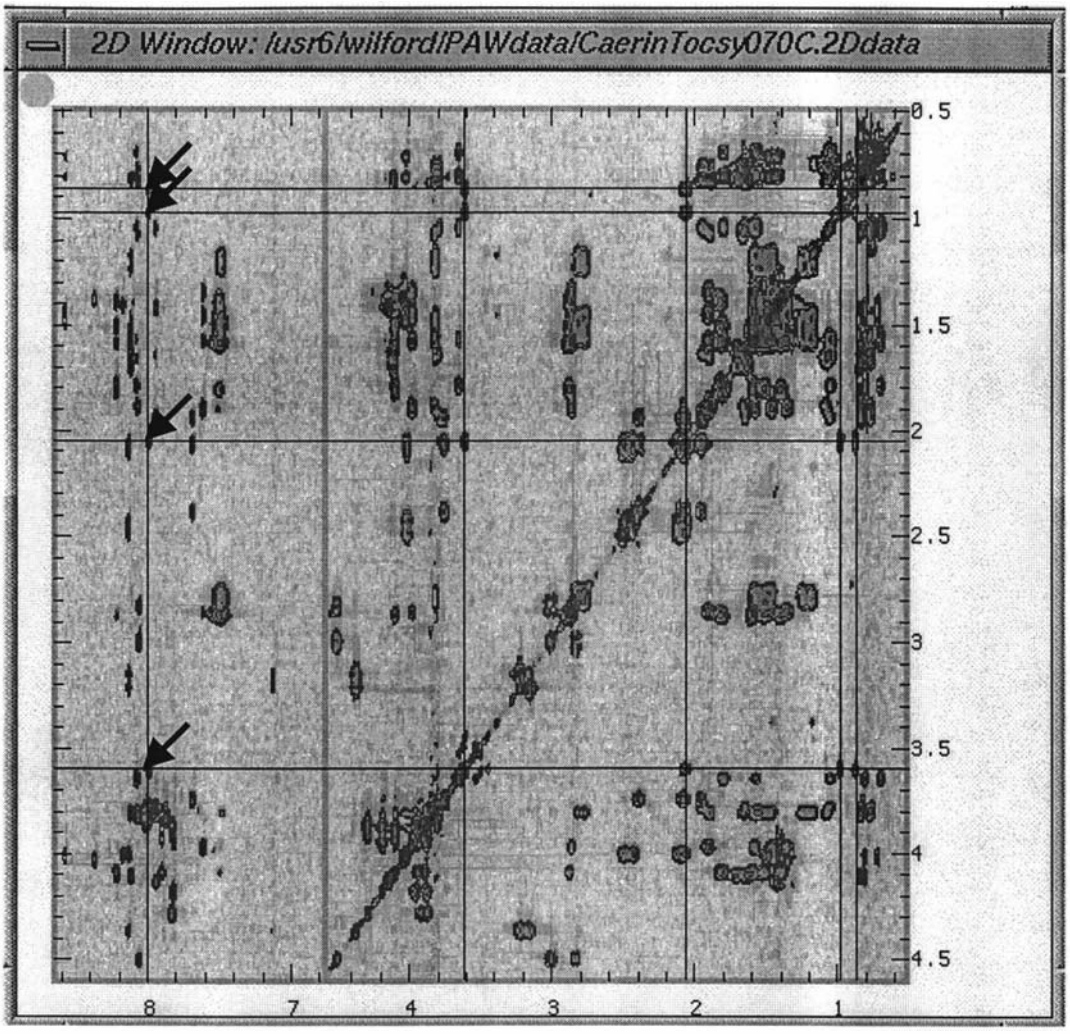


Figure 10.19 A multi-region plot that shows how the Val spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

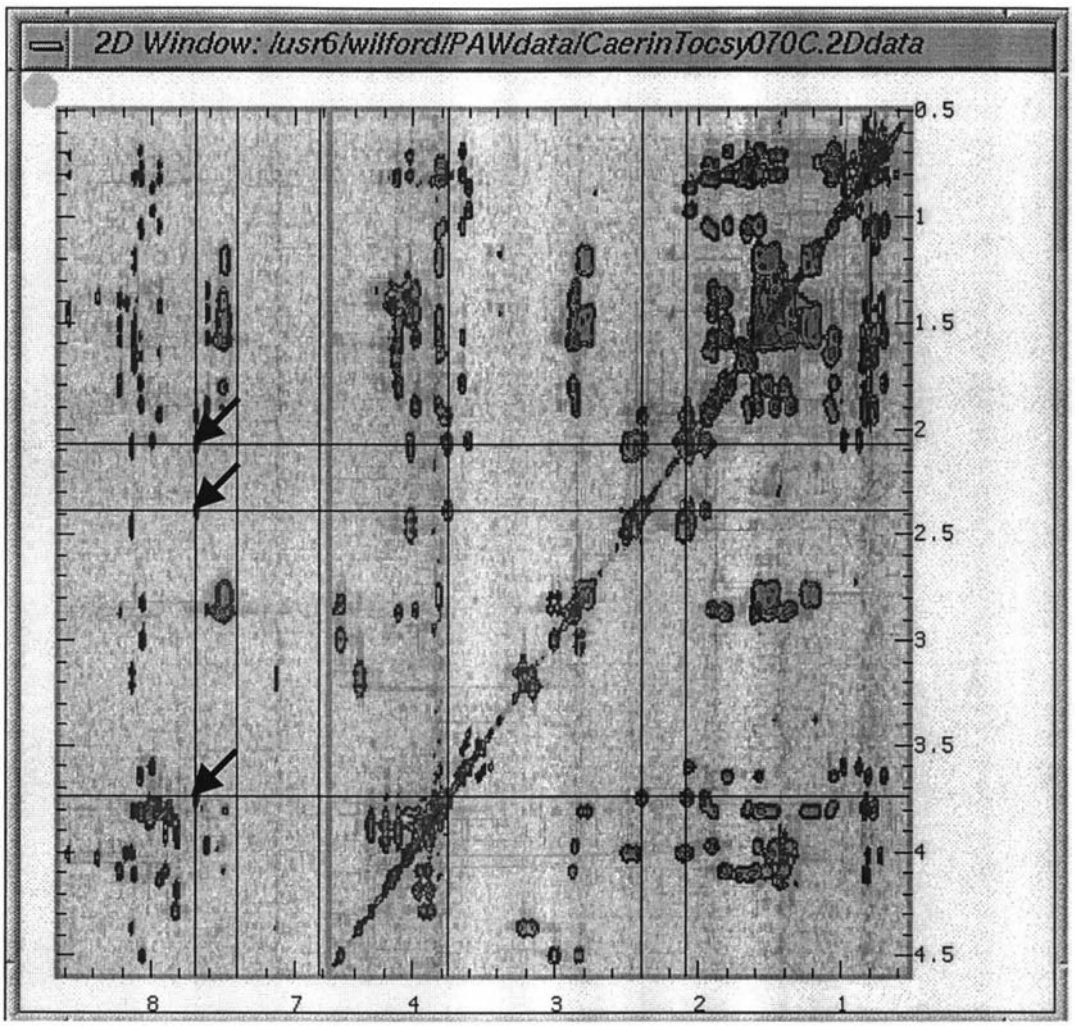


Figure 10.20 A multi-region plot that shows how the Gln spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

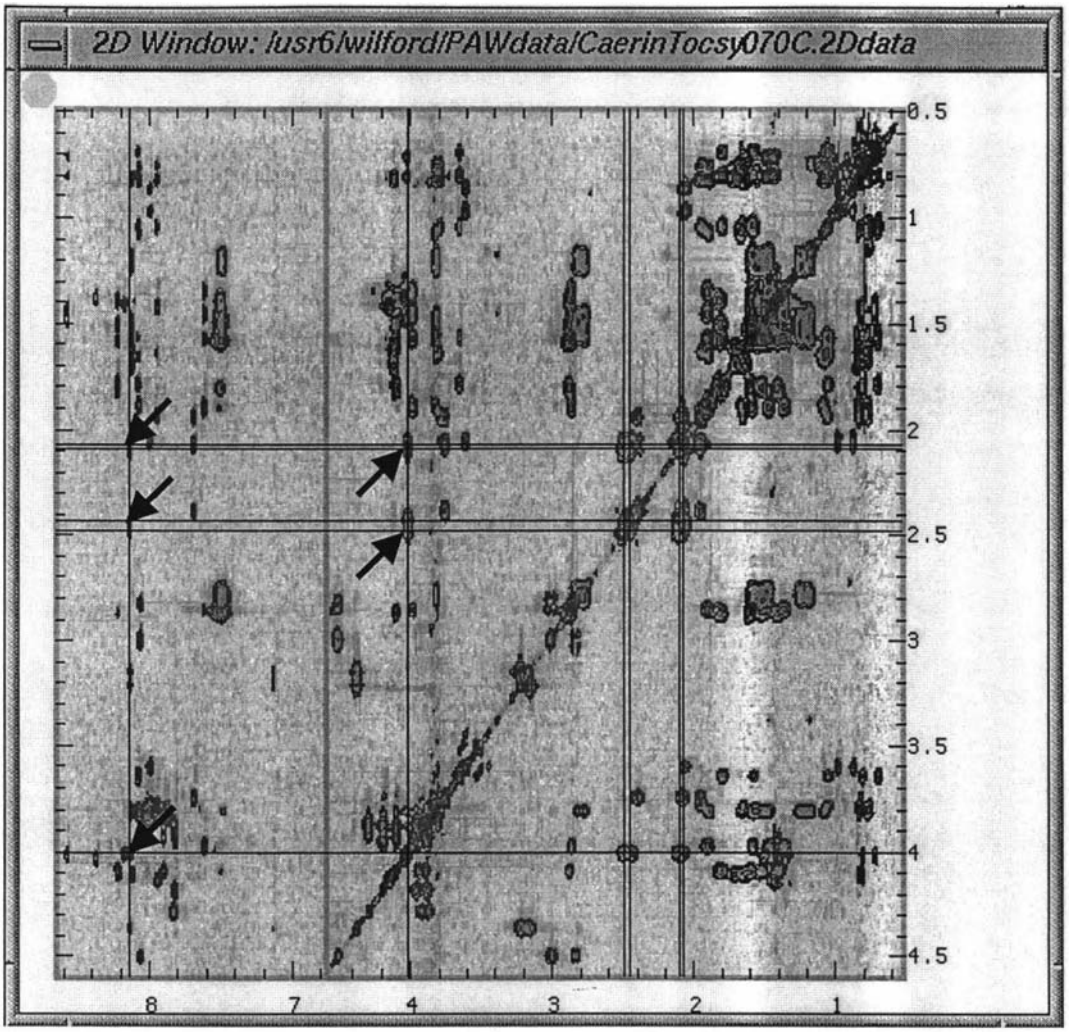


Figure 10.21 A multi-region plot that shows how the Glu spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

➢ Identification of Leu and Ilu residues

The two Leu residues in the sequence are identified from the remaining unidentified spin-systems by looking at their heavily overlapped peak-pattern that contains HB and HC (see the next two figures).

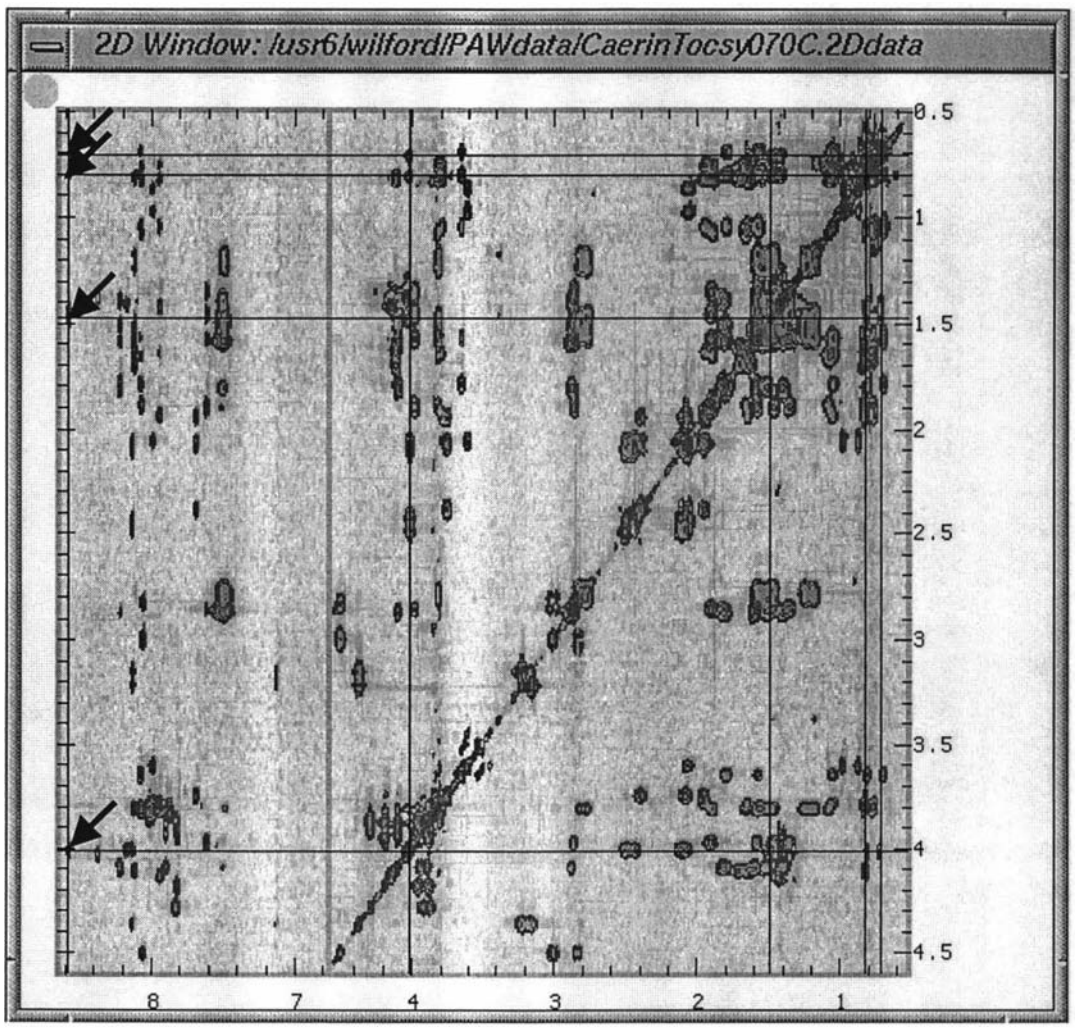


Figure 10.22 A multi-region plot that shows how one of the two Leu spin-systems in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

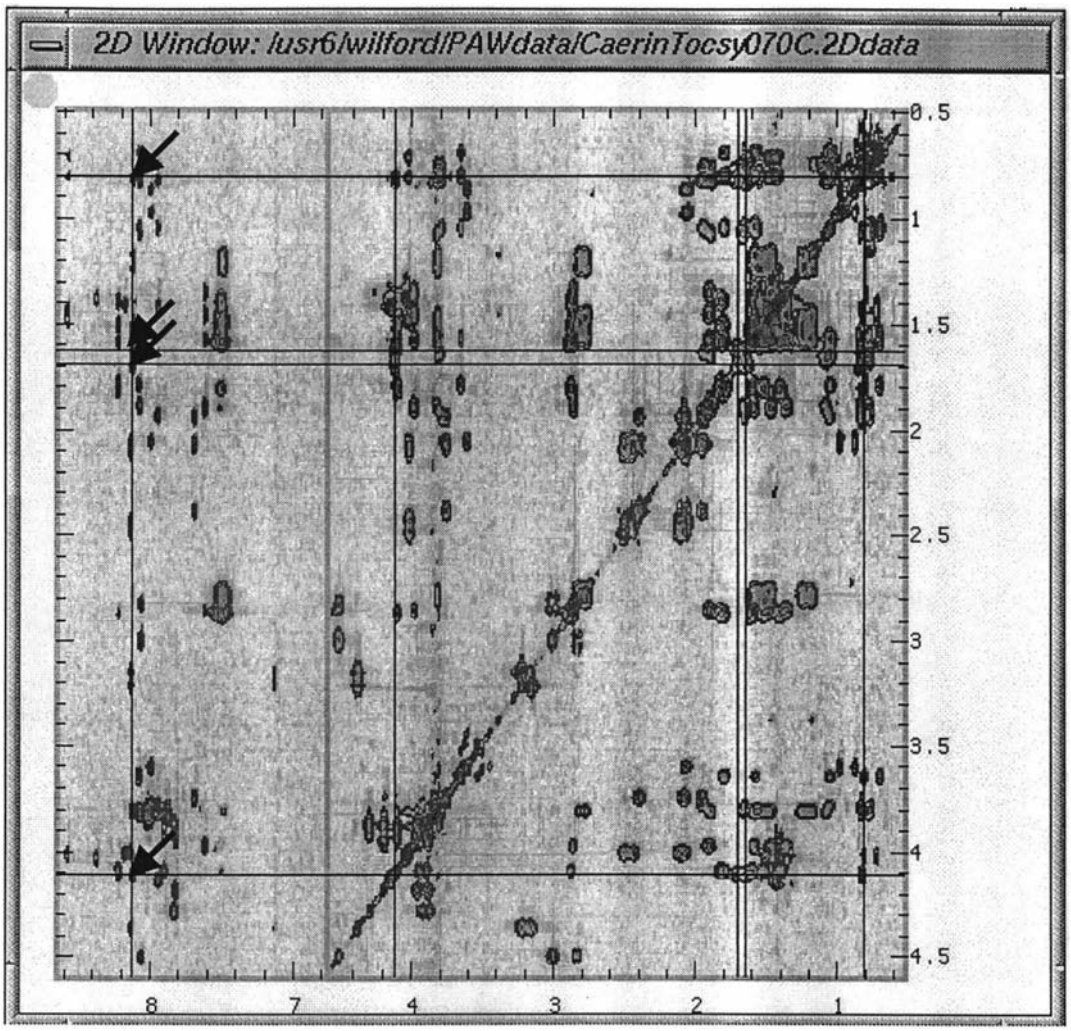


Figure 10.23 A multi-region plot that shows how the second Leu spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

The three Ilu are then unambiguously identified by their long side-chain structure, as shown in a typical example in Figure 10.24.

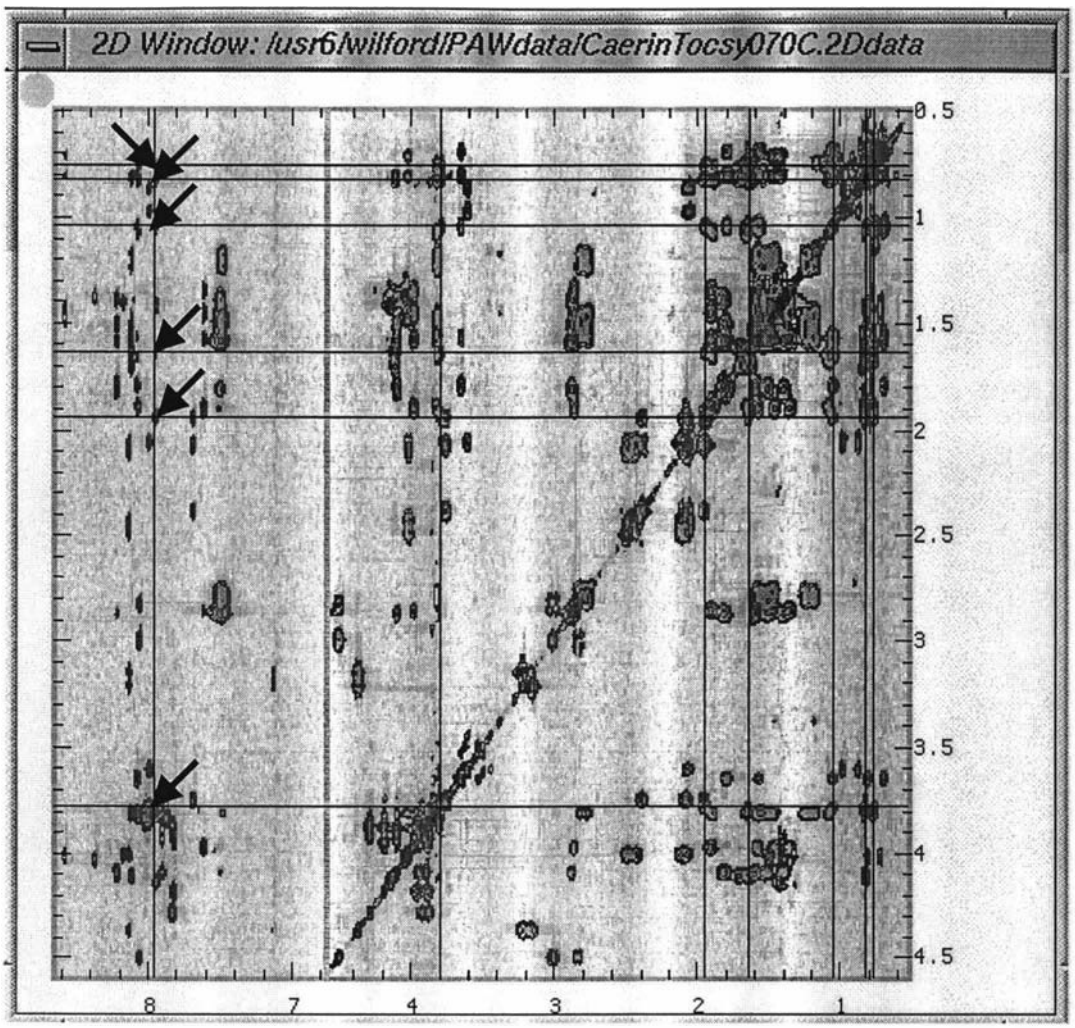


Figure 10.24 A multi-region plot that shows how one of the Ile spin-system in Caerin 4.1 was identified. The crosshairs were drawn from the arrowed cross-peaks.

Again, the figures in the subsection were obtained using the method described in previous subsection.

Chapter 11:

Peak-assignment Process for the Caerin 4.1 NMR Spectra

11.1 Introduction

This chapter describes the peak-assignment operations for the Caerin 4.1 NMR spectra. Again, the high-resolution spectra and refined peak lists obtained by the methods described in Chapter 8 and Chapter 9 were used in the process, namely:

- **CaerinNoesy150C.2Ddata**
- **CaerinTocsy150C.2Ddata**
- **CaerinCosyC.2Ddata** (optional)
- **Caerin98.RPeaks**
- **Caerin98.DPeaks**
- **Caerin98.CPeaks**

To avoid unexpected errors, it is again recommended that PAW is restarted and the workbench used in Chapter 7 is again loaded.

➢ The *Peak-display Toolbox* and the *Peak-assignment Toolbox*

The display-mode switches and command-buttons that are frequently used during the peak-assignment process can be found in the *Peak-display Toolbox* and the *Peak-assignment Toolbox*, as shown in Figure 11.1.

To open the *Peak-display Toolbox*, either type opd or choose [PkDsplTBox] from the *Process Menu*. To close the toolbox, either type xpd or double-click on the window-control button at its top-left corner.

To open the *Peak-assignment Toolbox*, either type opa or choose [PkAssgnTBox] from the *Process Menu*. To close the toolbox, either type xpa or double-click on the window-control button at its top-left corner.

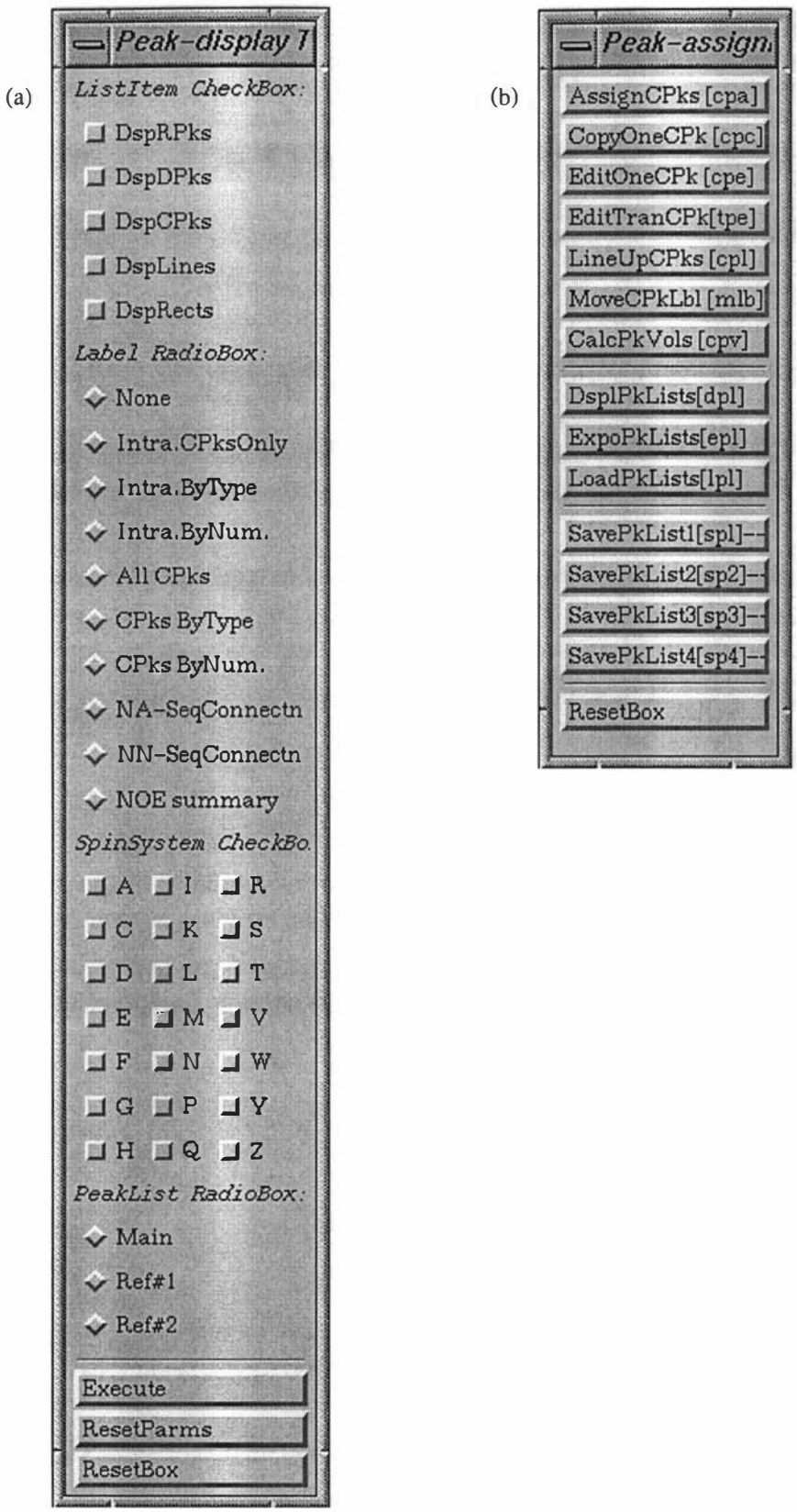


Figure 11.1 (a) The Peak-display Toolbox. (b) The Peak-assignment Toolbox.

➢ **The cross-peak symbols**

PAW uses different symbols to indicate the completeness of the cross-peak assignment. The shape of a cross-peak symbol depends on how many fields have been assigned for the cross-peak. These fields are the amino-acid codes, residue numbers and proton codes in D1 and D2.

In short the rules are:

- Unassigned cross-peaks are indicated by symbols like hour-glasses (see Figure 11.4).
- Partially assigned cross-peaks are indicated by differently shaped small rectangles.
- A small square indicates one field is assigned in each direction (see Figure 11.6). A medium-sized square indicates two fields are assigned in each direction.
- A vertical oblong indicats the number of fields assigned in D1 is more than that in D2. Likewise, A horizontal oblong indicates the number of fields assigned in D2 is more than that in D1.
- Fully assigned cross-peaks are displayed with the biggest squares.

11.2 Preliminary

The preliminary steps for starting PAW, opening the workbench, and loading the spectra as well as peak lists have been described in Chapter 7 to 10.

- Define seven zoomed-patterns as shown in Figure 11.2. (Zm#6 is perhaps too small to be clearly seen in the view window and will be defined later.)

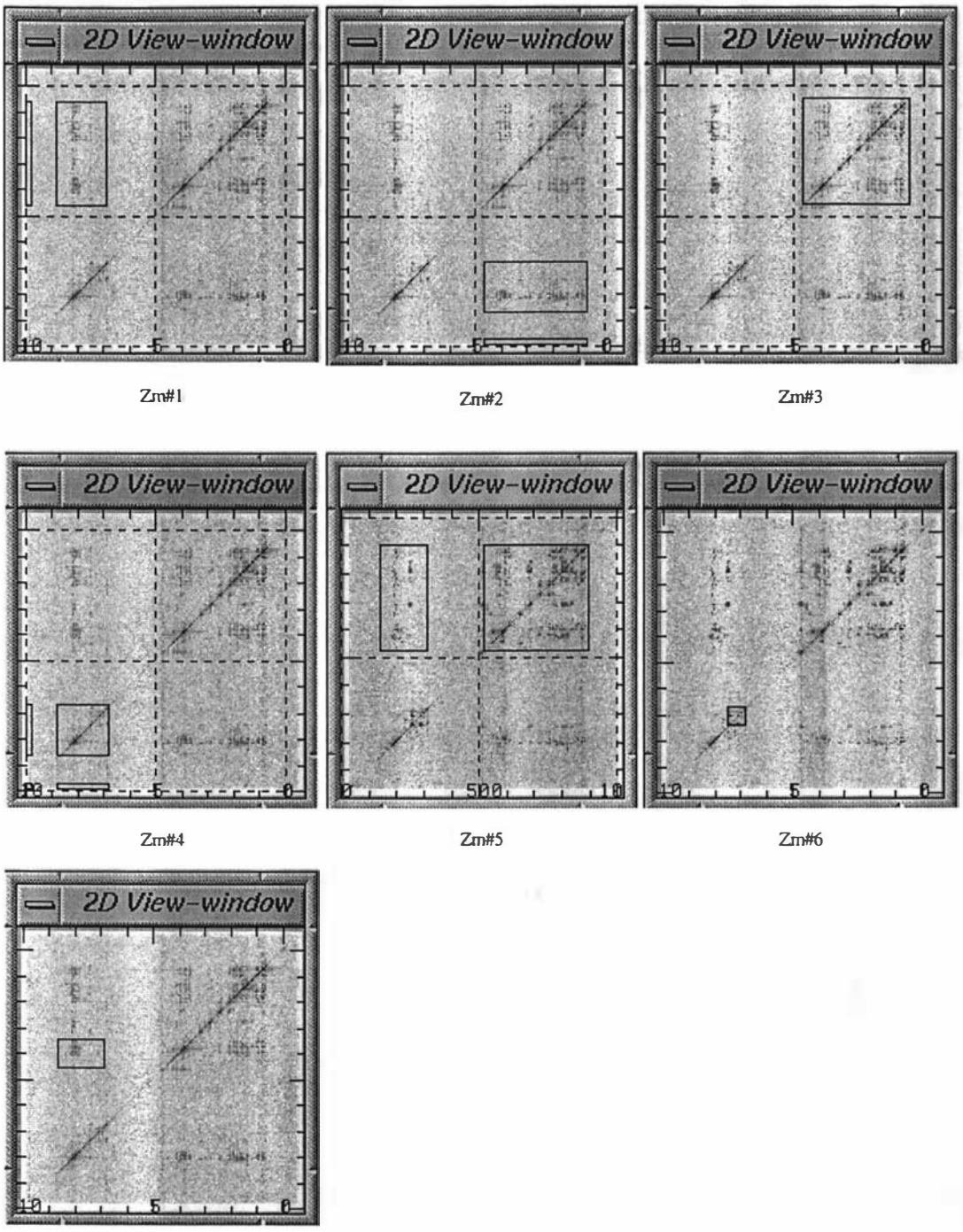


Figure 11.2 The seven zooming patterns used during the peak-assignment process for the Caerin 4.1 NMR spectra.

11.3 Assigning Cross-peak Proton-codes in Groups

At the initial stage, it is wise to concentrate on the peaks that are picked with the auto-pick routine, and to ignore the ambiguous ones. More raw peaks and cross-peaks can be added interactively if their existence is obvious.

➢ Assigning cross-peak proton-codes in the lower-left region (Zm#4)

This process is done using the DQF-COSY spectrum as follows:

- Click on the title-bar of the DQF-COSY window to bring it forward, and then click in the plot area.
- With the cursor in the plot area, type z4 to display Zm#4 with the peak symbols.

Figure 11.3 shows the DQF-COSY spectrum, where the transposed rectangles are drawn from three W3-related peaks (arrowed, see Chapter 10).

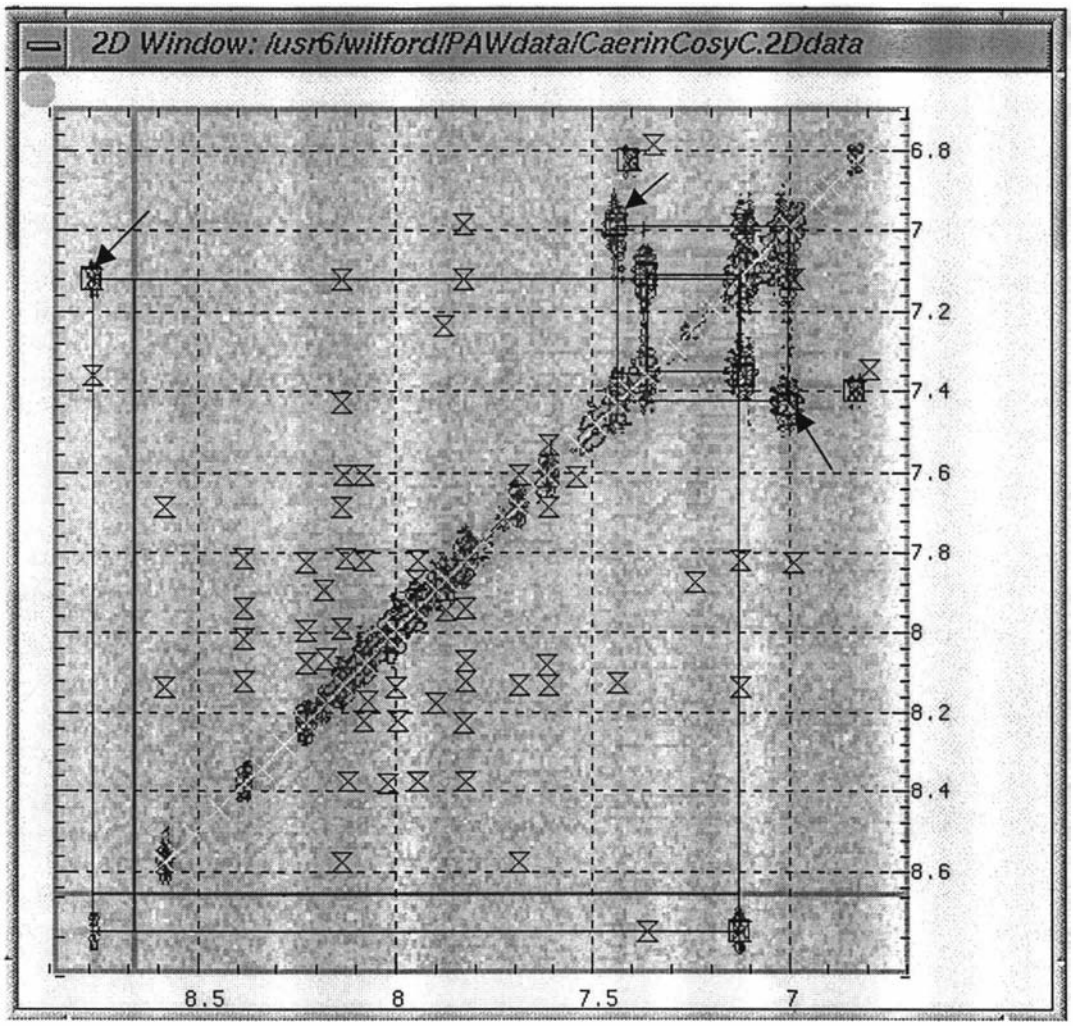


Figure 11.3 The lower-left regions of the Caerin 4.1 DQF-COSY spectrum, where the transposed rectangles have been drawn from the centres of three W3-related peaks.

By referencing Table 7.2, a few clues can be obtained from Zm#4 of the DQF-COSY spectrum (Figure 11.3):

1. It is known that W3:HD1 and W3:HE1 form a unique two-spin system in the Caerin 4.1 DQF-COSY and that W3:HE1 has the highest chemical shift. Therefore, the four NOESY cross-peaks at the upper-left and lower-right regions must be related to the W3:HE1. Among them, the two cross-peaks at the corners of the largest rectangle are associated with W3:HD1, because underneath them there are DQF-COSY cross-peaks.
2. The peaks linked to the diagonal peaks at [7.43,7.43], [7.36,7.36], [7.12,7.12] and [5.99,6.99] are associated with W3:HE3, W3:HZ3, W3:HH2, and W3:HZ3, respectively. These are the cross-peaks at the corners of the other two transposed rectangles.
3. The two peaks that are linked to the diagonal peak at around [6.83,6.83] are associated with Q4:HE2, as shown in the two small rectangles on the top-right corner.

The rest of the peaks, as shown inside the three rectangles in the next plot, can be assigned as follows:

1. The peaks that are enclosed inside the largest rectangle are the HN:HN cross-peaks.
2. All the other peaks that are enclosed inside the other two rectangles are related to either W3:HE3, W3:HZ3, W3:HH2, W3:HZ3 or X24:HN, where X24 is for the terminator HN2.

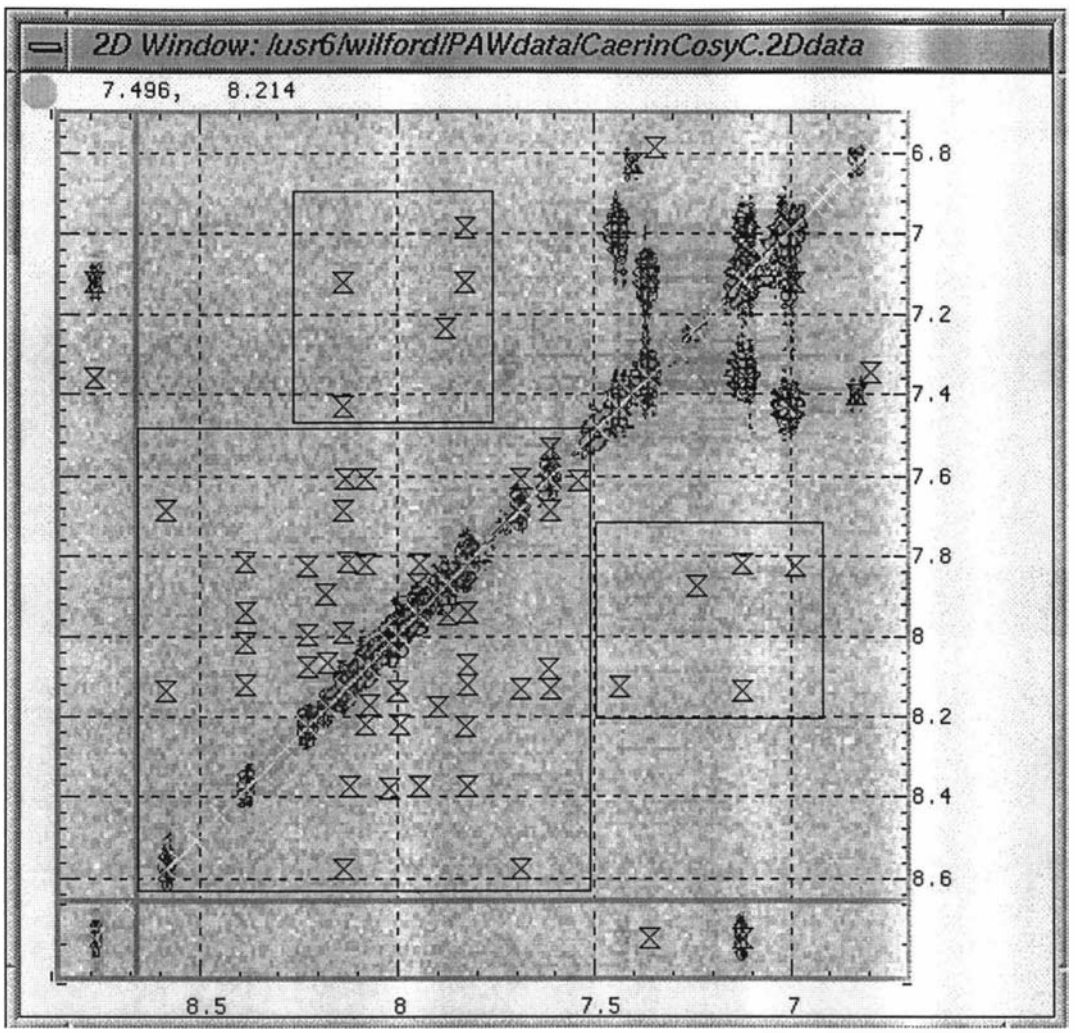
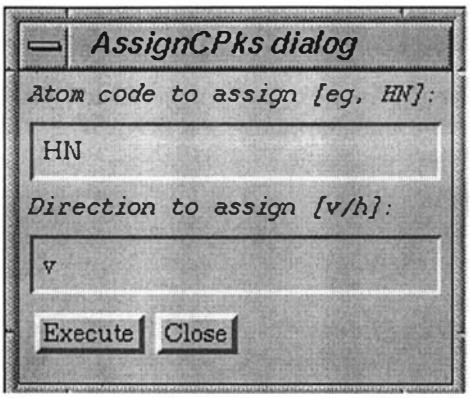


Figure 11.4 Zm#4 of **CaerinCosyC** spectrum. See the text above for explanation of the peaks in this plot.

These clues led to the following examples of assignment operations:

- Choose [AssignCPks] in the *Peak-assignment Toolbox*. The message 'Use button 1 to set a region for group assignment.' will appear on top of the plot.
- Select a region in the plot to enclose all peaks with their D1 chemical-shift from 7.5 ppm to 8.7 ppm. The *AssignCPks Dialog* will appear (Figure 11.5).



- In the dialog, enter HN for the item code to be assigned, and v for the direction to be assigned. Then, choose [Execute].
- Repeat the last two steps to assign HN in D2. This time the entry for the direction to be assigned must be h.
- Type <dr> to display the assigned peak symbols (Figure 11.6).

Figure 11.5 The AssignCPks Dialog.

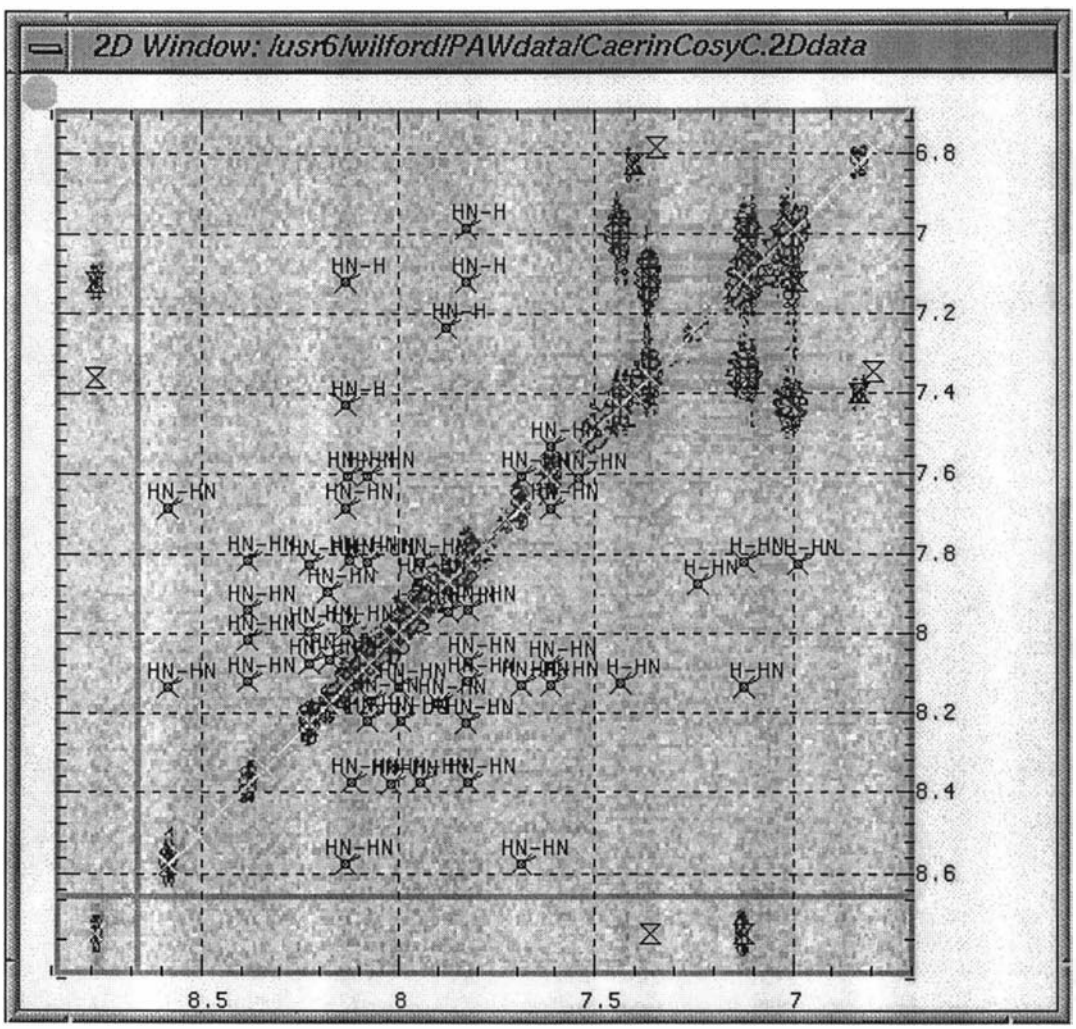


Figure 11.6 The cross-peaks assigned as HN-HN in Zm#4.

- Type spl to save the peak list, preferably with a different name.
- Repeat the operations above to assign all other peaks in Zm#4 according to the clues mentioned in Chapter 10.

- Click in the plot with the *MsBtn#3* to refresh the display if necessary.

Figure 11.7 shows the results of the above operations. All the proton codes in Zm#4 are assigned except two that are not linked with any of the significant diagonal peaks, as shown at the two corners of the transposed-rectangle in the middle.

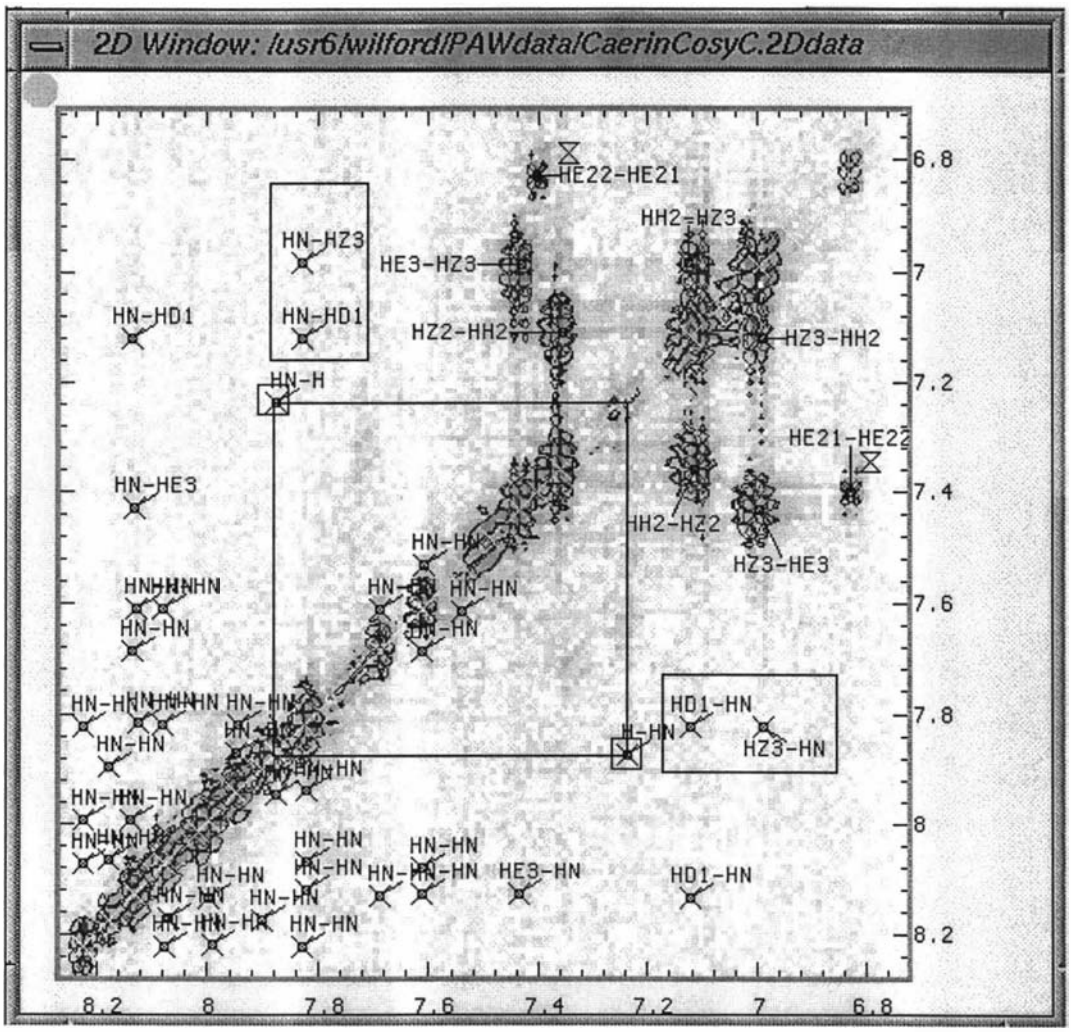


Figure 11.7 Other cross peaks assigned according to the clues mentioned in chapter 10.

Note:

- There is an ambiguity regarding the assignment of the W3:HD1-related peaks because their chemical shifts in D1 are almost identical with that of the W3:HH2. This is resolved by considering the fact that, within the Trp-residue, HH2 and HZ2 are far away from the other protons beyond the ring, and hence their connectivities with others is less likely to be seen in the NOESY spectrum.

- Because of overlap, there is not enough evidence to identify X24-related peaks at this stage. Four of the assignments, as enclosed by the two medium rectangles in the last figure, were in fact wrong and will be corrected later. This is to be expected in any assignment process.

➢ Assigning cross-peak proton-codes in the upper-left region (Zm#1)

This process is best done on the TOCSY spectrum, in which only intra-residue cross-peaks that correlate protons in the same spin-system occur.

- Click on the title-bar of the TOCSY window to bring it forward, and then click in the plot area.
- With the cursor on the plot area, type z1 to draw the peak symbols onto the plot (Figure 11.8).

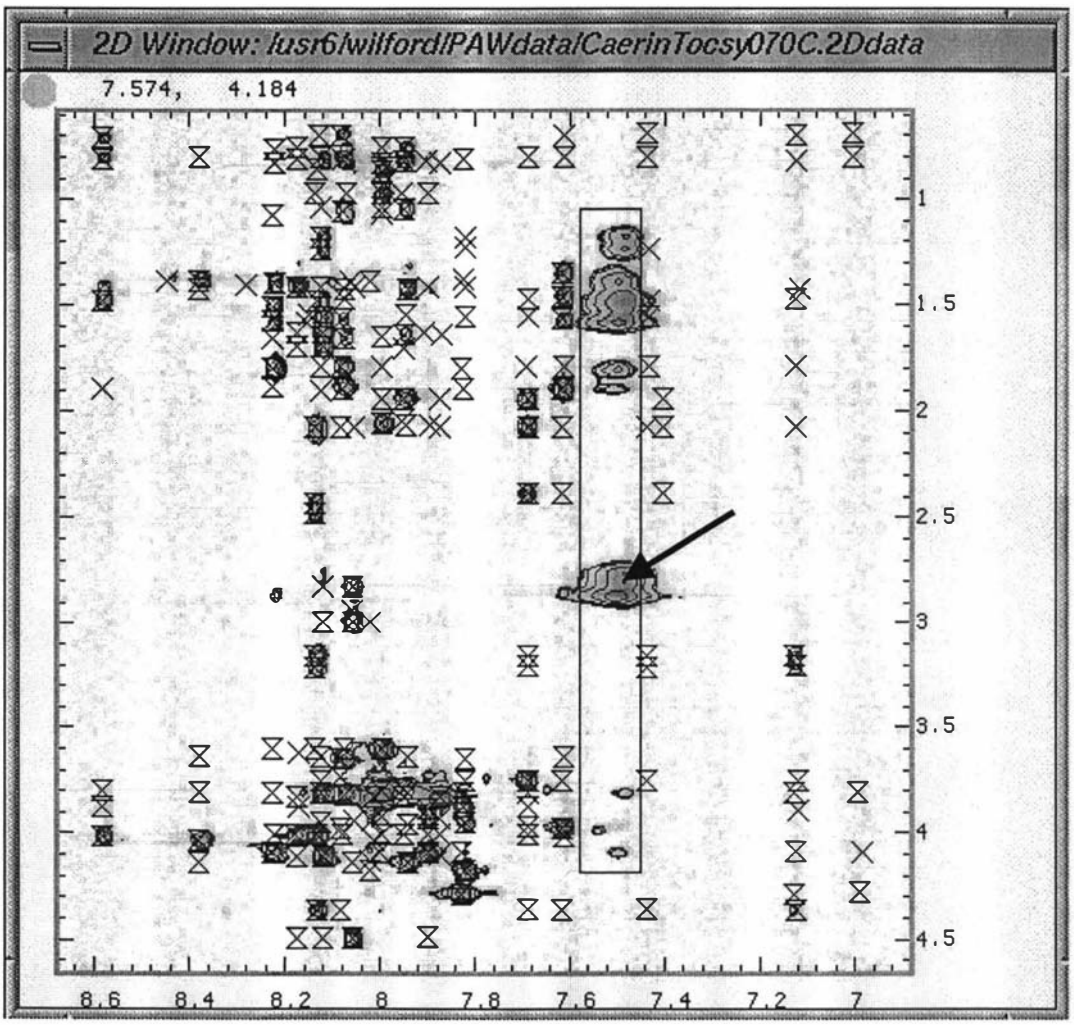


Figure 11.8 The upper-left region of the Caerin 4.1 TOCSY070 spectrum with the NOESY cross-peaks superimposed.

As can be seen, the NOESY cross-peaks fall into two categories. The ones that are associated with the intra-residue connectivities are overlapped on top of the TOCSY peaks, whereas those associated with the inter-residue connectivities are not.

One useful fact observed from Figure 11.8 is that those peaks enclosed inside the rectangle at around 7.5 ppm in D1 are not related to any of the NOESY peaks. These are the HZ3-related peaks of the Lys residues in the protein. Also, only the HZ3-HE (arrowed) cross-peaks are seen in the DQF-COSY spectrum.

The operations to assign the proton codes for peaks in Zm#1 are as follows.

- Choose [AssignCPks] in the *Peak-assignment Toolbox.* The message '*Use button 1 to set a region for group assignment.*' will appear on top of the plot.
- Select a region in the plot to enclose all peaks with D1 chemical-shift from 7.5 ppm to 8.7 ppm.
- In the dialog, enter HN for the item code to be assigned, and v for the direction to be assigned. Then, choose [Execute].
- Repeat the above steps to assign the proton-codes for the rest of the peaks on the right according to the peaks assigned in Zm#4.
- Type dr to draw the peak symbols onto the plot (Figure 11.9).

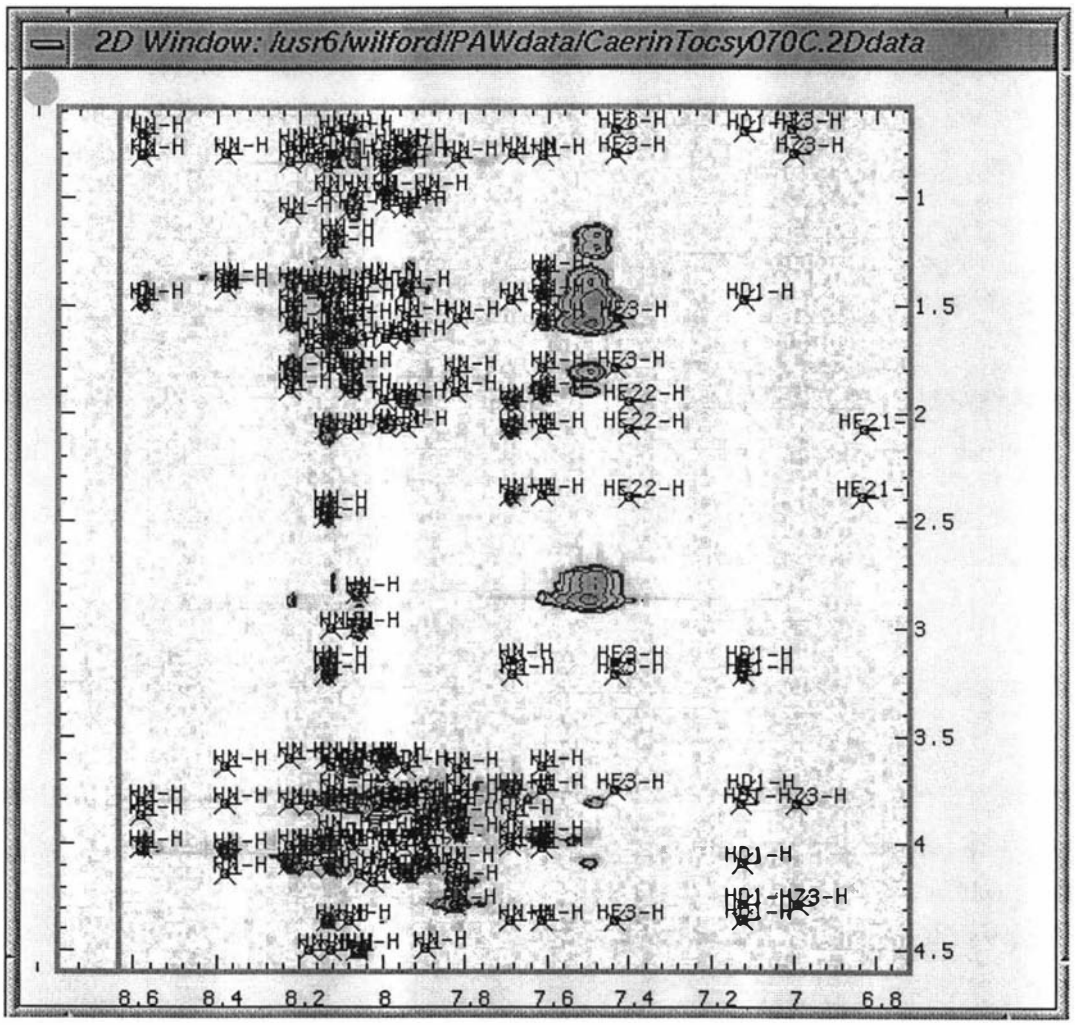


Figure 11.9 The group assignment result for all proton codes along D1 in the upper-left region of the Caerin 4.1 TOCSY070 spectrum.

- Assign all peaks in Zm#1 that have chemical shifts in D2 larger than 3.6 ppm to HA (Figure 11.10), despite the fact that a small fraction of them are HB.

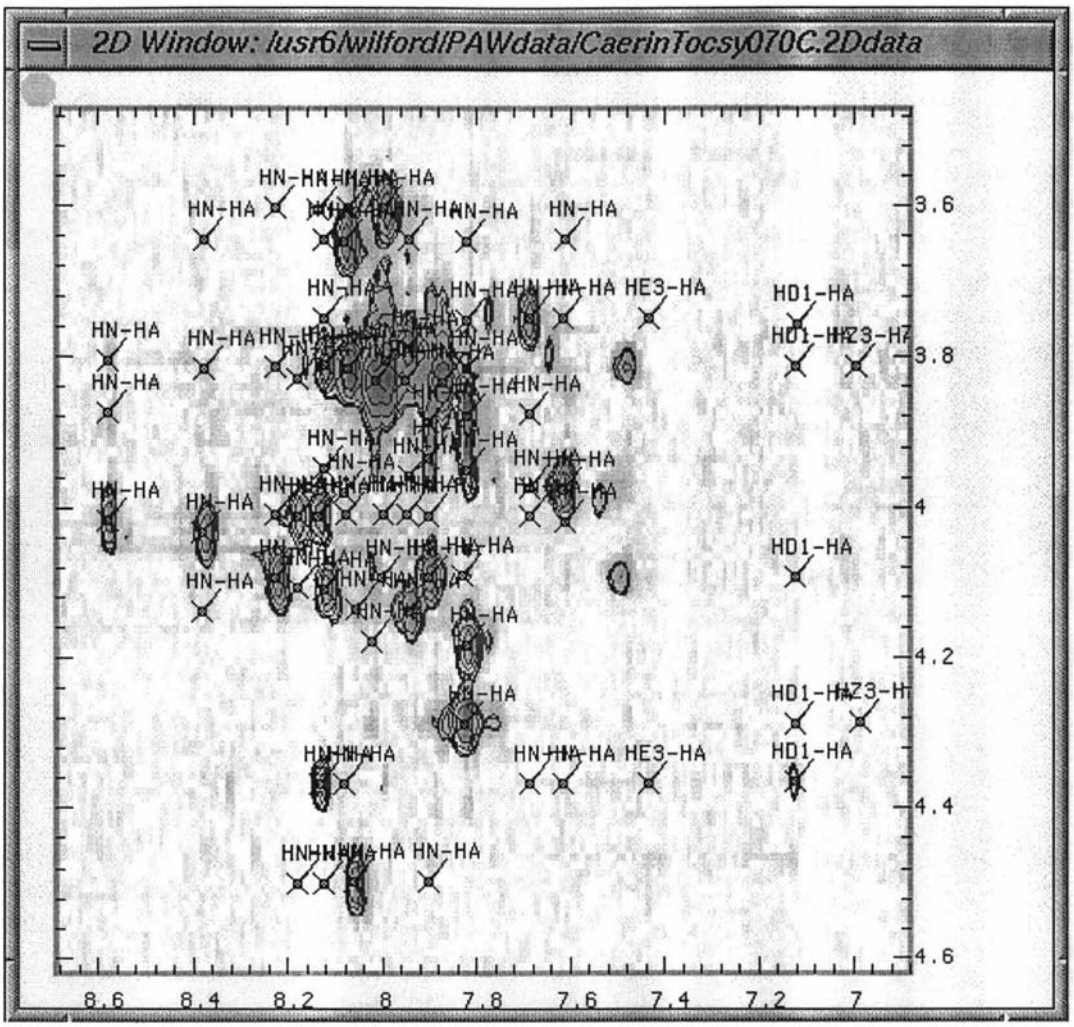


Figure 11.10 The group assignment result for all proton codes along D2 in the lower part of the upper-left region of the Caerin 4.1 TOCSY070 spectrum.

The proton codes in D2 for the remaining peaks are still unclear at this stage, and can only be assigned later.

➢ Assigning cross-peak proton-codes in the lower-right region (Zm#2)

The assignment of cross-peak proton-codes in the lower-right region is optional and can be done in a similar way to that for Zm#1. Perhaps, it is better to leave this area unassigned to keep a 'clean' reference in the assignment process.

For the Caerin 4.1 spectra, there is another reason why the assignment should not be done: some peaks appeared to be split, due to better resolution in D1.

➢ Assigning cross-peak proton-codes in the upper-right region (Zm#3)

This process can again be more easily done with the TOCSY spectrum. Because of the higher peak-intensities in this region, it is better to display the spectrum with a higher *First-level Threshold* for the contour plot. The operations for the assignment process are as follows:

- Click on the title-bar of the TOCSY window to bring it forward, and then click in the plot area.
- With the cursor on the plot area, type z3 to draw the peak symbols onto the plot (Figure 11.11).
- Choose [DefContrs] in the *2D-display Toolbox* and change the entry for the *First-level Threshold* to 28.
- Click in the plot with the *MsBtn#3* to refresh the display if necessary. The TOCSY Zm#3 plot will be displayed with the NOESY cross-peak symbols.

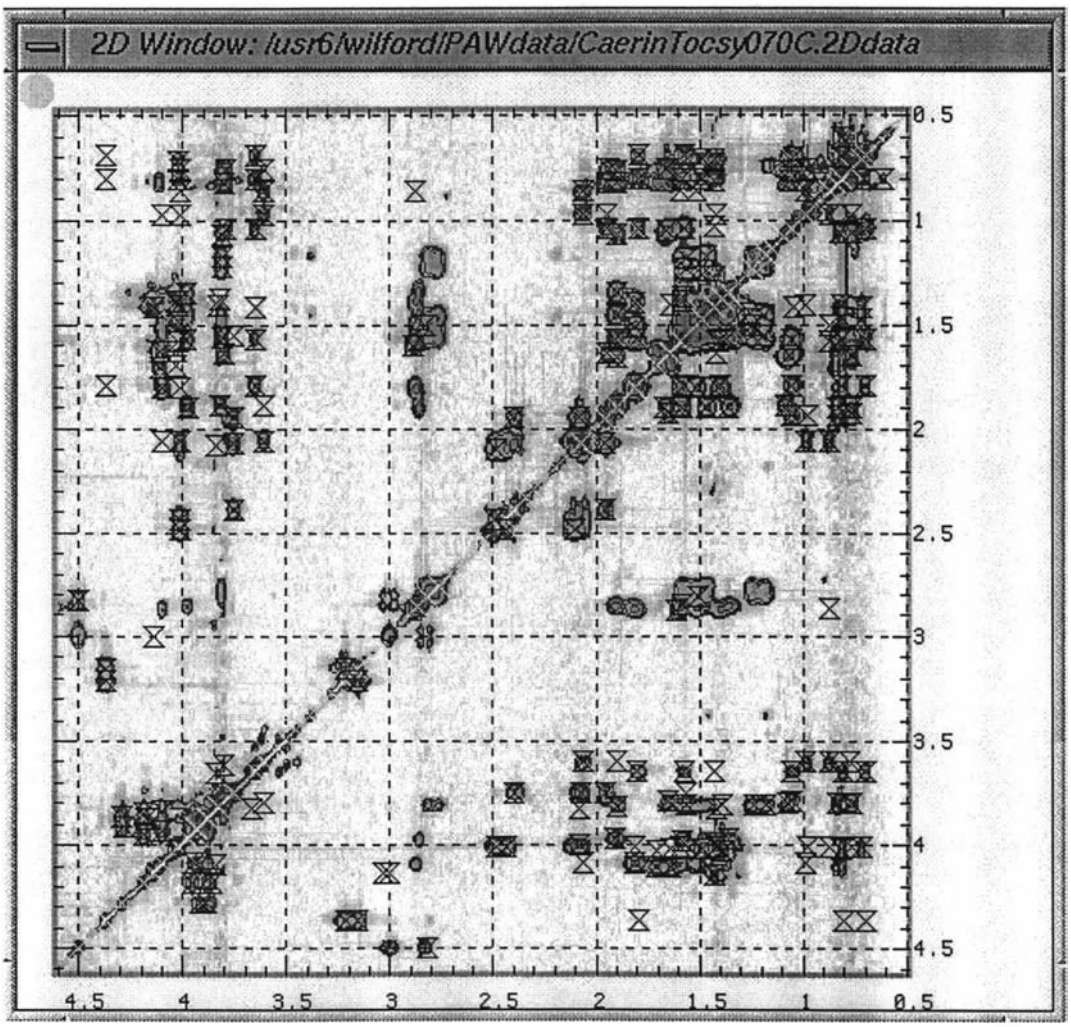


Figure 11.11 The upper-right region of the Caerin 4.1 TOCSY070 spectrum, with the NOESY cross-peak symbols superimposed.

At this stage, nothing much can be done in Zm#3 except for the group assignment of HA for all peaks that have chemical shifts larger than 3.6 ppm, despite the fact that some of them are HB. The operations result in the plot shown in Figure 11.12. (Note that the assignment of peaks below the diagonal is optional.)

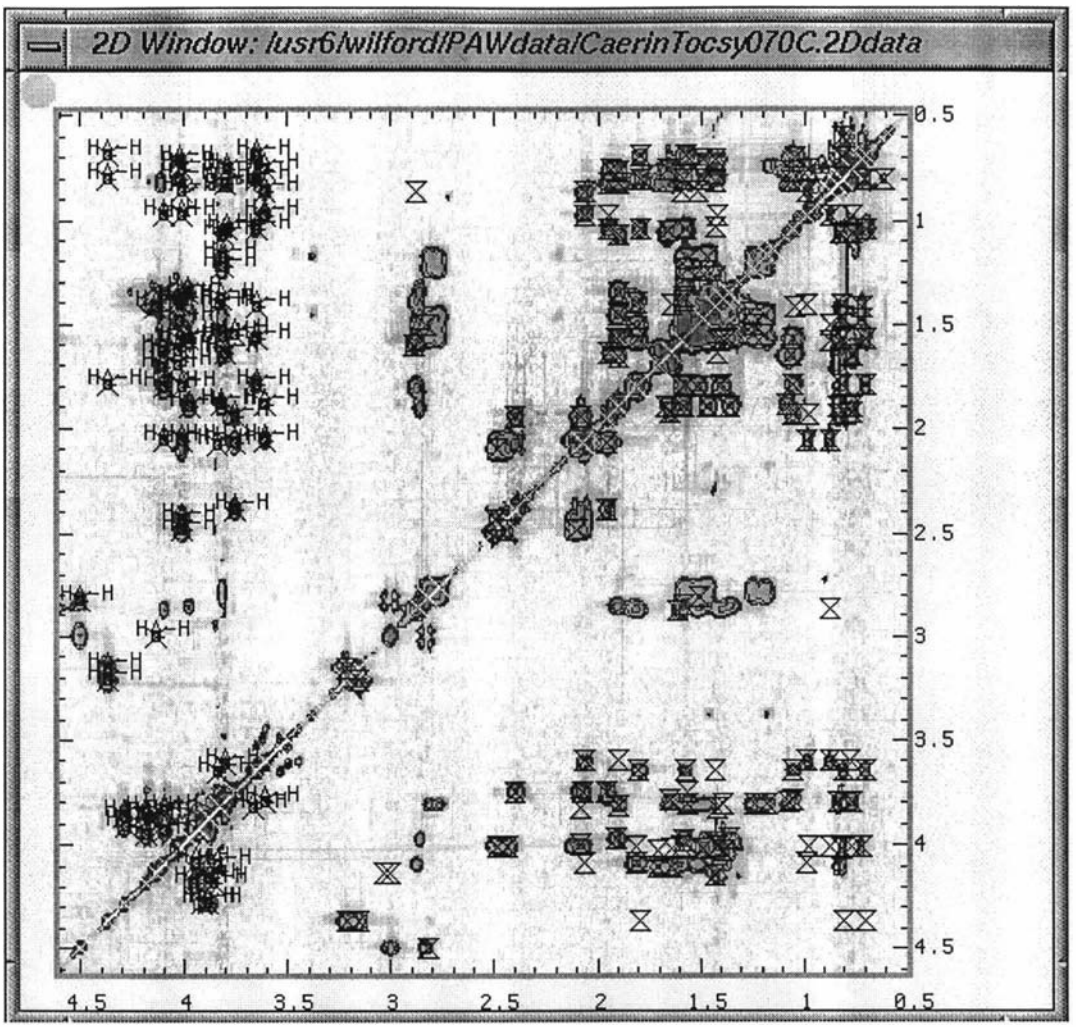


Figure 11.12 The assignment results for HA in D1in the upper-right region. Some of them should in fact be HB, and they are corrected later

- Type spl to save the peak list, preferably with a different name.

So far the group assignment has been done efficiently. The complete result may be viewed by typing zf. The full-screen view may be obtained by pressing <Alt-F10> and then click on the plot with MsBtn#3 to draw the peak symbols.

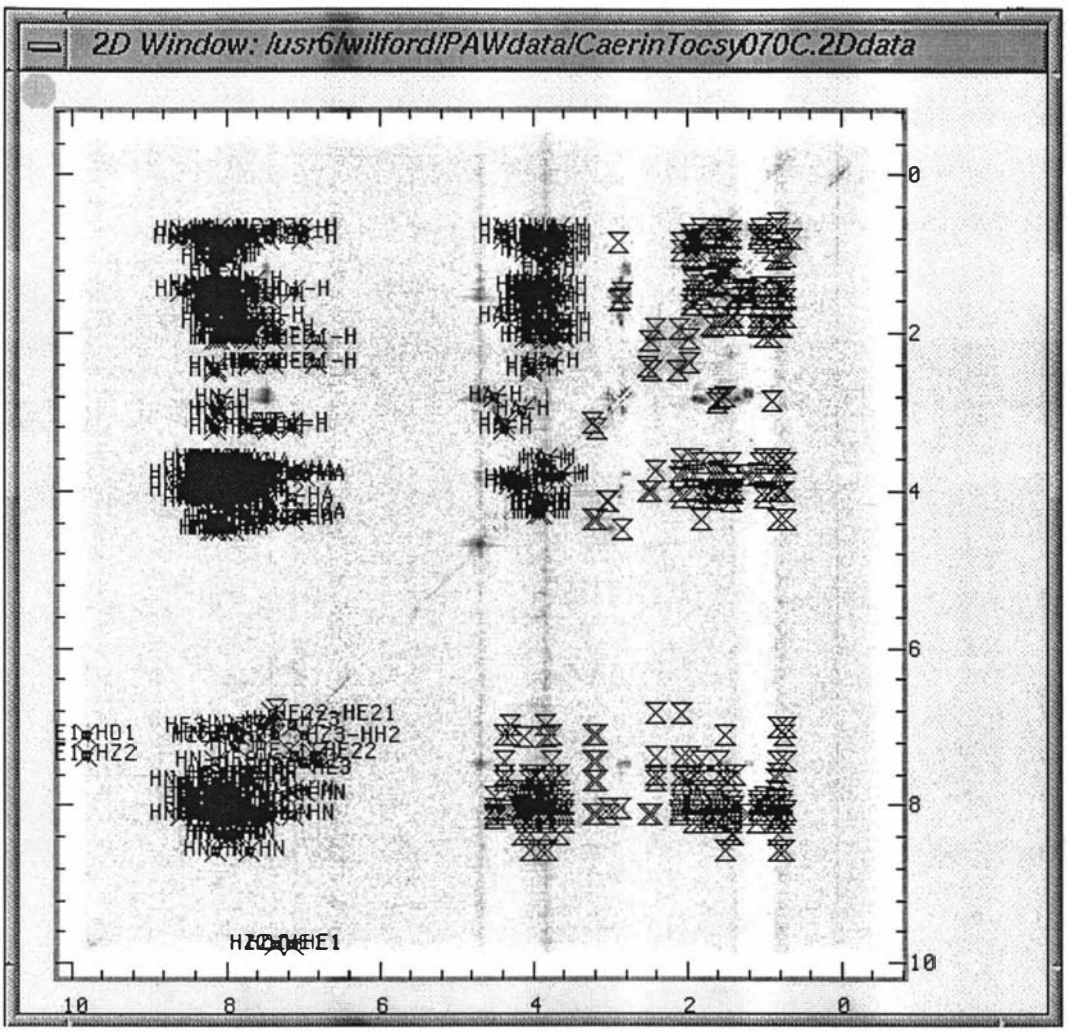


Figure 11.13 The complete result of the operations in Section 11.3 shown on a full view plot of the **CaerinTOCSY0707** spectrum.

11.4 Assigning Cross-peak Residue-codes

For the same reason mentioned in the previous section, this process is also more easily done using the TOCSY spectrum.

This process frequently requires the dtr and dch commands, which allow the drawing of transposed-rectangles and crosshairs, respectively. Be aware that these two commands are not terminated until *MsBtn#3* is clicked. These are two of the special operations that allow the drawing of more than one object. However, there could be some confusion since other commands will be ignored unless the dtr or dch command is cancelled. If this happens, just click in the draw-window with *MsBtn#3* and reissue the next command.

In the rest of this section, a few examples are given to show the assignment of residue-codes for Group-C cross-peaks in the lower-left region (Zm#4).

➢ An example of cross-peak analysis using multi-spectral display

The following example shows how to analyse and edit a number of NOESY cross-peaks in Zm#4, which have been displayed on top of a TOCSY spectrum.

- Click on the title-bar of the TOCSY window to bring it forward, and then click in the plot area.
- With the cursor on the plot area, type z4 to display the pattern shown in Figure 11.14.

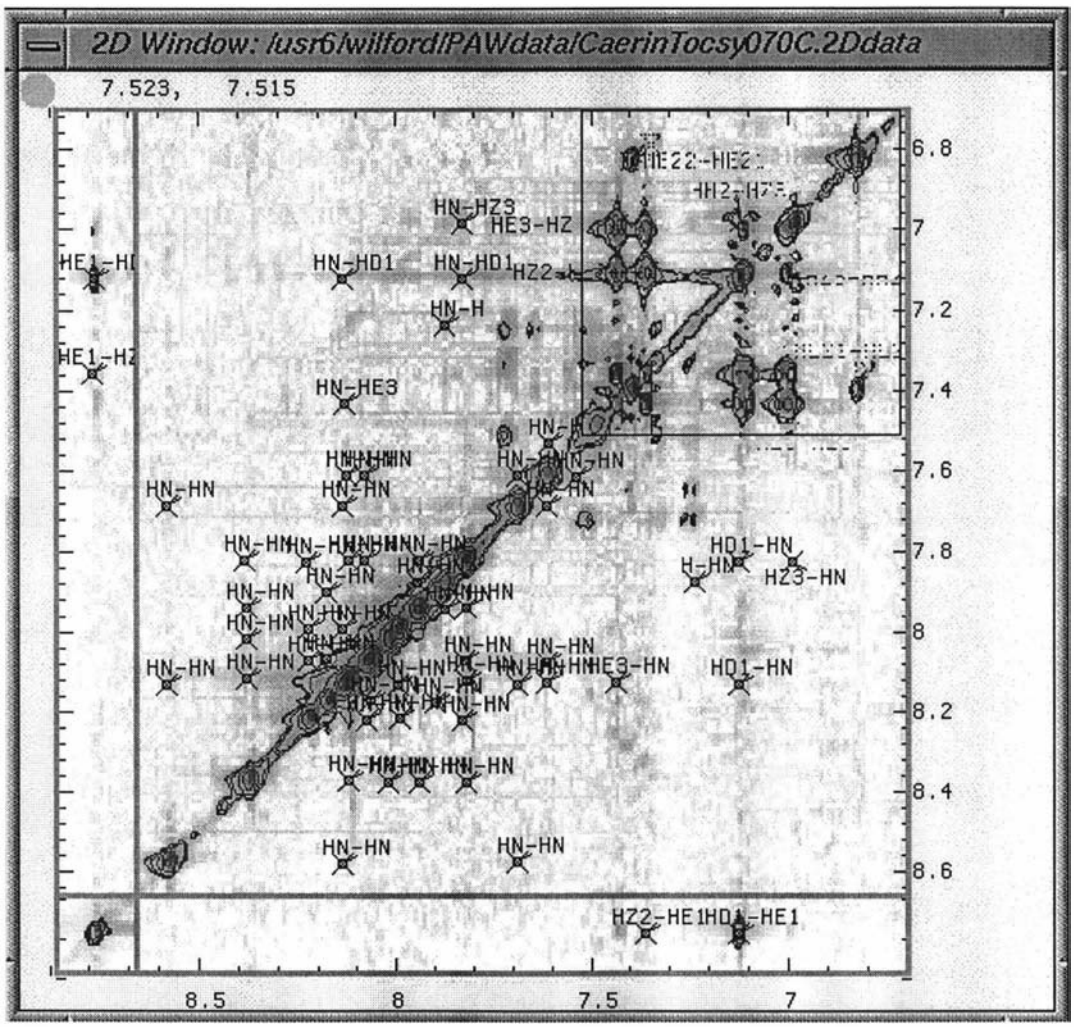


Figure 11.14 Zm#4 of the **CaerinTOCSY0707** spectrum.

- Type zs and select a region at the upper-right corner that includes all Group-C cross-peaks, as shown by the rectangle in the last plot. (See Table 7.4 for the definition of the Group-C cross-peaks.)
- Define this pattern as Zm#6.
- Repeat the last two steps for the NOESY and DQF-COSY window.

The three plots, as shown in Figure 11.15, can be cross-referenced for the unambiguous assignment of protons in the aromatic ring of the Trp and the HE2 protons of the Gln. Both amino acids are unique in the Caerin 4.1 sequence.

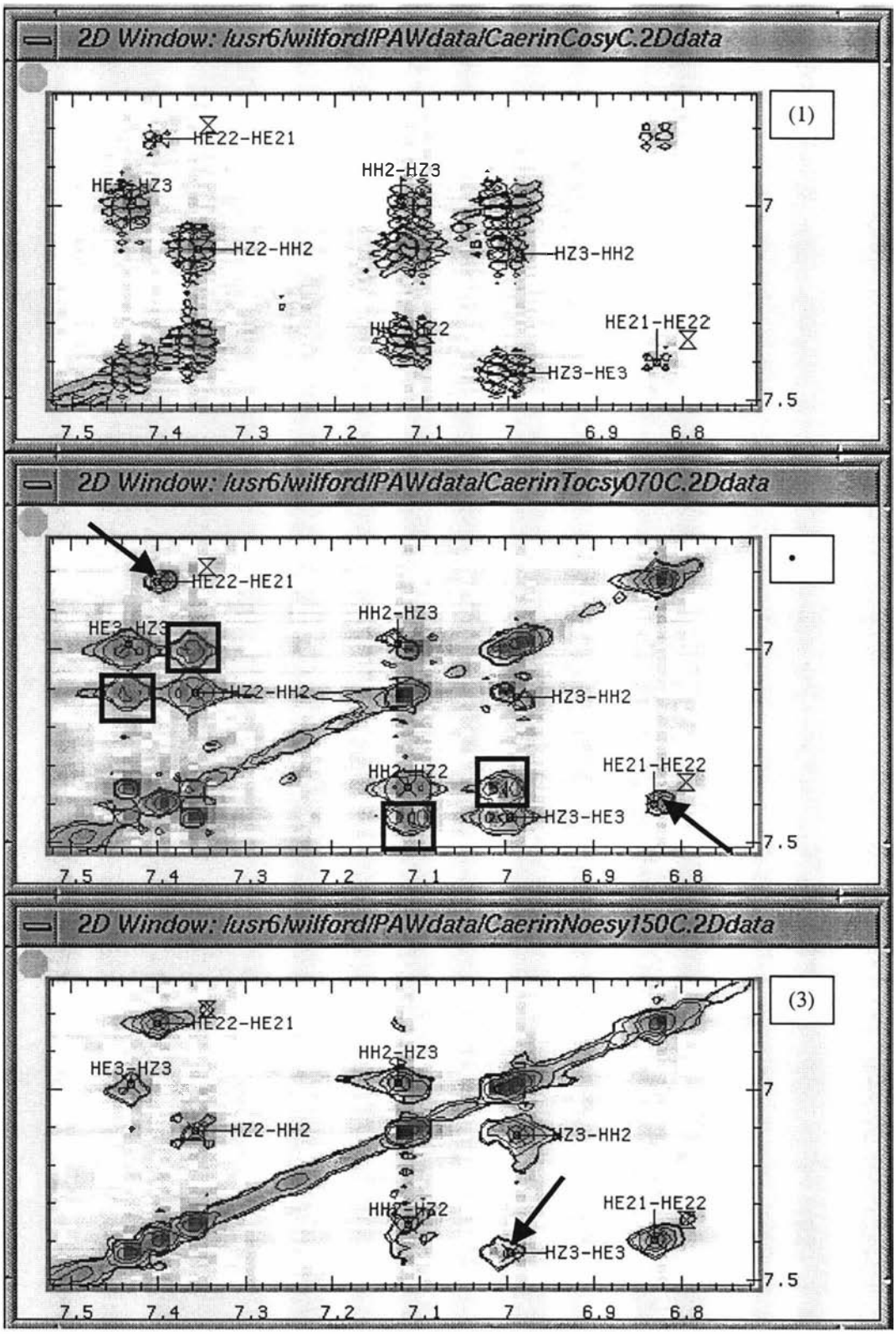


Figure 11.15 An example of a multi-spectrum display used in the cross-peak analysis.

The four TOCSY cross-peaks enclosed in the bold rectangles are apparently missing in the NOESY and DQF-COSY. This is a good indication of a Trp ring-structure, because HE3 and HZ3 are far apart from each other in the ring structure. (See Table 7.1 for the primary structure.) The two cross-peaks on the top and right (arrowed in Plot 2 of Figure 11.15) can only belong to Gln.

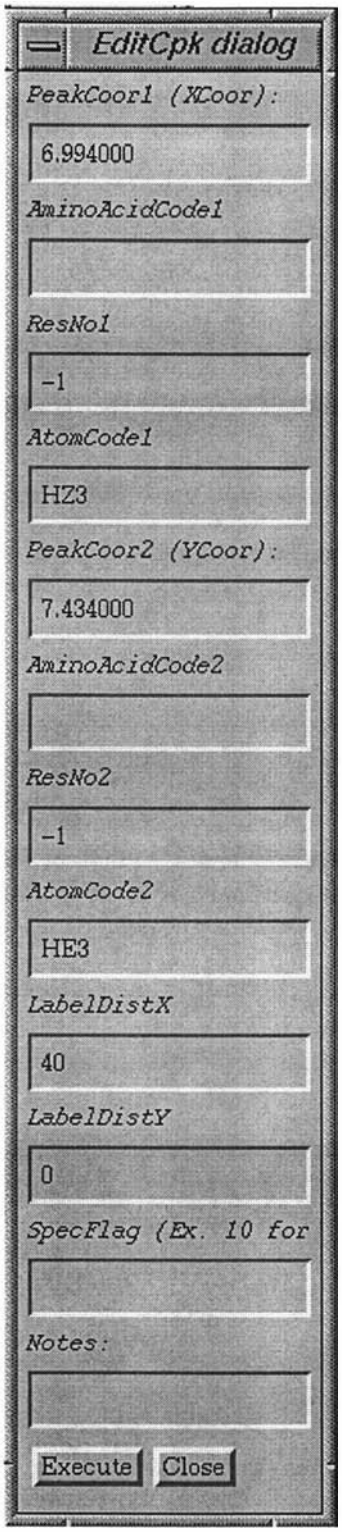
The analysis above leads to the assignment of all cross-peaks in Zm#6.

➢ Editing a cross-peak

The following example leads to the assignment of a peak arrowed in Plot 3 of Figure 11.15):

- Click on the title-bar of the TOCSY window to bring it forward, if necessary.
- Click on the plot.
- Type z6 to zoom into Zm#6.
- Type cpe or choose [EdtOneCPk] in the *Pick-assignment Toolbox* to start editing a cross-peak record. The message '*Use MsBtn#2 to select a peak to edit.*' will appear above the plot.
- Click on the HZ3:HE3 peak arrowed in Plot 3 of Figure 11.15 to open the *EditCPk Dialog*, as shown in Figure 11.16 (a), where the coordinates in D1 and D2 were drawn from the peak position and the *ResNo1* and *ResNo2* are set to –1 for any unassigned peaks.

(a)



(b)

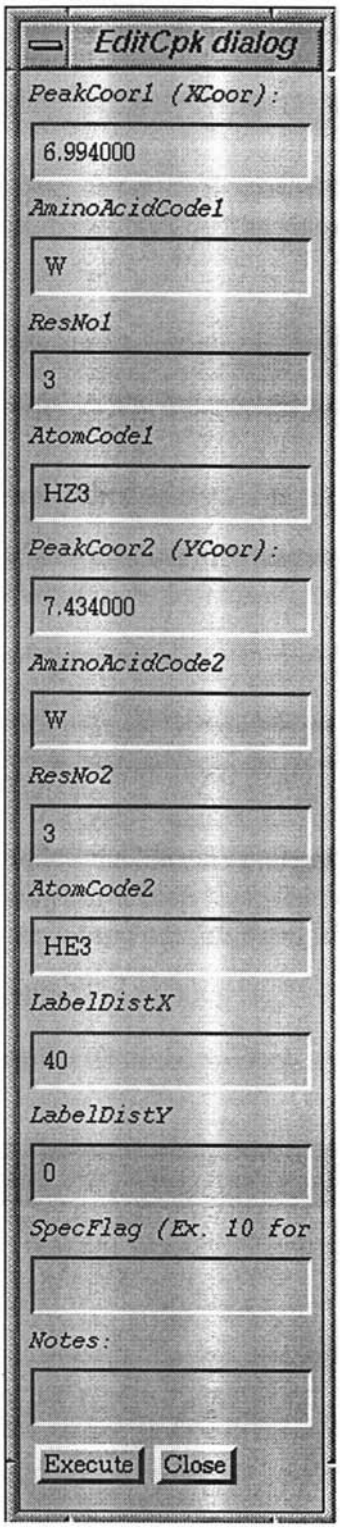


Figure 11.16 (a) The dialog for editing a cross-peak record, where –1 indicates an unassigned residue number. (b) The dialog for editing a cross-peak record with additional assigned values.

- Of the first eight items in the dialog, ignore the items for the *PeakCoor1* and *PeakCoor2*. Replace the other six items, starting from the *AminoAcidCode1*, with the values shown in Figure 11.16 (b).
- Choose [Execute].
- Click on the plot with *MsBtn#3* or type dr to refresh the display, if necessary. A noticeable change of the peak-label for the peak can be observed on the plot, as shown in the next figure.

➢ Editing a transposed cross-peak record

The following example leads to the transposed peak at around [7.50,7.04] being assigned.

- Type dtr to draw a transposed rectangle.
- Click on the cross-peak just edited in the last subsection. A transposed rectangle will be drawn onto the plot (Figure 11.17).
- Click in the window with *MsBtn#3* to stop transposed-rectangle drawing.
- Type tpe or choose [EditTranCPk] in the *Peak-assignment Toolbox*. The message '*Use Button 2 to select two peaks (one to copy, one to edit).*' will appear on top of the plot.
- Click on the peak centred at the lower-right corner of the rectangle first, and then click on the peak at the upper-left corner. The *EditCPk Dialog* will appear with the correct content for the second peak selected.
- Choose [Execute].
- Click on the plot with *MsBtn#3* to see the second peak assigned (Figure 11.17).

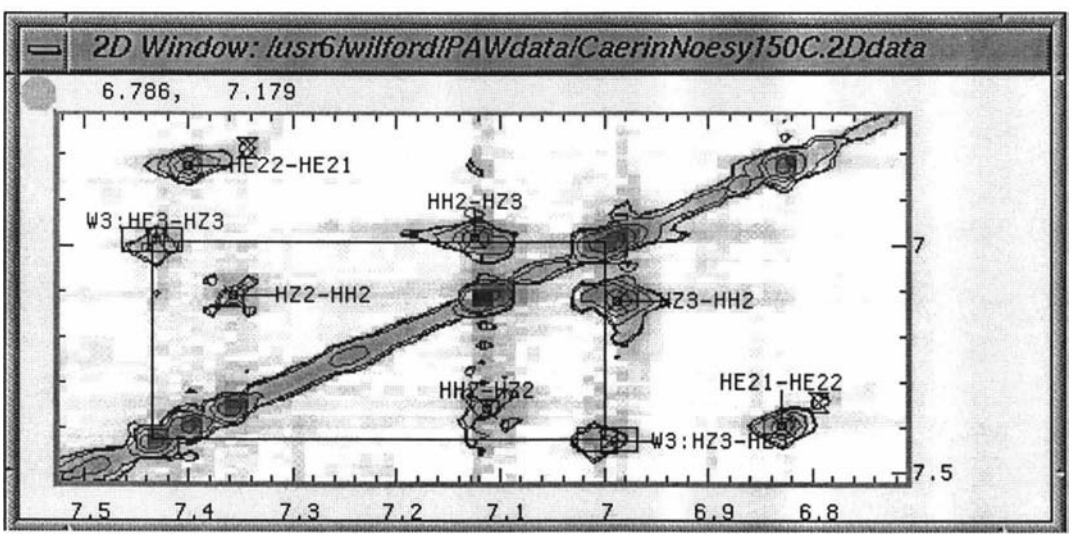


Figure 11.17 The result of a cross-peak assignment of W3:HZ3-HE3 after correction by using 1D spectral slices. (See Figure 11.14 for comparison.)

➢ Editing a cross-peak position

Often, a cross-peak position needs to be shifted slightly because of overlap. It can be seen from the last plot that the two W3:HZ3-related cross-peaks on the right do not align vertically. Likewise, the corresponding transposed cross-peaks are poorly aligned in the horizontal direction. This is because there is an overlap with the terminal NH_2 cross-peaks, symbolised respectively as X24:NH1 and X24:HN2.

The approximate D2 coordinate can be set to 7.007 ppm, which can be seen from the two 1D spectra (figures not shown) that are drawn from the row (or column) that pass through the centres of the two cross-peaks. The following operations show the steps to shift a cross-peak position.

- Type cpe and then click on one of the cross-peaks to be edited. This operation opens the EditCpkDlg.
- In the dialog, change either the PeakCoor1 or PeakCoor2 to 7.007, depending on the cross-peak selected.
- Choose [Execute] in the dialog.
- Type dr to redraw the plot, as shown in Figure 11.18.

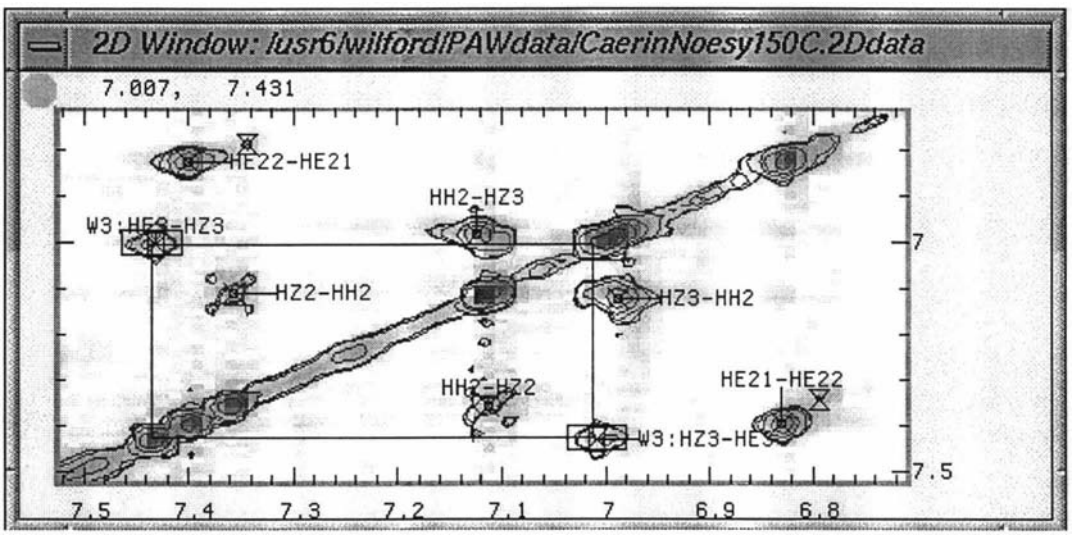


Figure 11.18 The result of a cross-peak assignment of W3:HZ3-HE3 after being refined.

➢ Adding a pair of cross-peaks during the assignment

When splitting a cross-peak into two, another pair of cross-peaks needs to be added. Two steps are required: picking a raw peak and then finding a transposed cross-peak. The approximate position of a cross-peak can be obtained by drawing crosshairs from related cross-peaks that are well defined. The following operations show the steps required for this purpose.

To identify the peak position:

- Type dch to start drawing crosshairs.
- Click on the centres of two peaks inside the bold rectangles in Figure 11.19.

- Click in the plot with MsBtn#3 to stop the draw-line operation.

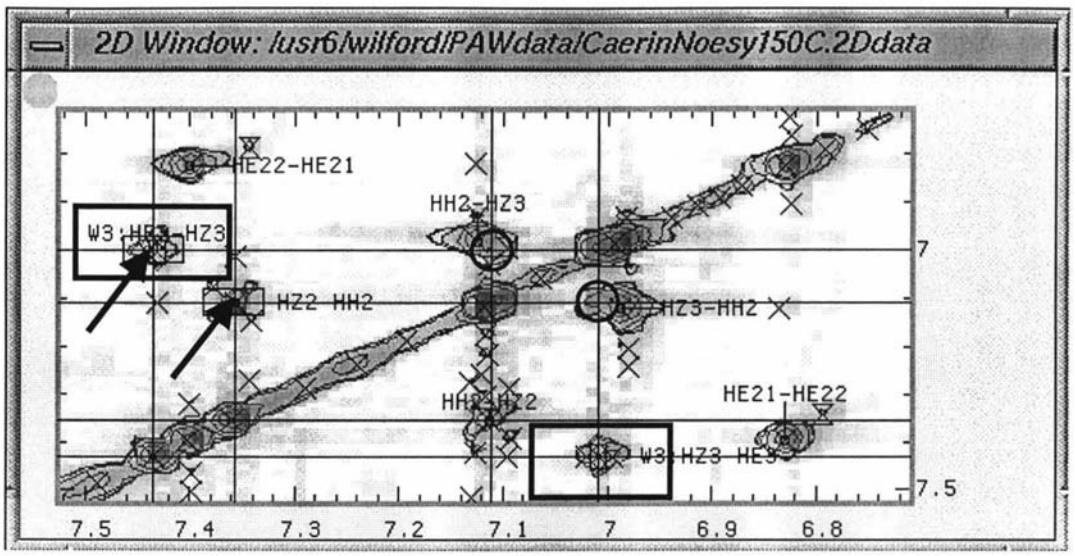


Figure 11.19 A plot that shows two sets of crosshairs that were drawn from the two peaks arrowed.

The two crosshair intersections (circled) define the missing peak positions. The following two steps add two cross-peaks at the intersections.

- Type rp1 to pick one raw peak, where the last letter is the numeral 1 that refers to one. Then, click on one of the intersections to add a raw peak.
- Type tpf to find a transposed peak. Then, click on the new raw peak to add a pair of cross-peaks.

➢ Lining up cross-peaks during the assignment

Once a cross-peak has been correctly assigned in one dimension, all other related cross-peaks can be lined up and assigned in the dimension using the *LineUpCpk* command. This process can largely reduce the necessity of entering individual records. It has been widely used in the entire assignment process. The examples below illustrate the steps required:

- Type cpl to line up cross-peaks, where the last letter is the alphabet l that refers to line-up. The *LineUpCpk Dialog* will pop up (Figure 11.20).
- Click on the W3:HZ3-HE3 peak at the bottom of Figure 11.21 as the reference peak. Then, click on the unassigned new cross-peak above it.
- Choose [Execute] in the dialog. The new peak position will be vertically lined up with the reference peak and the plot will be re-displayed with the W3:HZ3 correctly assigned for the new peak.
- Repeat the last three steps using the W3:HZ2-HH2 peak as a reference. The new peak position will be horizontally lined up with the reference peak and the HH2 will be correctly assigned for the new peak.

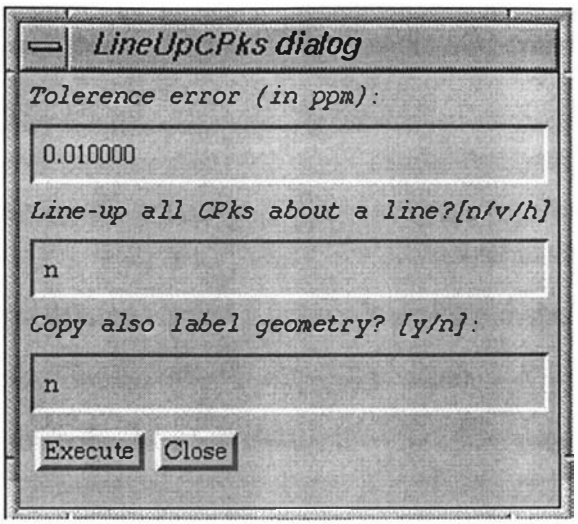


Figure 11.20 The *LineUpCPks Dialog*.

Figure 11.21 shows the peaks assigned in Zm#6, where the false raw peaks around the new cross-peaks have been removed.

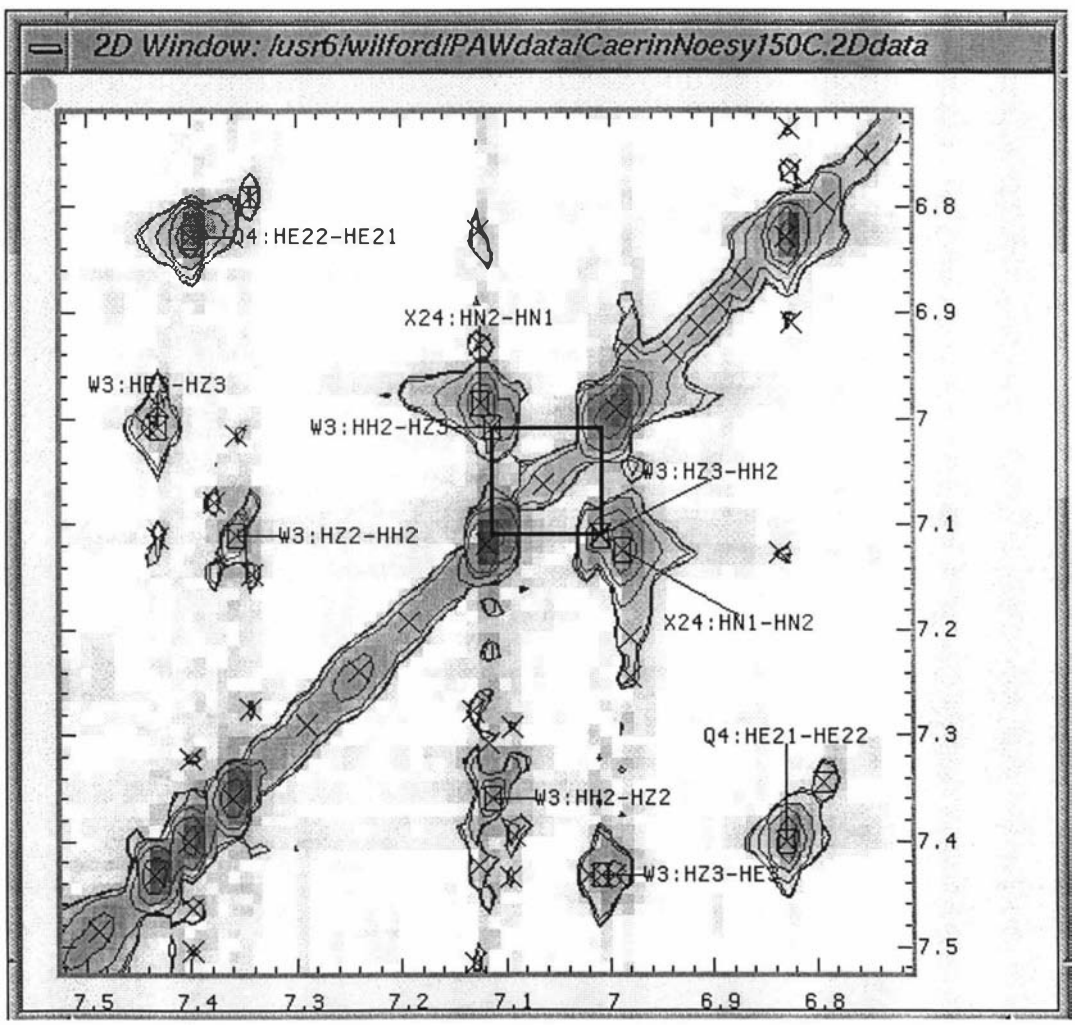


Figure 11.21 A plot that shows a new pair of cross-peaks that have been picked and assigned based on the positions of the W3:HE3-HZ3 and W3:HZ2-HH2 cross-peaks.

PAW allows all peaks in a column or a row within a specified range to be lined up and assigned at the same time. This enables the assignment to be done without having to zoom into every region, as long as it is clear that no other peaks are within the range. It is recommended that the peaks be lined up with an isolated peak, for which a good estimate of the peak centre can be obtained. The following example shows how this can be done within Zm#6.

- Type cpl to line up cross-peaks. The *LineUpCpk Dialog* (Figure 11.20) will pop up. In the dialog, change the entry for the second item to v.
- Click on the W3:HE3-HZ3 peak on the left of Figure 11.22 as a reference.
- Choose [Execute] in the dialog.
- Repeat the same operation, from left to right, for Q4:HE22-HE21 and Q4:HE21-HE22.
- Type z1 to show all the peaks in these three columns that have been lined up and assigned (Figure 11.22).

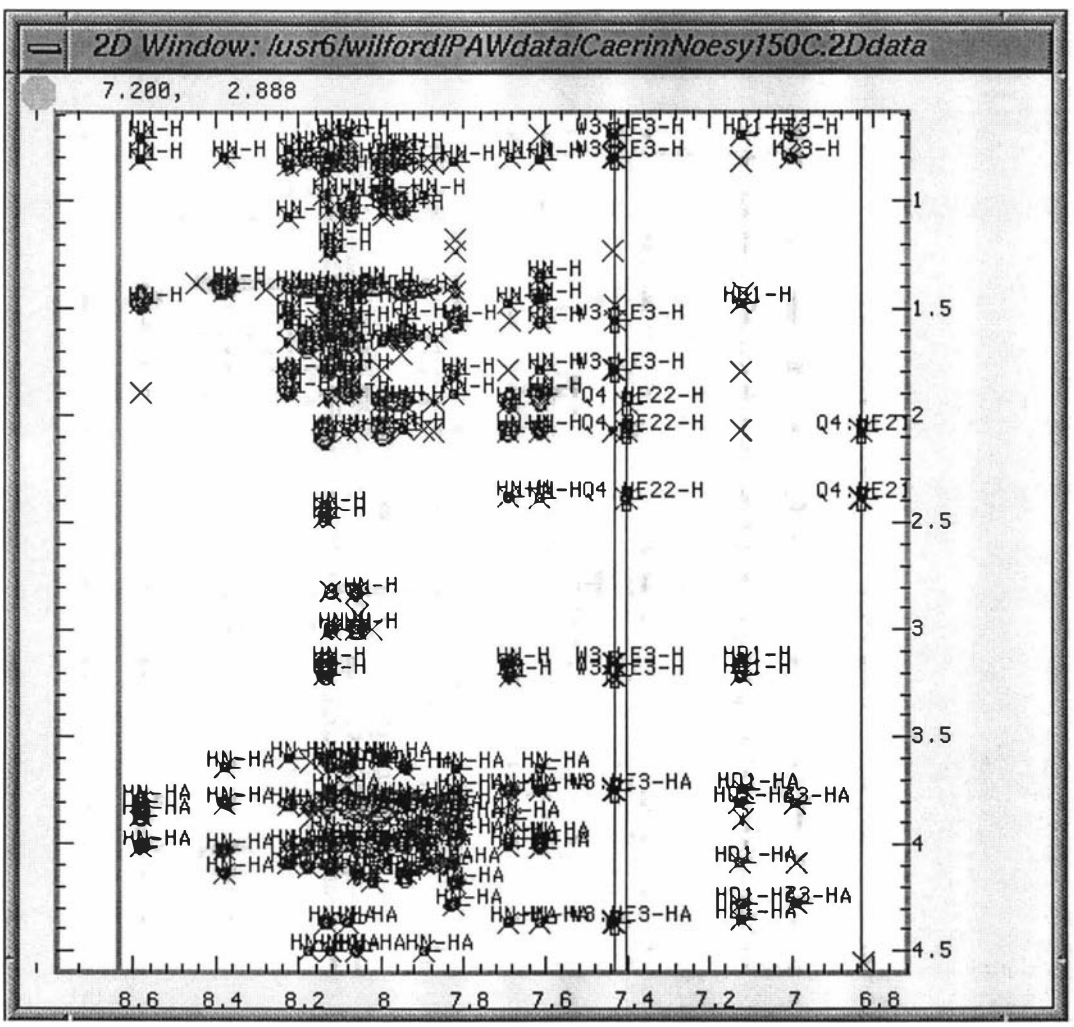


Figure 11.22 A plot that shows three columns of peaks that have been properly lined up and assigned in the upper-left region of the Caerin 4.1 NOESY150 spectrum.

Note:

- It is recommended that the peak lists be saved before performing a lining up process, just in case some peaks may be lined up unexpectedly.
- Peaks that have very similar chemical shifts can only be lined up individually by careful judgement, such as the W3:HD1- and X24:HN2-related peaks.
- 127 cross-peaks can be lined up at one time. To line up more than one peak, click on a reference peak first and then keep clicking on all peaks to be lined up before choosing [Execute] in the dialog.

➢ Moving cross-peak labels

By default, the label for an assigned peak is centred at the upper-right corner of the peak. This arrangement may not be the best, especially in a crowded region. For example, the label arrangement in the last plot is not suitable for publication. The positions of the labels in Figure 11.21 are acceptable, and they have in fact been shifted.

One way to shift the positions is to follow the peak editing procedure described previously, and change the values for the *LabelDistX* and *LabelDistY* from the *EditCpk Dialog*. PAW also provides a graphic-interface method used to shift the peak labels, as illustrated in the following example.

- Type z6.
- Type mlb for moving a peak label. The message '*Use Button 2 to click on a peak then a position.*' will appear above the plot.
- Click on the centre of any peak for which the label is to be shifted. A large cross will be drawn from the peak centre as a reference for the position selection, as shown in Figure 11.23.

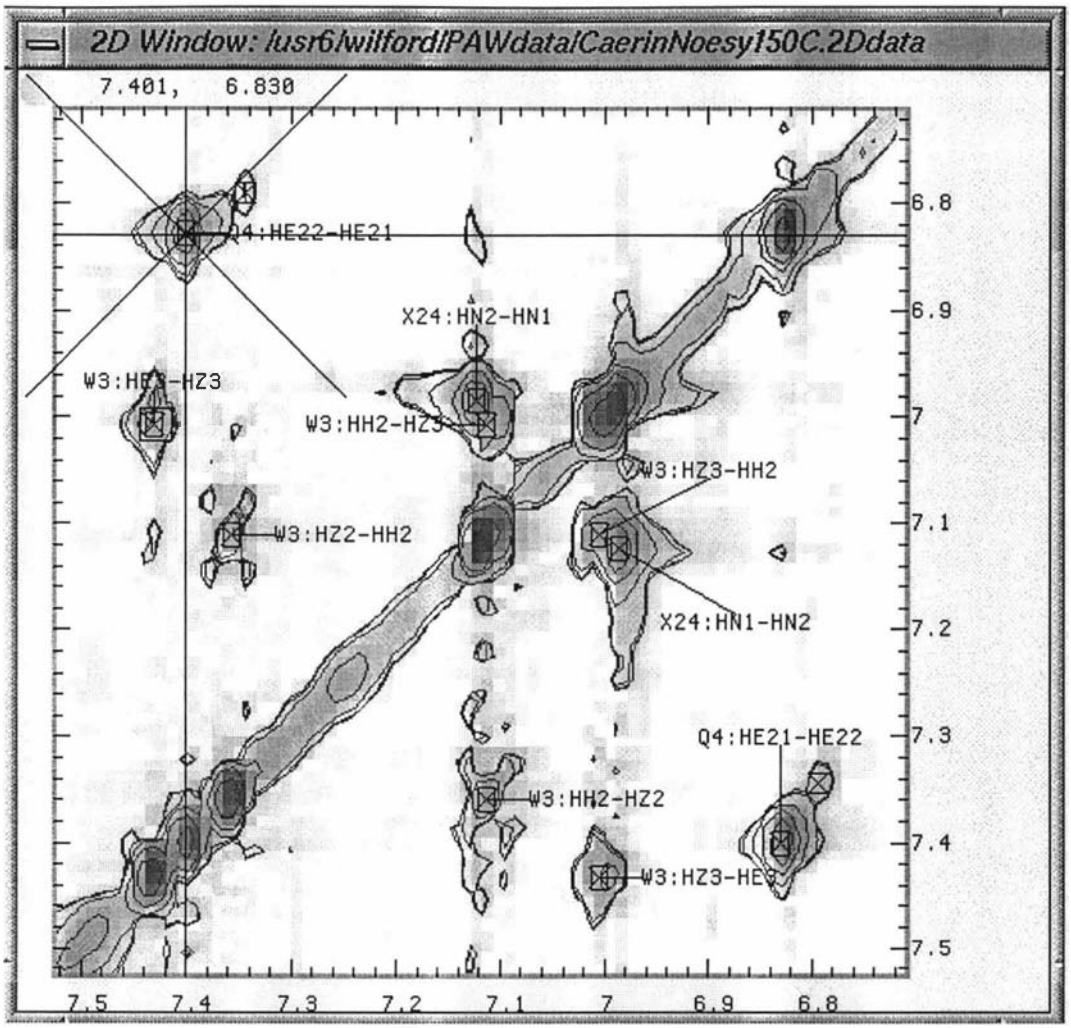


Figure 11.23 A large cross that was drawn on a plot for the reference to select a new peak-label position.

- Click on any position where the label is to be placed, not necessarily on the cross.
- Repeat the above steps to shift other labels, if any.
- Type dr to show the new label positions, as shown in Figure 11.24.

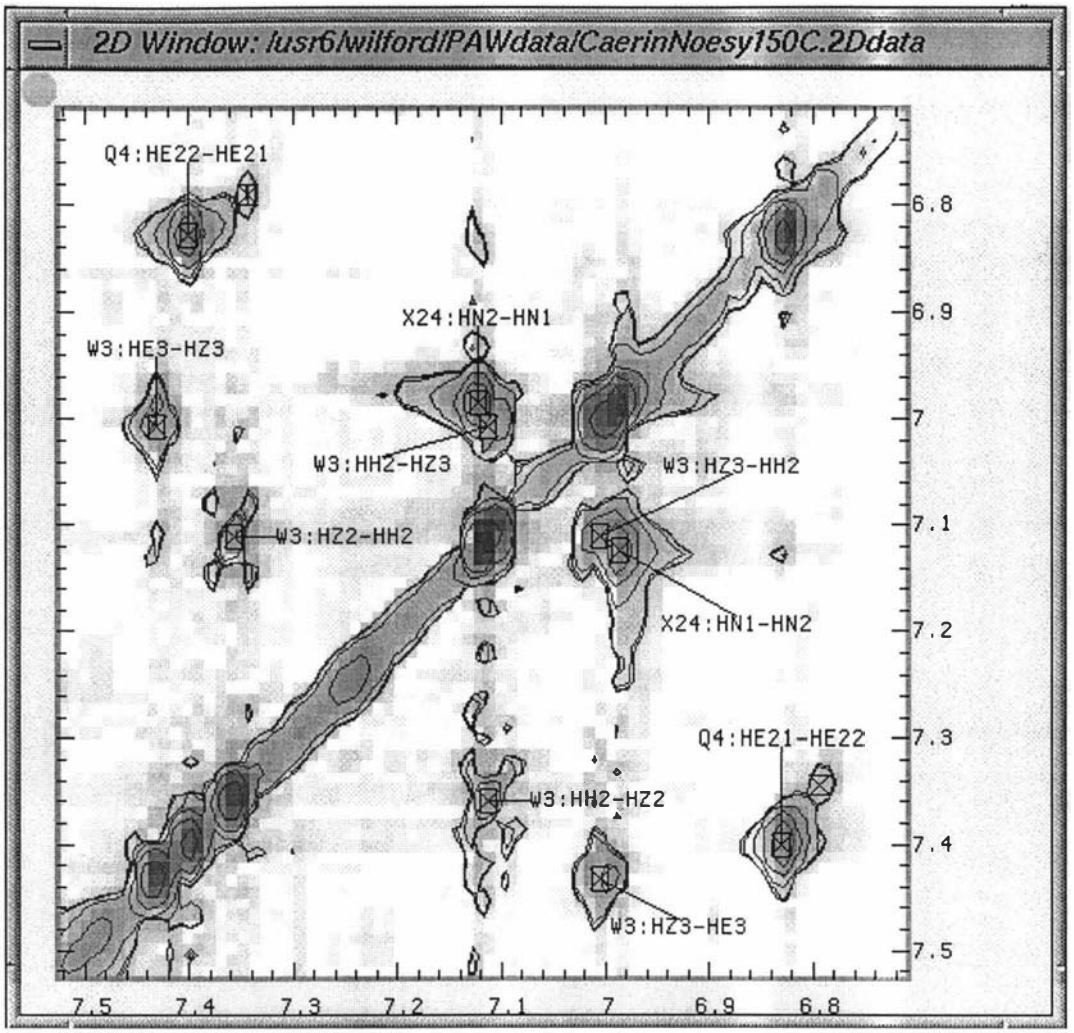


Figure 11.24 A plot with a better label-display layout. The labels were moved using the graphic-interface method.

11.5 Sequence-specific Assignment

Sequence-specific assignment is a very time-consuming process, especially for larger molecules.

Both the NOESY and TOCSY spectra are used during this process. Displaying the NOESY peaks on top of the TOCSY spectrum allows most inter-residue connectivities to be identified, except those in heavily overlapped regions. In addition, the comparison between the intensities of the NOESY inter-residue cross-peaks allows the sheet or helical structure to be recognised. Important information about the distances between protons in basic protein secondary structures can be found in Table 7.5.

The fingerprint and HN-HN regions are usually two of the regions to look at first.

➢ Assignment for the unambiguously known residues

The example below starts with the assignment of all unambiguously known residues in the sequence. In Caerin 4.1, there are five of them. Figure 11.25 shows their HN-HA cross-peaks in the fingerprint region of the **CaerinTocsy070C** spectrum. These are, from top to bottom, V18, Q4, E19, W3 and D12.

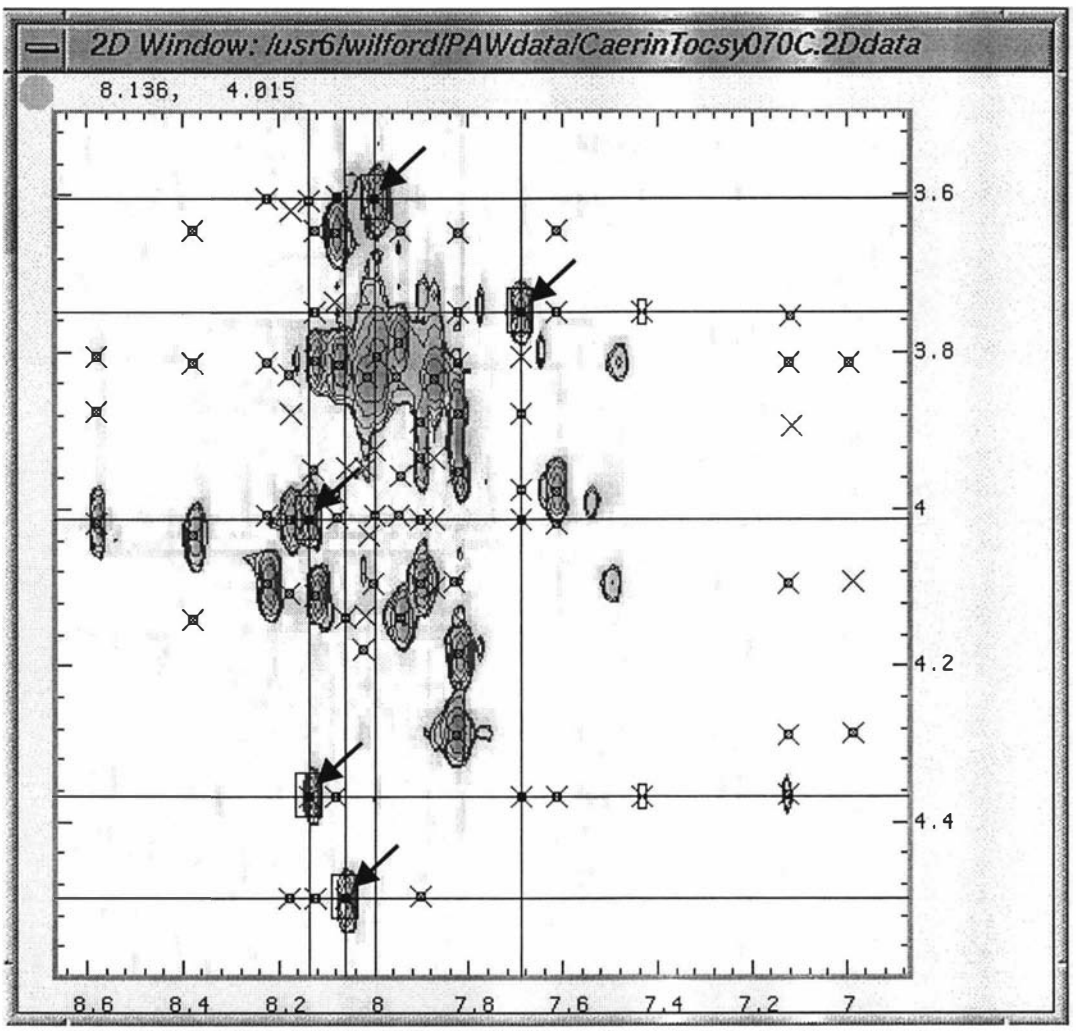


Figure 11.25 The HN-HA cross-peaks of five unambiguously known residues in the sequence. The crosshairs were drawn to identify cross-peaks that are related to them.

- Type rad to remove all drawing items, including all lines and rectangles.
- Click in the TOCSY draw-window.
- Type z7 to zoom into the region shown in the last plot.
- Assign the HN-HA cross-peaks of the five known residues.
- Line up all cross-peaks that can be clearly separated vertically and horizontally, including those in the columns for V18, Q4, W3 and D12, as well as those in the rows for D12 and Q4.

Figure 11.26 shows the Zm#1 plot of the NOESY spectrum after the operation.

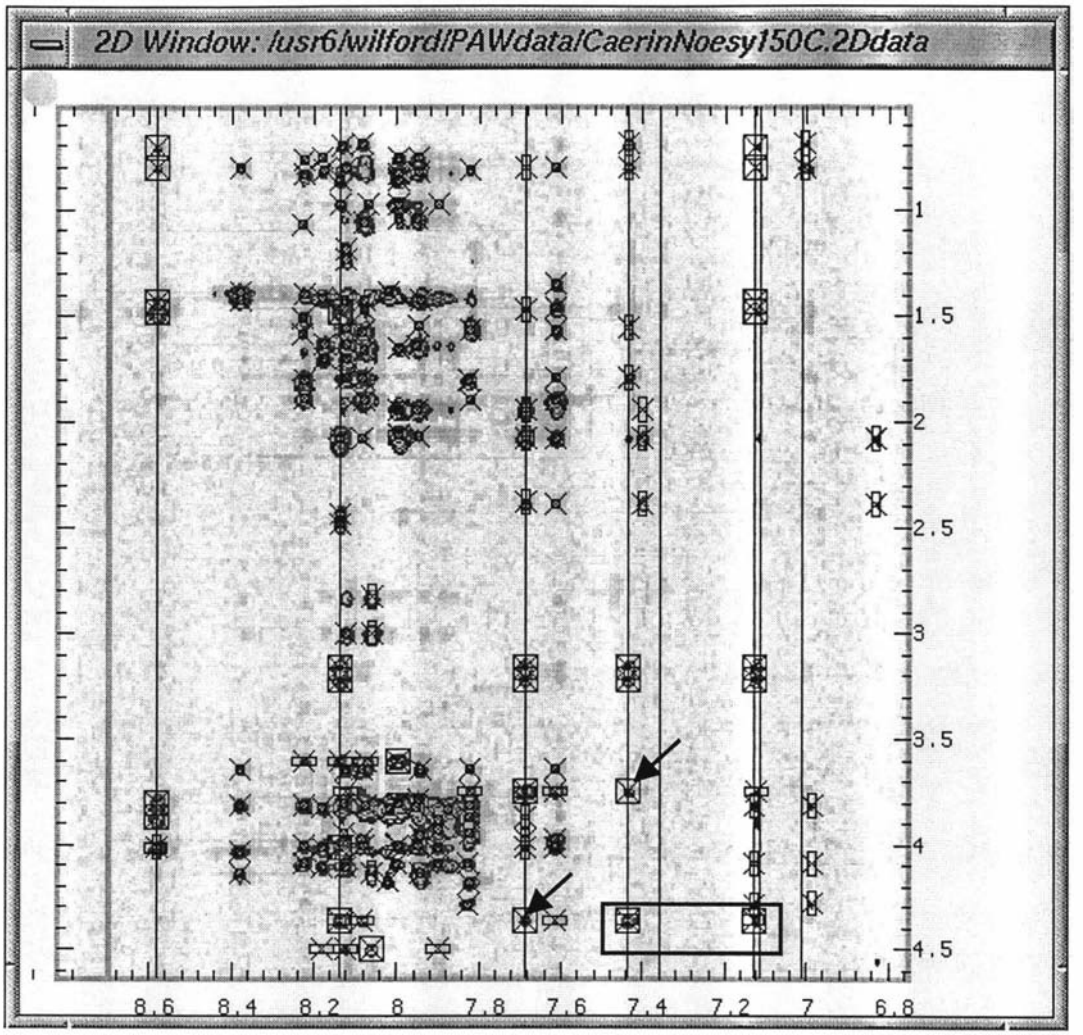


Figure 11.26 The cross-peaks in the upper-left region of the **CaerinNoesy150C** spectrum. The two fully-assigned cross-peaks that are blocked are W3:HE3-HA (left) and W3:HD1-HA (right). and two inter-residue cross-peaks arrowed are W3:HE3-Q4:HA (left) and Q4:HN-W3:HA (right). The peak labels are plotted later in larger plots.

As can be seen from the symbols in the last plot, some peaks are partly assigned, and some are fully assigned. Apart from the five intra-residue HN-HA peaks that have been assigned individually, there are four other cross-peaks that are fully assigned by the lining up operations. These are the two W3 intra-residue cross-peaks W3:HE3-HA and W3:HD1-HA that are along the bottom row, and two W3-Q4 inter-residue cross-peaks W3:HE3-Q4:HA and Q4:HN-W3:HA, as arrowed in the figure.

➢ Identification of the adjacent residues

Starting from the cross-peaks of a known residue, most of the adjacent residues can be found.

For example, there are two Leu residues in the sequence. The following operations allow L2 cross-peaks to be identified from the assigned W3 cross-peaks.

- Click on the TOCSY spectrum.
- Type z7 to show mainly HN-HA-related cross-peaks.

- Type dch to draw crosshairs.
- Click on the HN-HA cross-peaks of W3, L2 and L13 residues. Then, hit the MsBtn#3 to stop the drawing operation.
- Click in the NOESY draw-window and type z4, because the inter-residue cross-peaks between W3:HN-HA and any of the two HN-HA cross-peaks of Leu residues cannot be unambiguously identified in Zm#7.

From the plot in Zm#4, it is clear that the chemical shift of L2:HN is the one at approximately 8.6 ppm, as labelled in Figure 11.27. (As a consequence, the chemical shift of L13:HN is also identified to be about 8.12 ppm, also as labelled in the same figure.)

Note that the next three figures were produced by switching off the raw-peak display and with the label format set to "None".

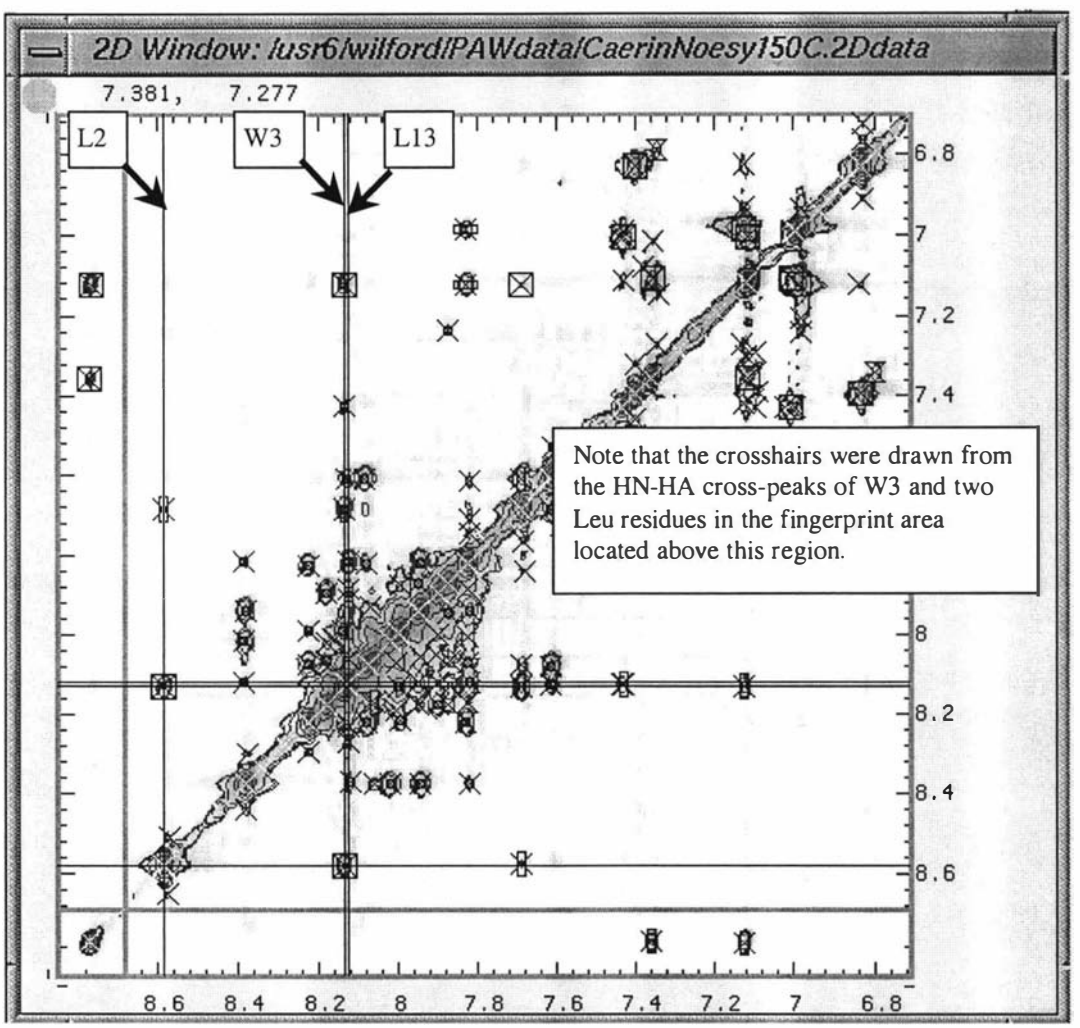


Figure 11.27 Three sets of crosshairs used to identify L2-related cross-peaks. The apparent cross-peaks at the intersections indicate clearly that the chemical shift of L2:HN is the one at around 8.6 ppm.

Q4-related peaks can be found in the same way. The next two figures show all the L2, W3 and Q4-related cross-peaks in Zm#4 and Zm#1. The figures were obtained as follows:

- Type ral to remove all crosshairs.
- Type dch to draw crosshairs.
- Click on the peaks arrowed in the figures.

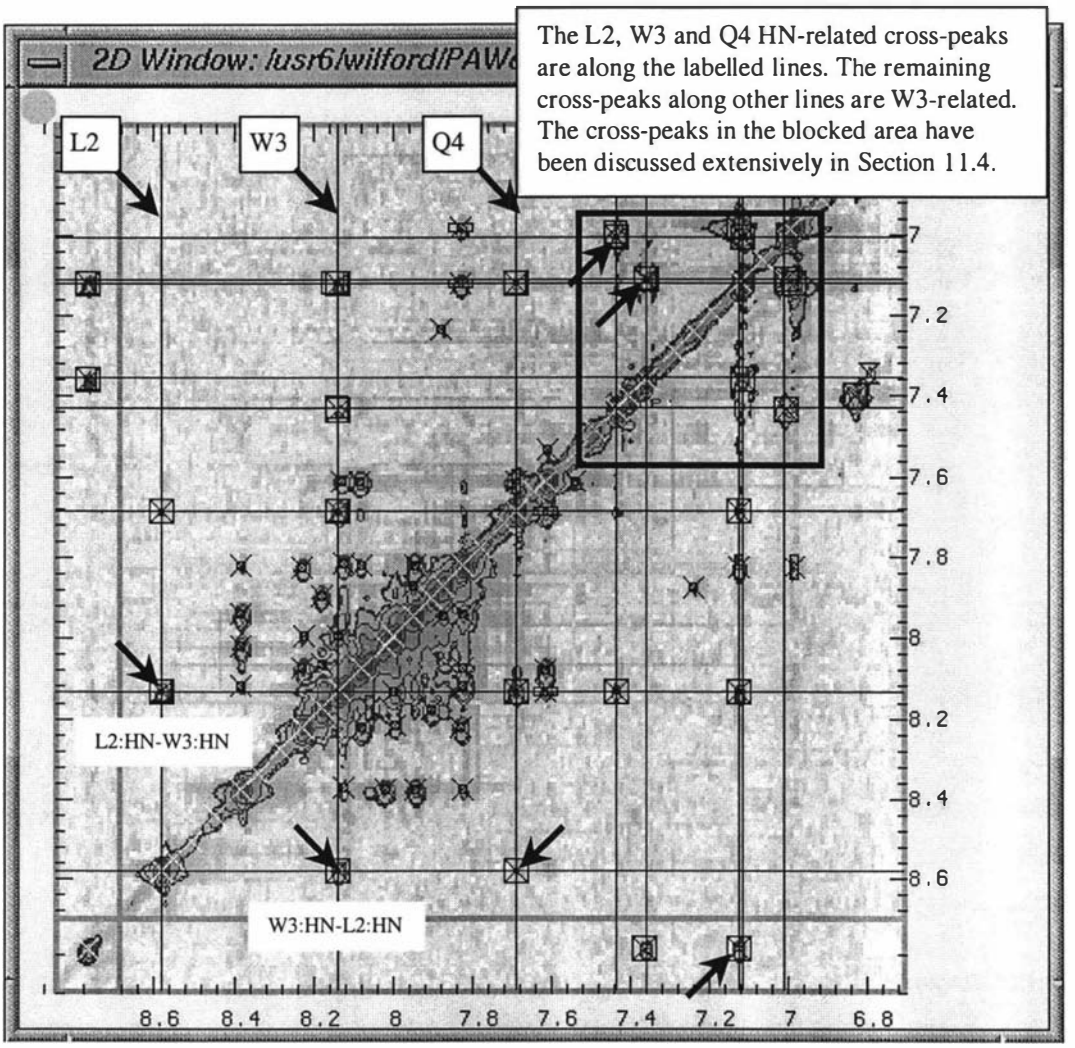


Figure 11.28 The crosshairs that highlight all the L2, W3 and Q4-related cross-peaks in the lower-left region.

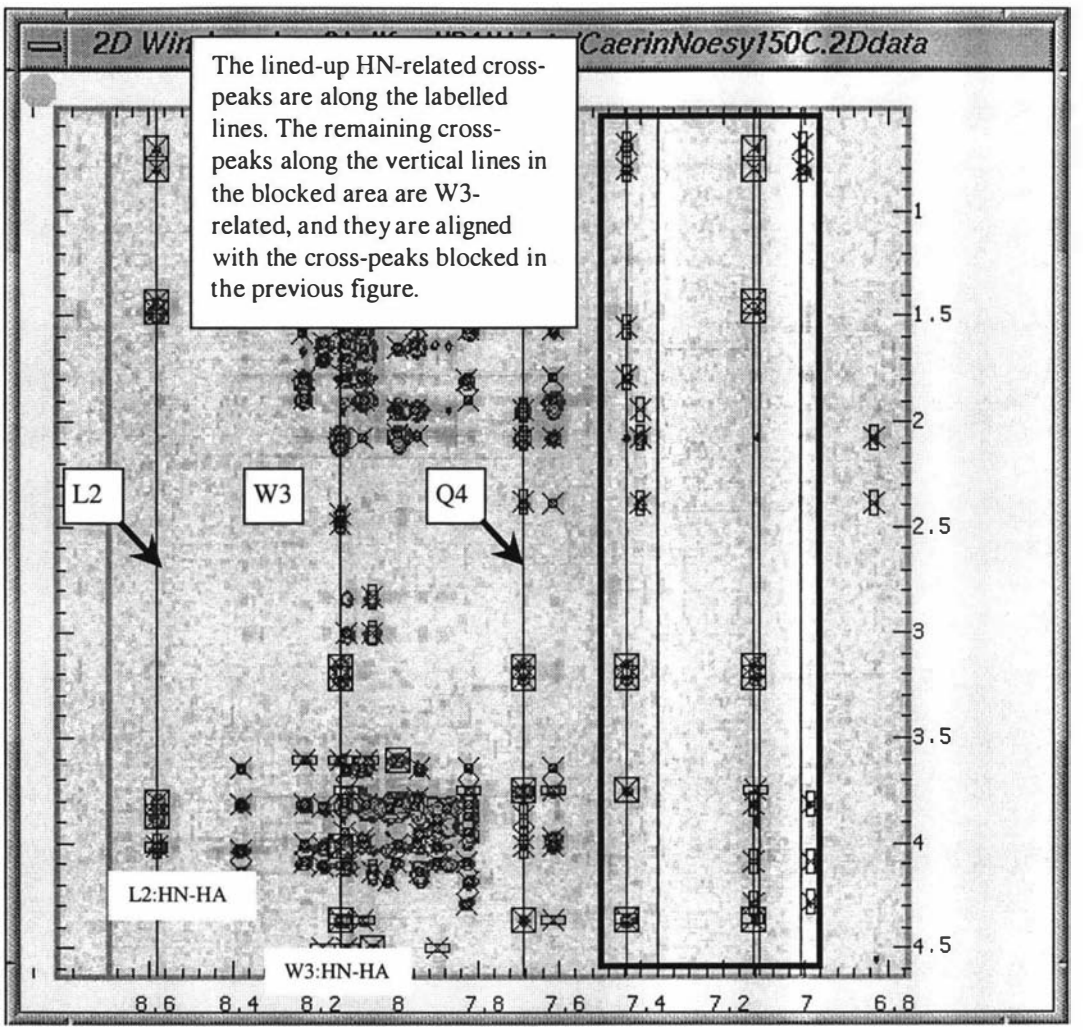


Figure 11.29 The crosshairs that highlight all the L2, W3 and Q4-related cross-peaks in the upper-left region.

Other inter-residue cross-peaks can be found and assigned in the same way. Most of them are assigned by the lining-up operation. Note that lining-up peaks is a powerful group assignment operation but is irreversible. To avoid unexpected mistakes, it is strongly recommended that the peak lists be saved before the operation.

➢ Using various peak-display options in the sequence-specific assignment

Various peak-display options can be used in the sequence-specific assignment to assist the search for the inter-residue cross peaks. These include the switching of six drawing-object display modes, ten label display formats, 21 spin-systems, and three peak-lists, as shown in the *Peak-display Toolbox*, as shown in Figure 11.1 (a). (The code Z in the *SpinSystem Checkbox* is used for any unassigned spin systems, and is not a standard code.)

A combination of peak-display options can be made through clicking on a number of buttons in the toolbox. To avoid unnecessary display operations, a plot will only be redisplayed after choosing [Execute] or typing dr.

To illustrate the advantages of various peak-display options, an almost fully assigned peak list set is required to be loaded as follows:

- Click on the CaerinNoesy150 draw-window.
- Type z1.
- Choose [LoadPkLists] in the *Peak-assignment Toolbox* to open a file-selection dialog.
- Double-click on the **data7noesy150h.RPeaks**. in the file list. This will also load the corresponding diagonal peaks and cross-peaks.

Now,

- Switch on the [DspRPks] and [DspCPaks] in the *Peak-display Toolbox.*
- Switch the label-display format to [None].
- Type dr.

These operations will result in Figure 11.30, where the simple crosses are raw peaks. The rectangles are assigned cross-peaks.

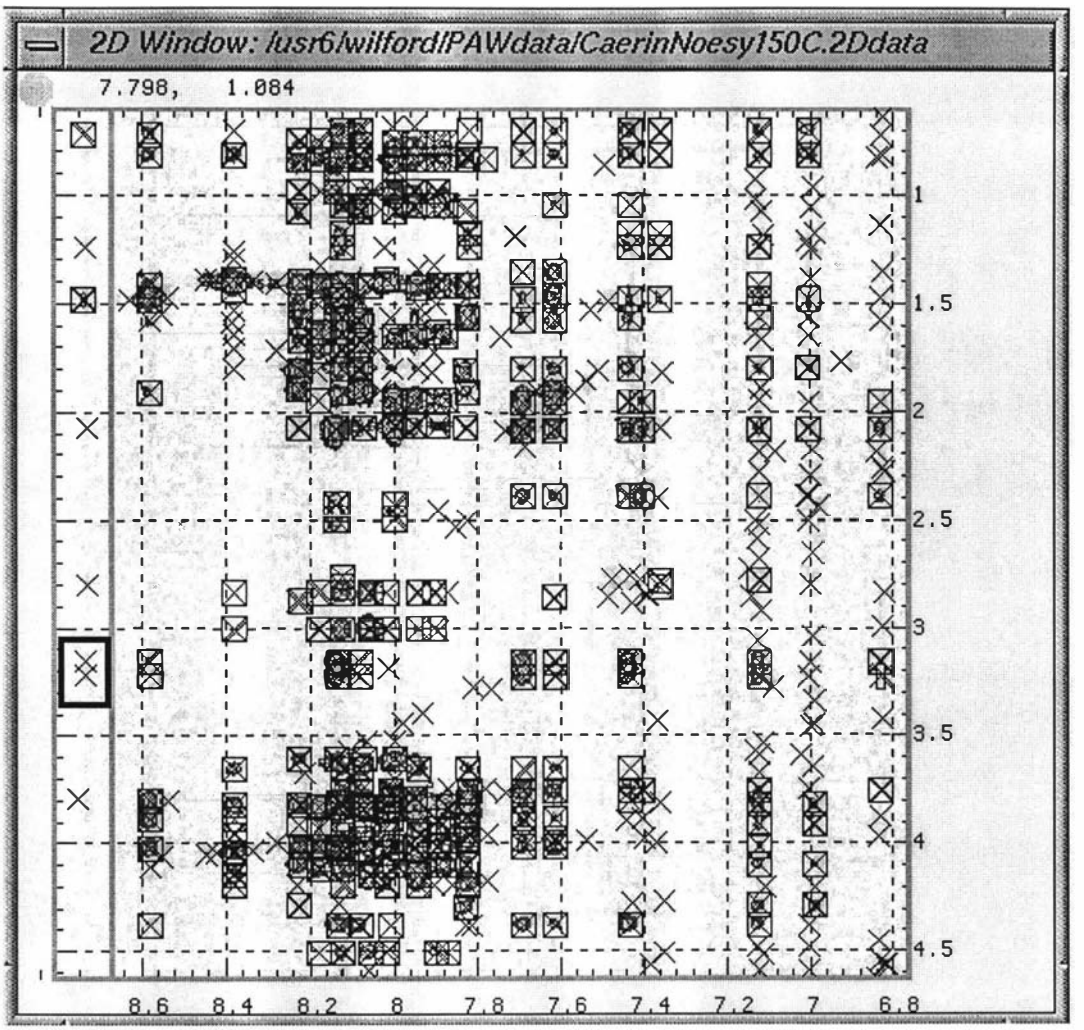


Figure 11.30 The upper-left region of the NOESY150 spectrum. This is displayed with an almost fully assigned cross-peak list and the raw peak list from which the cross-peaks were selected. The label format for both plots was set to "none" in order to have a better view of all peak symbols.

In can be seen that numerous raw peaks were not selected as cross-peaks. Most of them are very weak and are noise, but some are real, such as the two very weak raw peaks at about 3.15 ppm (in D2) on the left. (These two peaks were not assigned because they were less important in structure determination, but they did provide extra information about cross-peak connectivities.)

Now,

- Switch off the [DspRPks] button.
- Type dr to redraw.

The result is shown in the next figure.

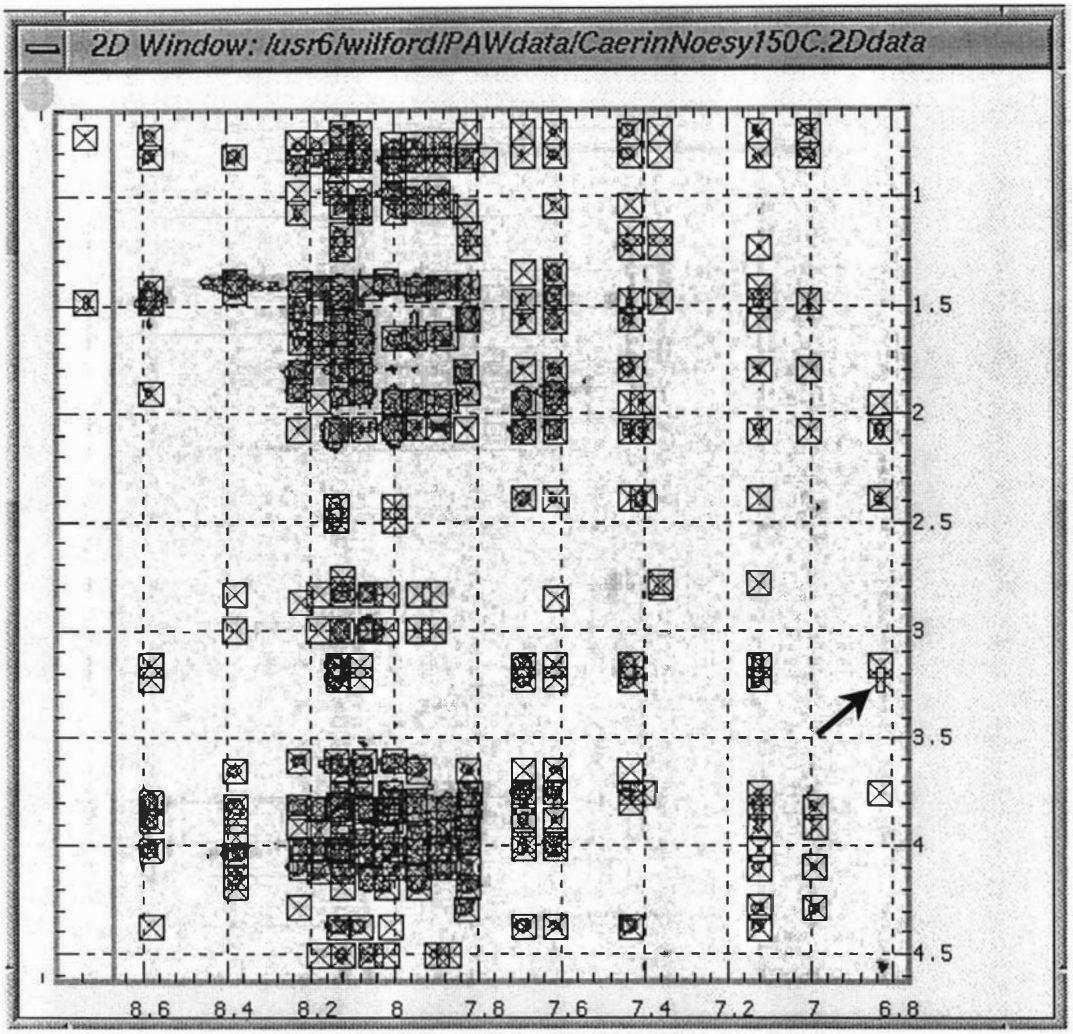


Figure 11.31 The same region as in the last figure, but with raw-peak symbols suppressed. The plot clearly shows that all of the strong cross-peaks have been fully assigned, except for one arrowed at around 3.2 ppm.

The plot gives a tidy overview that clearly shows the assigned and unassigned peaks. For example, the arrowed peak has a rectangle symbol, indicating that the assignment is incomplete.

The next figure displays the intra-residue cross-peaks of Q and K in the HN-HA region of the NOESY spectrum, which is obtained as follows:

- Type z7, then zs.
- Select the region as shown in the figure.
- Type ral to remove all lines.

From the *Peak-display Toolbox*, do the following:

- Switch on the [Intra.ByType] button.
- Switch on the [K] and [Q] buttons.

- Choose [Execute].

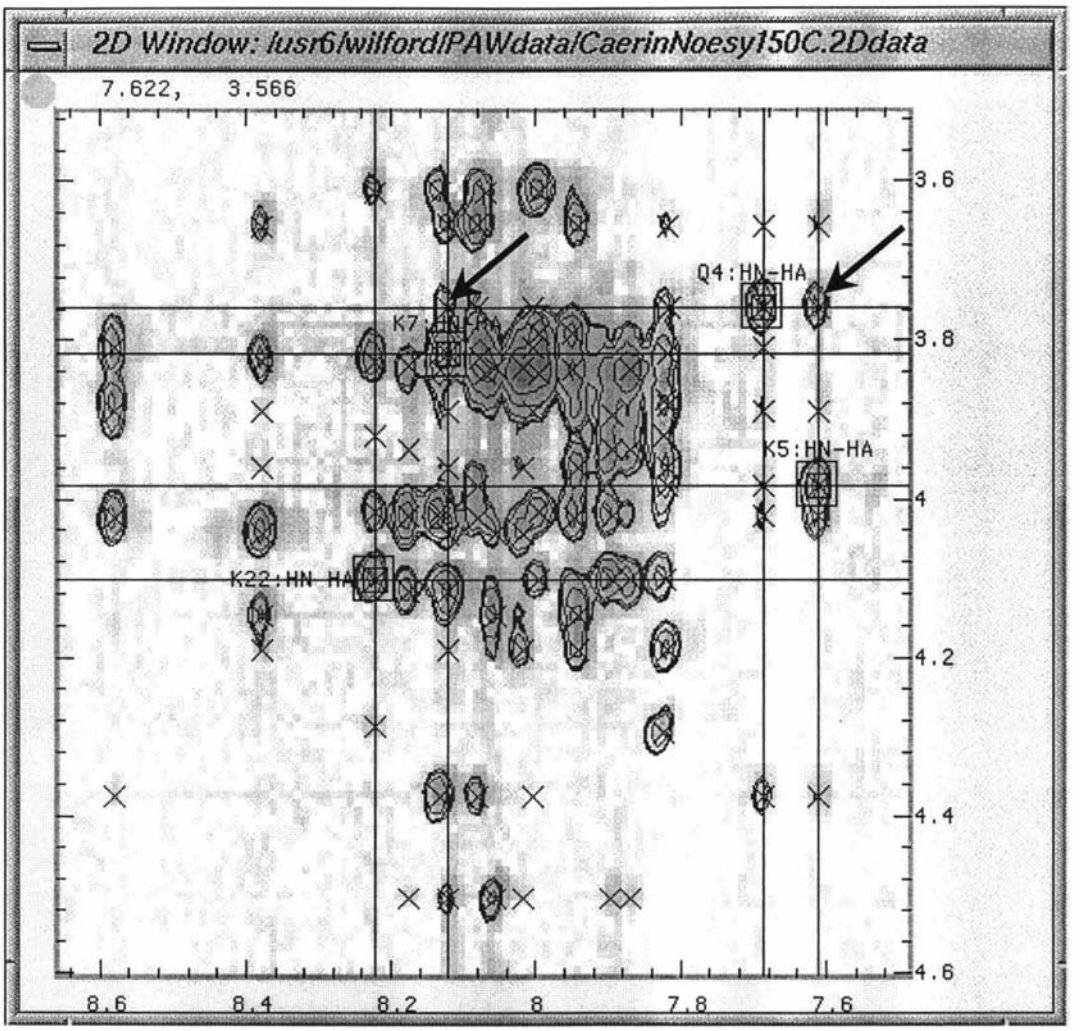
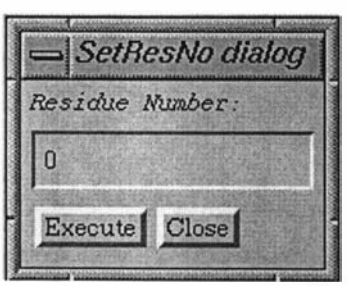


Figure 11.32 The intra-residue cross-peaks of Q and K residues in the HN-HA region of the NOESY150 spectrum. (Note that this is plotted at a higher first contour level in order to reduce the overlap.)

This plot allows the K5-related cross-peaks to be unambiguously identified.

Figure 11.34 displays all K5-cross-peaks in the HN-HA region of the NOESY spectrum, which is obtained as follows:



- Switch on the [CpksByNum] button from the *Peak-display Toolbox* to open the *SetResNo Dialog*, as shown below.
- Enter 5.
- Choose [Execute].

Figure 11.33 The SetResNo Dialog.

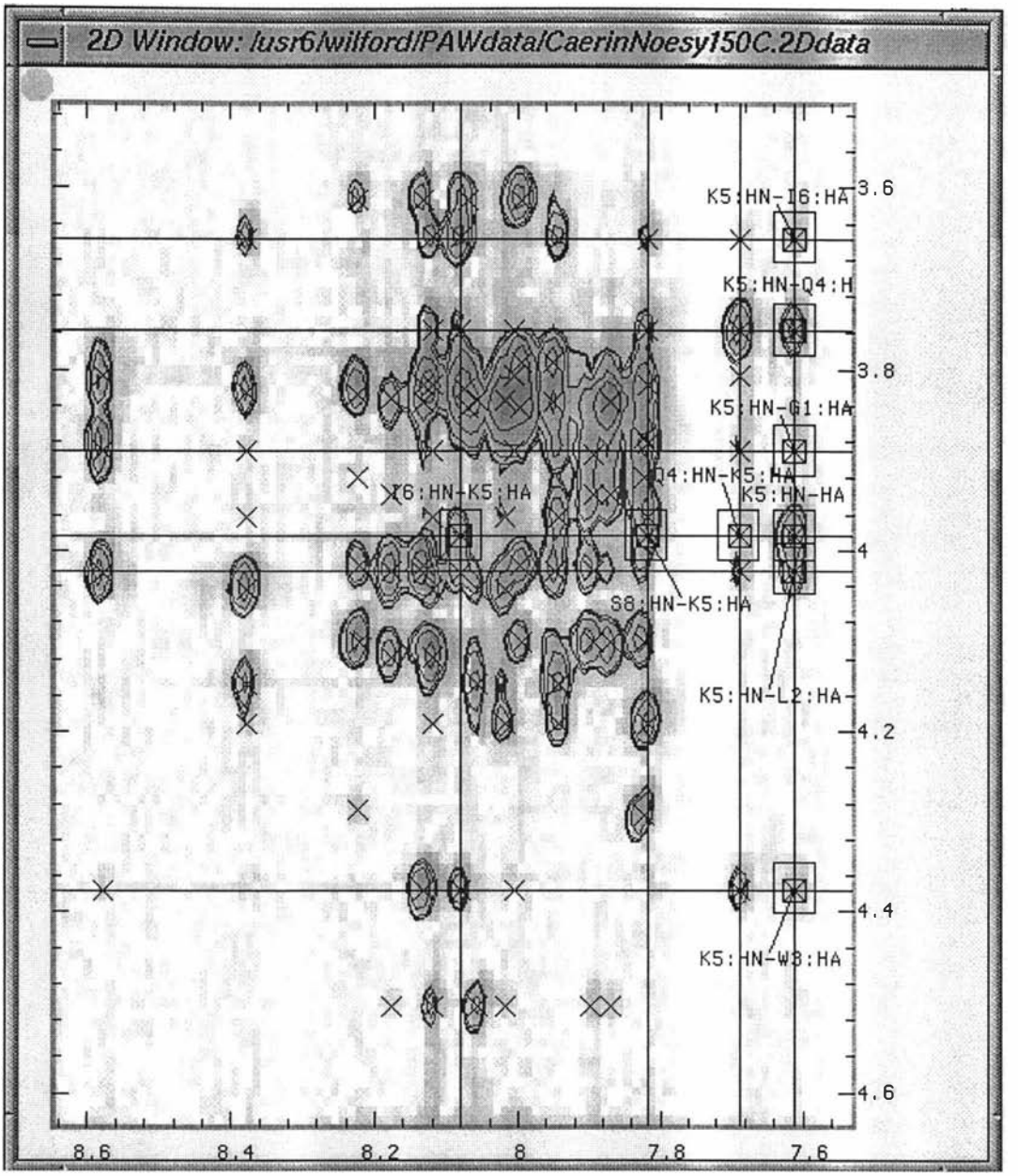
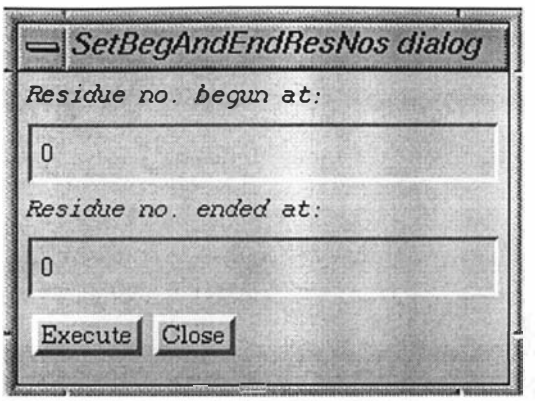


Figure 11.34 All K5-related cross-peaks in the HN-HA region of the NOESY150 spectrum.

This plot provides a clear view of the K5-connectivities and the helical structure between K5 and S8. The crosshairs and rectangles in the plot were drawn automatically for the display in this format to highlight the cross-peaks involved.

The next figure displays the NOE connectivities from L2 to S8 in the HN-HA region of the NOESY spectrum, which is obtained as follows:

- Switch off the [DspRPks] button from the *Peak-display Toolbox*.
- Switch on the [NA-seqConnectn] button to open the *SetBegAndEndResNo Dialog*, as shown below.



- Enter 2 and 8.
- Choose [Execute].

Figure 11.35 The SetBegAndEndResNo Dialog.

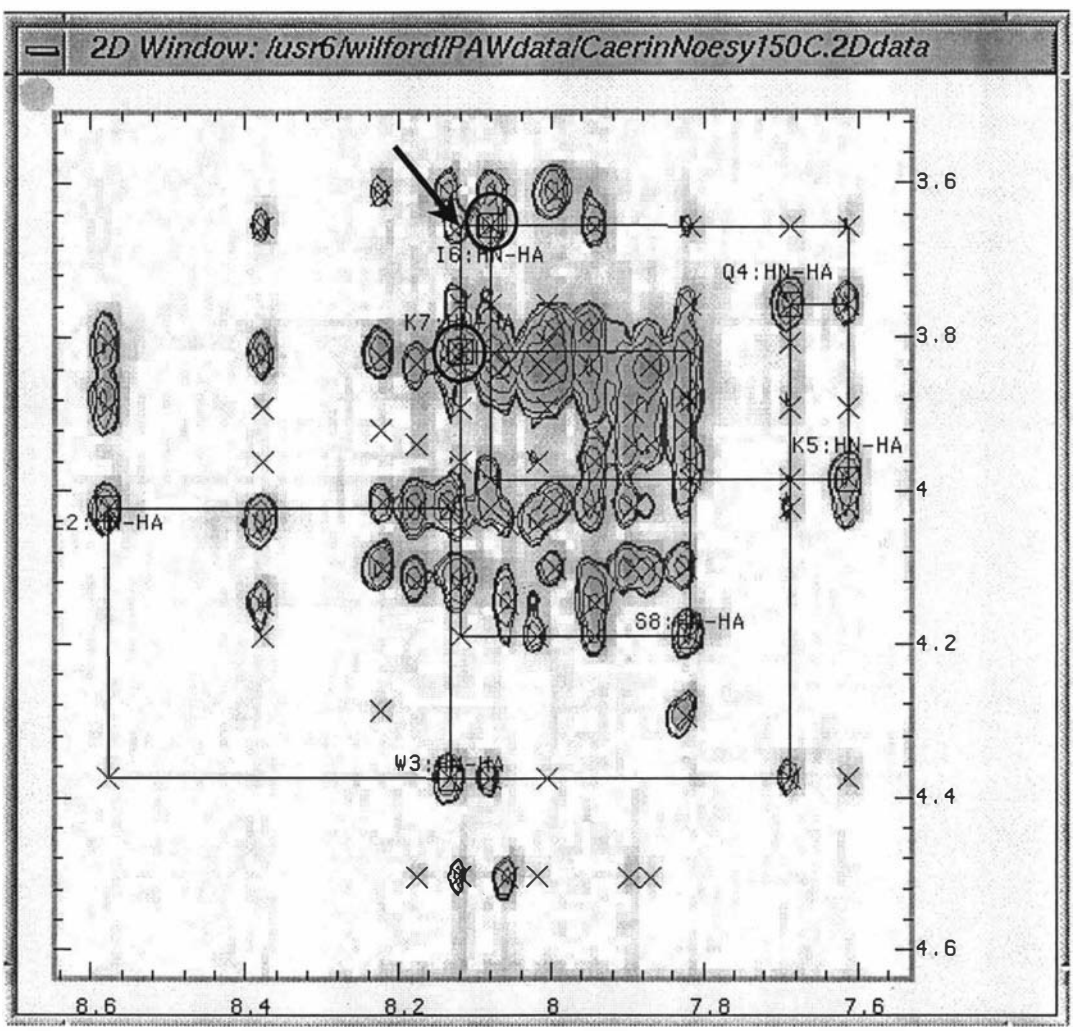


Figure 11.36 The NOE connectivities from L2 to S8 in the HN-HA region of the NOESY150 spectrum. The arrow shows the connectivities between I6:HN-HA and K7:HN-HA (circled).

This plot provides a clear view of the connectivities from L2 to S8 in the HN-HA region. The lines in the plot were drawn automatically for the display in this format to highlight the cross-peaks involved.

The next figure displays the NOE connectivities from L2 to S8 in the HN-HN region of the NOESY spectrum, which is obtained as follows:

- Switch off the [DspRPks] button from the *Peak-display Toolbox.*
- Switch on the [NN-seqConnectn] button to open the *SetBegAndEndResNo Dialog* (Figure 11.35).
- Enter 2 and 8.
- Choose [Execute].

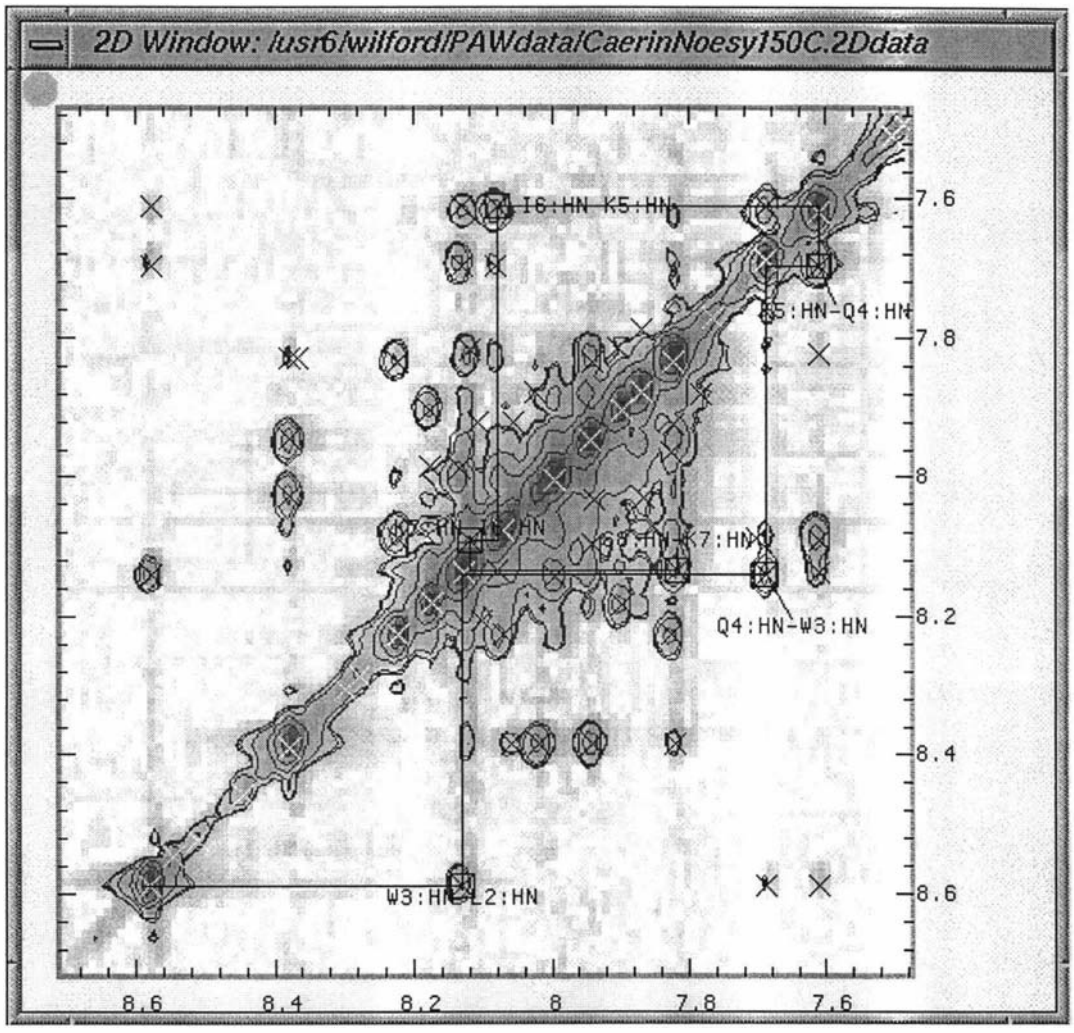


Figure 11.37 The NOE connectivities from L2 to S8 in the HN-HN region of the NOESY150 spectrum.

This plot provides a clear view of the connectivities from L2 to S8 in the HN-HN region. Again, the lines in the plot were drawn automatically for the display in this format to highlight the cross-peaks involved.

The next figure displays the NOE summary from G1 to S24, which is obtained as follows:

- Switch the label-display format to [None] from the *Peak-display Toolbox*.
- Switch on the [NOE Summary] button to open the *SetBegAndEndResNo Dialog* (Figure 11.35).
- Enter 1 and 24.
- Choose [Execute].
- Extend the window height.
- Type clr to clear the window.
- Type dr to draw.

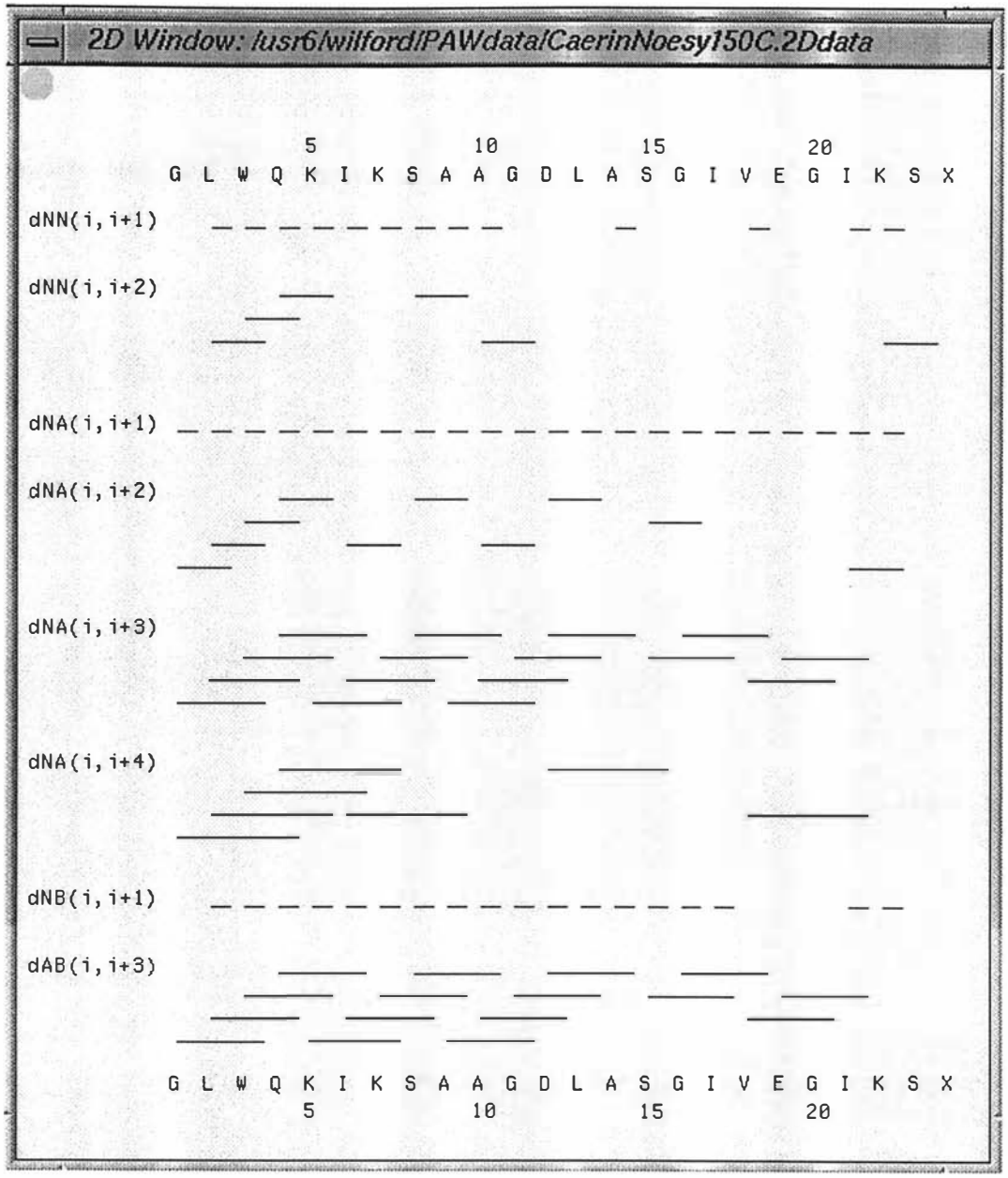


Figure 11.38 The NOE summary from G1 to X24. The notation in the left column correspond to the NOE distances listed in Table 7.5.

Note that when showing an NOE summary, the spectral display is switched off. To switch it back on, simply select any label display option then type dr.

11.6 Calculating Peak Volumes

The following example shows how to calculate a cluster of cross-peak volumes from a 2D spectrum:

- Click in the **CaerinNoesy150C** window.

- Type z7.
- Choose [CalPkVols] in the *Peak-assignment Toolbox* to open the *CalPkVol Dialog* (Figure 11.39). The message "*Use MouseBtn#1 to set a region for Calculating peak volumes.*" will be shown on top of the 2D draw-window.

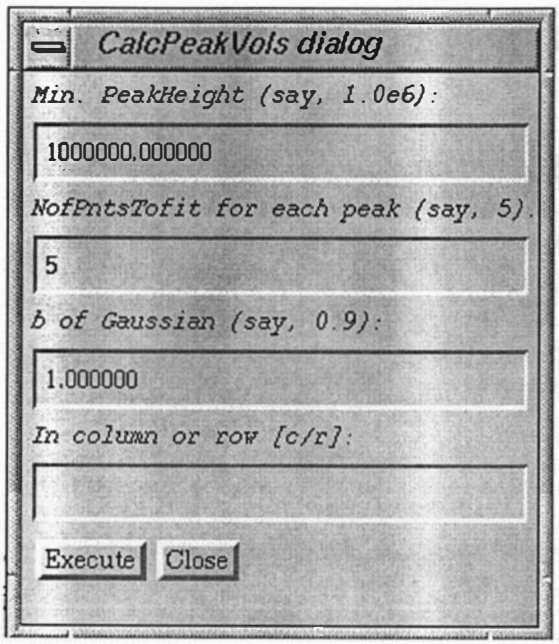


Figure 11.39 The *CalPkVol Dialog*.

- Select the region containing two columns of peaks as shown in Figure 11.40.
- Enter c for the last field so that the peak volumes will be calculated in column.
- Choose [Execute].

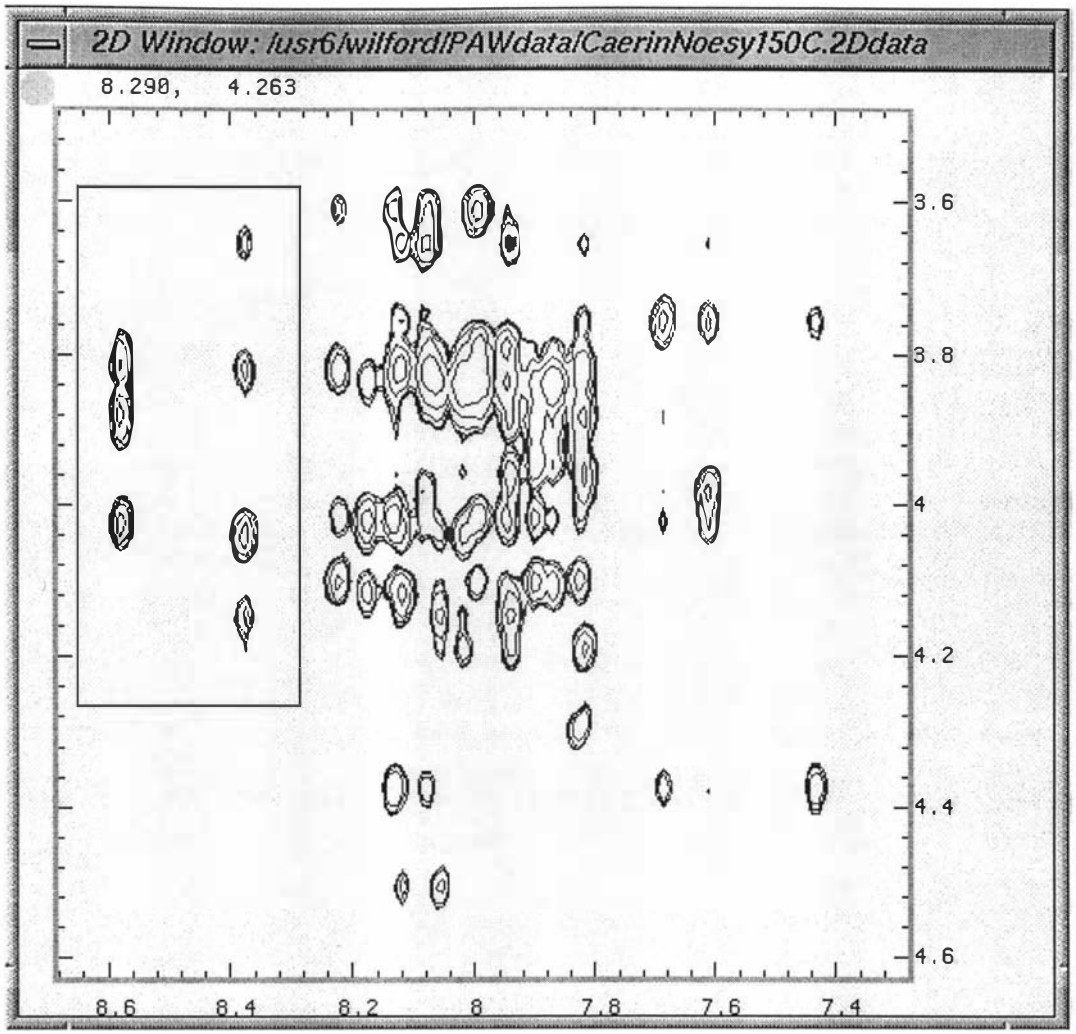


Figure 11.40 The NOE summary from G1 to X24. The notation in the left column correspond to the NOE distances listed in Table 7.5.

The calculated result will be reported on the Unix shell as follows:

```
LRTB= 8.61478 8.25404, 3.60996 4.29202
ColNo = 150

_NofSelectedPeaks = 3
Peak[0]: Height, Centre, b, Area, Vol =
         3.16293e+07 573.859 0.999863 1.12107e+08 3.97356e+08
Peak[1]: Height, Centre, b, Area, Vol =
         2.61563e+07 587.256 1.27944 1.18632e+08 5.38057e+08
Peak[2]: Height, Centre, b, Area, Vol =
         2.03069e+07 593.654 1.36886 9.85385e+07 4.78156e+08
ColNo = 169

_NofSelectedPeaks = 5
Peak[0]: Height, Centre, b, Area, Vol =
         1.68117e+06 559.012 1.27099 7.57459e+06 3.41277e+07
Peak[1]: Height, Centre, b, Area, Vol =
         8.51801e+06 562.742 1.01423 3.06252e+07 1.10108e+08
Peak[2]: Height, Centre, b, Area, Vol =
         3.40407e+07 572.468 1.00885 1.21739e+08 4.3537e+08
Peak[3]: Height, Centre, b, Area, Vol =
         9.74588e+06 592.799 1.0751 3.71427e+07 1.41555e+08
```

```
Peak[4]: Height, Centre, b, Area, Vol =
         4.99715e+06 608.413 0.982899 1.74115e+07 6.06666e+07
```

11.7 Exporting Peak Lists

The following example shows how to export a peak list:

- Type z7.
- Choose [ExpoPkLists] in the *Peak-assignment Toolbox* to open a save-file dialog (see Chapter 4).
- Enter a name (with the extension "**.PkList**") for the peak list to be saved as.
- Choose [Save] in the dialog.

The export file will be stored in the data directory.

Chapter 12:

Commands and Variables

12.1 Introduction

This chapter provides the macro variables and commands used by PAW. Section 12.2 presents a variables list sorted by name, along with a brief description for each variable. Section 12.3 provides a command list sorted by functionality, along with a brief description of the command. Whenever it is necessary, macro commands are also presented with syntaxes and examples. Section 12.4 provides a command list sorted by name.

12.2 The Macro Variable List

The following symbols are used for the variable types:

- B = Boolean type (F for false or T for true)
- C = character type
- D = integer type
- F = float type
- F{} = arrayed-float type
- S = character-string type

TYPE	Name	Description
D	%I	a loop variable
D	%J	a loop variable
D	%K	a loop variable
D	%L	a loop variable
D	%M	a loop variable
D	%N	a loop variable
S	bcCom1	Command for baseline correction in D1
S	bcCom2	Command for baseline correction in D2
S	BkgrColor	Background colour
D	BlockSize	Block size of an NMR data record
F{}	Buf0, Buf1, ...	1DBuf0, 1DBuf2, ... 1DBuf8
F{}	BslnBuf	The baseline buffer
S	Color01	Defined colour 1
S	Color02	Defined colour 2
S	Color03	Defined colour 3
S	Color04	Defined colour 4
S	Color05	Defined colour 5
S	Color06	Defined colour 6
S	Color07	Defined colour 7
S	Color08	Defined colour 8
S	Color09	Defined colour 9
S	Color10	Defined colour 10
S	Color11	Defined colour 11
S	Color12	Defined colour 12
S	Color13	Defined colour 13
S	Color14	Defined colour 14

S	Color15	Defined colour 15
S	Color16	Defined colour 16
S	Color17	Defined colour 17
S	Color18	Defined colour 18
C	ColorScheme	Colour-map scheme
S	CPeakColor	Cross-peak colour
D	Cur1DBufNo	Current 1DBuf No.
D	D1	Data size in D1
C	D1Struct	Data structure in D1
D	D2	Data size in D2
C	D2Struct	Data structure in D2
S	DataDir	Data directory
S	DataFile	Data filename
C	DataType	Data type
B	DebugOn	Flag to switch the debugging mode on/off
C	DomNoToTrans	The domain to transform, either 1, 2 or both
S	DPeakColor	Colour for the diagonal-peak symbols
S	DrawColor	Colour for the drawing objects
B	DrawOn	Flag to switch the drawing on/off
B	DspGrids	Flag to switch the grids on/off
B	DspInHz	Flag to switch the Hz-display on/off
B	DspInLog	Flag to switch the Log-display on/off
B	DspInPnt	Flag to switch the Pnt-display on/off
B	DspInPPB	Flag to switch the ppb-display on/off (NOT USED)
B	DspInPPM	Flag to switch the ppm-display on/off
D	EnoughMemory	Flag to indicate if there is enough memory (NOT USED)
D	Error	Flag to tell if there was an error in any operation
C	ExprmtType	Experiment type
C	FileFormat	File format
C	FileStruct	File structure
C	FileType	File type
F{}	FltBuf	The filter buffer
S	fltCom1	Command for apodisation in D1
S	fltCom2	Command for apodisation in D2
D	I0	Integer-type variable 0
D	I1	Integer-type variable 1
D	I2	Integer-type variable 2
D	I3	Integer-type variable 3
D	I4	Integer-type variable 4
D	I5	Integer-type variable 5
D	I6	Integer-type variable 6
D	I7	Integer-type variable 7
D	I8	Integer-type variable 8
D	I9	Integer-type variable 9
S	KeyB1F1	The combined key of MsBtn#1 and F1
S	KeyB1F2	The combined key of MsBtn#1 and F2
S	KeyB1F3	The combined key of MsBtn#1 and F3
S	KeyB1F4	The combined key of MsBtn#1 and F4

S	KeyB1F5	The combined key of MsBtn#1 and F5
S	KeyB1F6	The combined key of MsBtn#1 and F6
S	KeyB1F7	The combined key of MsBtn#1 and F7
S	KeyB1F8	The combined key of MsBtn#1 and F8
S	KeyB1F9	The combined key of MsBtn#1 and F9
S	KeyB1F10	The combined key of MsBtn#1 and F10
S	KeyB1F11	The combined key of MsBtn#1 and F11
S	KeyB1F12	The combined key of MsBtn#1 and F12
S	KeyB2F1	The combined key of MsBtn#2 and F1
S	KeyB2F2	The combined key of MsBtn#2 and F2
S	KeyB2F3	The combined key of MsBtn#2 and F3
S	KeyB2F4	The combined key of MsBtn#2 and F4
S	KeyB2F5	The combined key of MsBtn#2 and F5
S	KeyB2F6	The combined key of MsBtn#2 and F6
S	KeyB2F7	The combined key of MsBtn#2 and F7
S	KeyB2F8	The combined key of MsBtn#2 and F8
S	KeyB2F9	The combined key of MsBtn#2 and F9
S	KeyB2F10	The combined key of MsBtn#2 and F10
S	KeyB2F11	The combined key of MsBtn#2 and F11
S	KeyB2F12	The combined key of MsBtn#2 and F12
S	KeyB3F1	The combined key of MsBtn#3 and F1
S	KeyB3F2	The combined key of MsBtn#3 and F2
S	KeyB3F3	The combined key of MsBtn#3 and F3
S	KeyB3F4	The combined key of MsBtn#3 and F4
S	KeyB3F5	The combined key of MsBtn#3 and F5
S	KeyB3F6	The combined key of MsBtn#3 and F6
S	KeyB3F7	The combined key of MsBtn#3 and F7
S	KeyB3F8	The combined key of MsBtn#3 and F8
S	KeyB3F9	The combined key of MsBtn#3 and F9
S	KeyB3F10	The combined key of MsBtn#3 and F10
S	KeyB3F11	The combined key of MsBtn#3 and F11
S	KeyB3F12	The combined key of MsBtn#3 and F12
S	KeyF1	The function key F1
S	KeyF2	The function key F2
S	KeyF3	The function key F3
S	KeyF4	The function key F4
S	KeyF5	The function key F5
S	KeyF6	The function key F6
S	KeyF7	The function key F7
S	KeyF8	The function key F8
S	KeyF9	The function key F9
S	KeyF10	The function key F10
S	KeyF11	The function key F11
S	KeyF12	The function key F12
S	lpCom1	Command for linear prediction in D1
S	lpCom2	Command for linear prediction in D2
S	MacroDir	Macro directory
D	Max1DDim	Maximum size in D1

D	MD1	Matrix size in D1
D	MD2	Matrix size in D2
D	MD3	Matrix size in D3 (NOT USED)
S	NgtvCtourColor	Colour for negative contours
F	OneK	= 1024.0
S	phCom1	Command for phasing in D1
S	phCom2	Command for phasing in D2
F	PI	= 3.1415925...
D	ProjNo	Project number
S	PstvCtourColor	Colour for positive contours
S	RPeakColor	Colour for raw peaks
S	S0	String-type variable 0
S	S1	String-type variable 1
S	S2	String-type variable 2
S	S3	String-type variable 3
S	S4	String-type variable 4
S	S5	String-type variable 5
S	S6	String-type variable 6
S	S7	String-type variable 7
S	S8	String-type variable 8
S	S9	String-type variable 9
F	Sc1	Scale factor for 1D display
D	ScrnHeight	Screen height in pixels
D	ScrnWidth	Screen width in pixels
S	Sequence	String of the amino-acid sequence for a project
D	SiteNo	Site number
F	SpecFreq	Spectrometer frequency
F	SweepWidth	Spectral width
S	SystDir	System directory
S	TextColor	Colour for the text objects
F	V0	Float-type variable 0
F	V1	Float-type variable 1
F	V2	Float-type variable 2
F	V3	Float-type variable 3
F	V4	Float-type variable 4
F	V5	Float-type variable 5
F	V6	Float-type variable 6
F	V7	Float-type variable 7
F	V8	Float-type variable 8
F	V9	Float-type variable 9
F	XofCalibPnt	X-coordinate of the calibration point in points
F	XPPMofCalibPnt	X-coordinate of the calibration point in ppm
F	YofCalibPnt	Y-coordinate of the calibration point in points
F	YPPMofCalibPnt	Y-coordinate of the calibration point in ppm

12.3 The Command List sorted by functionality

Note:

- The three items in each line are command type, name and description. The type of a command can either be K (or I) for interactive-only, M for macro-only, and G for both.
- '1DBuf0' means 'the data set in 1D Buffer 0'.
- 'by entry' means 'using a dialog'.
- 'geom-pars' means 'the geometric parameters'.
- A 1D operation modifies the contents of 1DBuf0 if it is not specified.
- For 1D processing, see also the filter, phasing and baseline-correction commands.
- The commands dr, zf, zm and zs must follow either sw1 or sw2.

The following symbols are used for the command types:

- K = keyboard command
- M = macro-programming command
- G = both keyboard and macro-programming command

The following symbols are used for the parameter types:

- @ = character type
- % = integer type
- ~ = float type
- ~{} = arrayed float type
- $ = character-string type

➢ 1D-display commands

Type	Name	Description
K	cty	Switch the [CenterY] on/off
K	d1r	Switch the 1D [DspRealOnly] on/off
K	dbf	Switch the [DspBuf] on/off
K	dbl	Switch the [DspBsline] on/off
K	dft	Switch the [DspFilter] on/off
G	dr1	Draw 1D plot(s) Syntax: dr1
K	eql	Switch the [Equalise] on/off
K	ovl	Switch the [Overlap] on/off
K	xsh	Set XShift
K	ysc	Set YScale
K	ysh	Set YShift
G	z1d	Zoom into a 1D region Syntax: z1d
G	zf1	Zoom to 1D full-view plot(s) Syntax: zf1
K	zm1	Zoom to 1D mutiple-region plot(s)
K	zs1	Zoom to 1D single-region plot(s)

➢ 1D-processing commands

Type	Name	Description
G	add	Add a constant to 1DBuf0 Syntax: add
G	avr	Average every (2m+1) data Syntax: avr (%m,%NofTimes,@PartToAvrg) Example: avr (10, 5, r)
G	ch1	Change the size of 1D buffers Syntax: ch1 (%NofPnts) Example: ch1 (2048)
G	chk	Check if there is data loaded for a draw-window Syntax: chk (%DomNo) Example: chk (2)
G	ctc	Convert 1DBuf0 to its complex conjugate Syntax: ctc
G	ctp	Produce a power spectrum from 1DBuf0 Syntax: ctp
G	fft	Apply a fast Fourier transform Syntax: fft
G	flp	Reverse the order of the contents in 1DBuf0 Syntax: flp
G	ftg	Fit 1DBuf0 with Gaussian(s) Syntax: ftg (%NofPntsTofit) Example: ftg (5)
G	fws	f-domain water-signal suppression Syntax: fws (~WaterPkCentre,%NofPntsTofit) example: fws (511.9, 10)
G	gsp	Generate a 1D spectrum Syntax: gsp(~Freq) example: gsp (88.0)
G	hft	Apply a Hilbert transform Syntax: hft
G	htf	Apply a Hilbert transform (fast) Syntax: htf
G	hts	Apply a Hilbert transform (slow) Syntax: hts
G	ift	Apply an inverse fast Fourier transform Syntax: ift
G	imz	Assign zeros to the imaginary part Syntax: imz
G	lp	Apply linear prediction Syntax: lp (%FirstPntUsedForLP,%NofPntsUsedForLP, %NofPolesForLP, %NofPntsToPredict, %PartToPred,SegmToPred) example: lp (100, 412, 20, 288 r, t)

G	mul	Multiply 1DBuf0 by a constant Syntax: mul (~Const) example: mul (300.0)
G	ncb	Normalise current 1D buffers to 1.0e6 Syntax: ncb
G	neg	Negate 1DBuf0 Syntax: neg
K	ocb	Open the Calibration Dialog
G	ptc	Convert a power spectrum into complex Syntax: ptc
G	rft	Apply a real Fourier transform Syntax: rft
G	rmo	Remove an offset calculated from sbs Syntax: rmo
G	shc	Shift data circularly Syntax: shc (%NofPnts) example: shc (62)
G	shl	Shift data to the left Syntax: shl (%NofPnts) example: shl (62)
G	shr	Shift data to the right Syntax: shr (%NofPnts) example: shr (962)
G	tws	t-domain water-signal suppression Syntax: tws

➢ 2D-display commands

Type	Name	Description
K	d2r	Switch the 2D [DspRealOnly] on/off
G	dr2	Redraw 2D plot(s) Syntax: dr2
K	dz1	Set a 2DzmPtern to Zm#1
K	dz2	Set a 2DzmPtern to Zm#2
K	dz3	Set a 2DzmPtern to Zm#3
K	dz4	Set a 2DzmPtern to Zm#4
K	dz5	Set a 2DzmPtern to Zm#5
K	dz6	Set a 2DzmPtern to Zm#6
K	dz7	Set a 2DzmPtern to Zm#7
K	dz8	Set a 2DzmPtern to Zm#8
G	dzf	Define ZmFactor (in %) for zo and zi operations Syntax: dzf (~ZmFactor) example: dzf (20)
K	dzp	Define a 2DZmPternNo
K	plc	Define 2D contour parameters
K	rzp	Zoom to a defined 2DzmPtern by entry

K	z1	Zoom to a pattern set by Zm#1
K	z2	Zoom to a pattern set by Zm#2
G	z2d	Zoom into a 2D region Syntax: z2d
K	z3	Zoom to a pattern set by Zm#3
K	z4	Zoom to a pattern set by Zm#4
K	z5	Zoom to a pattern set by Zm#5
K	z6	Zoom to a pattern set by Zm#6
K	z7	Zoom to a pattern set by Zm#7
K	z8	Zoom to a pattern set by Zm#8
K	zd	Zoom down
G	zf2	Zoom to 2D full-view plot(s) Syntax: zf2
K	zi	Zoom in
K	zl	Zoom left
K	zm2	Zoom to 2D mutiple-region plot(s)
K	zn	Zoom to next pattern
K	zo	Zoom out
K	zp	Zoom previous pattern
K	zr	Zoom right
K	zs2	Zoom to 2D single-region plot(s)
K	zsp	Display the same pattern for next active draw-window
K	zt	Zoom to the transposed region
K	zu	Zoom up

➢ 2D-processing commands

Type	Name	Description
G	lc	Load a column from a 2D buffer Syntax: lc(%ColNo,@ColStruct,%BufNo) example: lc(100, c, 0)
K	lc0	Load a column from a 2D buffer to 1DBuf0
K	lc1	Load a column from a 2D buffer to 1DBuf1
K	lc2	Load a column from a 2D buffer to 1DBuf2
K	lc3	Load a column from a 2D buffer to 1DBuf3
K	lc4	Load a column from a 2D buffer to 1DBuf4
K	lc5	Load a column from a 2D buffer to 1DBuf5
K	lc6	Load a column from a 2D buffer to 1DBuf6
K	lc7	Load a column from a 2D buffer to 1DBuf7
K	lc8	Load a column from a 2D buffer to 1DBuf8
G	ldc	(same as lc)
G	ldr	(same as ld)
K	lp2	Load and process 2D NMR data
G	lr	Load a row from a 2D buffer Syntax: lr (%RowNo,@RowStruct,%BufNo) example: lr (100, c, 0)

K	lr0	Load a row from a 2D buffer to 1DBuf0
K	lr1	Load a row from a 2D buffer to 1DBuf1
K	lr2	Load a row from a 2D buffer to 1DBuf2
K	lr3	Load a row from a 2D buffer to 1DBuf3
K	lr4	Load a row from a 2D buffer to 1DBuf4
K	lr5	Load a row from a 2D buffer to 1DBuf5
K	lr6	Load a row from a 2D buffer to 1DBuf6
K	lr7	Load a row from a 2D buffer to 1DBuf7
K	lr8	Load a row from a 2D buffer to 1DBuf8
K	pc2	process 2D NMR data
G	wrc	Write a column to a 2D buffer Syntax: wrc (%ColNo,@ColStruct) example: wrc (100, r)
G	wrr	Write a row to a 2D buffer Syntax: wrr (%RowNo,@RowStruct) example: wrr (100, c)

➢ Baseline, base-level commands

Type	Name	Description
G	bc	Correct baseline Syntax: bc (@FuncType, %Degree) example: bc (p, 5)
K	s1l	Set 1D base level
K	s2l	Set 2D base level
G	sbs	Set 1D baseline segments (in points) and run s1l Syntax: sbs (%NofBslnSegms,%X1,%Y1,%X2,%Y2,...) example: sbs (2, 340,480, 960,1020) Note that the first segment is used to calculate 1D base-levels.

➢ Buffer-management commands

Type	Name	Description
G	axy	Add 1DBuf#Y to 1DBuf#X Syntax: axy (%BufNo1,%BufNo2) example: axy (0,1)
G	cr1	Create 1D buffers Syntax: cr1 (%NofPnts) example: cr1 (1024)
G	cr2	Create a 2D buffer Syntax: cr2 (%D1,%D2) example: cr2 (1024,1024)
G	lb0	Set 1dLoadingBuf = 0 Syntax: lb0
G	lb1	Set 1dLoadingBuf = 1 Syntax: lb1
G	lb2	Set 1dLoadingBuf = 2 Syntax: lb2
G	lb3	Set 1dLoadingBuf = 3 Syntax: lb3
G	lb4	Set 1dLoadingBuf = 4 Syntax: lb4
G	lb5	Set 1dLoadingBuf = 5 Syntax: lb5
G	lb6	Set 1dLoadingBuf = 6 Syntax: lb6
G	lb7	Set 1dLoadingBuf = 7 Syntax: lb7
G	lb8	Set 1dLoadingBuf = 8 Syntax: lb8
G	mxy	Calculate dot product of data in two 1D buffers Syntax: mxy (%BufNo1,%BufNo2) example: mxy (0,1)
G	sri	Swap the real and imaginary part in 1DBuf0 Syntax: sri
G	sxy	Swap 1DBufX with 1DBufY Syntax: sxy (%BufNo1,%BufNo2) example: sxy (0,1)
G	szd	halve the size of 1D buffers Syntax: szd
G	szu	double the size of 1D buffers Syntax: szu
G	wxy	Write the contents of 1DBufX to 1DBufY Syntax: wxy (%BufNo1,%BufNo2) example: wxy (0,1)

➢ Common-display commands

Type	Name	Description
K	dgr	Switch the [DspGrid] on/off
K	dlg	Switch the [DspInLog] on/off
G	dr	Redraw 1D plot(s) Syntax: dr
K	hz	Switch the [Hz] on/off
K	pnt	Switch the [Pnt] on/off
K	ppm	Switch the [ppm] on/off
G	zf	Zoom current window to full-view plot(s) Syntax: zf
K	zm	Zoom current window to mutiple-region plot(s)
K	zs	Zoom current window to single-region plot(s)

➢ Drawing commands

Type	Name	Description
K	dal	Draw a line
K	dar	Draw a rectangle
K	dch	Draw a crosshair
K	dtr	Draw a transposed rectangle
K	dvl	Draw a vertical line
M	lli	Load a defined line (in ppm) from a line list Syntax: lli (~X0,~Y0, ~X1,~Y1) example: lli (7.8,7.8, 7.8,4.3)
M	lri	Load a defined rectangle from a rectangle list Syntax: lri (~L,~T,~W,~H) example: lri (7.8,4.3, 3.5,3.5)
M	lti	Load a defined text-string from a text-string list Syntax: lti (~L,~T, "string") example: lti (7.8,4.3, "This is a peak.")
K	rll	Remove one line
K	rlr	Remove one rectangle
K	rad	Remove all drawing objects
K	ral	Remove all lines
K	rar	Remove all rectangles

➢ Export commands

Type	Name	Description
K	pl1	Open the Plot1D Dialog
K	pl2	Open the Plot2D Dialog
M	prb	Print a 1d buffer Syntax: prb ($BuferName) example: prb (Buf0)
M	prn	Print the result of a mathematical expression Syntax: prn ($expression) or prn ($variable) example: prn (3*(2+7)+5*(7+5/6)); prn (V1*V2)
M	prs	Print a text-string Syntax: prs ("string") or prs ($string) example: prs ("Process 1 ends."); prs (DataDir)
K	swb	Save the workbench to WBench_new.mcr

➢ File-management commands

Type	Name	Description
M	afs	Alter FileStruct to current setting Syntax: afs
M	fcl	Close a file Syntax: fcl (%FileNo)
M	fop	Open a file Syntax: fop ("path", "filename")
M	frd	Read a file Syntax: frd(%FileNo, %NofBytesToRead)
M	fsk	Jump to a location of a file Syntax: fsk (%FileNo, %NofBytesToSkip)
G	ld1	Load 1D data Syntax: ld1 ("path", "filename") example: ld1 ("", "LSZ.1Ddata")
G	ld2	Load 2D data Syntax: ld2 ("path", "filename") example: ld2 ("", "LSZ.2Ddata)
K	ocv	Open the File-conversion Dialog
G	run	run a macro Syntax: run "macro filename") example: run MyData.2Dparm
G	sd1	Save 1D data Syntax: sd1 ("path", "filename") example: sd1 ("", "LSZ.1Ddata)
G	sd2	Save 2D data Syntax: sd2 ("path", "filename") example: sd2 ("", "LSZ.2Ddata")

➢ Filter commands

Type	Name	Description
G	cs	Apply a cosine filter Syntax: cs (~EndPoint) example: cs (512)
G	cs2	Apply a squared-cosine filter Syntax: cs2 (~EndPoint) example: cs2 (512)
G	esb	RunSetEnhSineBell Syntax: esb (~EndPoint,~ShiftAngle,~n,~a) example: esb (1024, 60, 2, -100)
G	exp	Apply a exponential filter Syntax: exp (~a) example: exp (200)
G	gau	Apply a Gaussian filter Syntax: gau (~a, ~b) example: gau (100, 500)
G	hm	Apply a Hamming filter Syntax: hm (~a) example: hm (???)
G	hn	Apply a Hanning filter Syntax: hh (~EndPoint,~ShiftAngle) example: hn (???)
G	sb	Apply a sinebell filter Syntax: sb (~EndPoint,~ShiftAngle) example: sb (1024, 60)
G	sb2	Apply a squared-sine-bell filter Syntax: sb (~EndPoint,~ShiftAngle) example: sb (1024, 60)
G	xp	(same as exp)
G	xp0	(same as xp but is ended with zero)

➢ Macro-directive commands

Type	Name	Description
M	brk	= 'break' that breaks a condition block Syntax: brk
M	eif	= 'else-if' that starts an else-if-condition block Syntax: eif (<condition>) Example: eif (V0>3)
M	els	= 'else' that starts an else-condition block Syntax: els
M	end	= 'end' that ends a block Syntax: end
M	for	= 'for' that starts a for-loop Syntax: for (%LoopVariable = %Beg, %End, %Step) example: for (%I=0, 1023, 1)

M	if	= 'if' that starts an if-condition block Syntax: if (<condition> Example: if (Error)
M	nex	= 'next' of a for-loop Syntax: nex

➢ Menu-bar commands

Type	Name	Description
M	bmb	Build menu-button Syntax: bmb ("label", @UnderlinedChar, $HotKey, "Callback indicator") Example: bmb ("Load1dData", 1, Alt+1, "com: ld1") (See also Chapter 3.)
M	bme	End a build-menu block Syntax: bme
M	bmm	Start a build-main-menu block Syntax: bmm
M	bsm	Start a build-sub-menu block Syntax: bsm

➢ Peak-assignment commands

Type	Name	Description
K	cpa	Assign cross-peaks in a group
K	cpc	Copy one cross-peak
K	cpe	Edit one cross peak
K	cpl	Line up cross-peaks
K	cpv	calculate cross-peak volumes from a 2D spectrum
K	tpe	Edit a transposed cross-peak

➢ Peak-display commands

Type	Name	Description
K	dpl	Display peak lists
K	zss	Zoom to a multi-region plot that shows all peaks of a spin-system in an NMR spectrum

➢ Peak-picking commands

Type	Name	Description
K	cpp	Pick cross-peaks in a region
K	cpr	Remove cross-peaks in a region
K	dpp	Pick diagonal peaks in a region
K	dpr	Remove one diagonal peak
K	rp1	Pick one raw peak
K	rpc	Count the number of raw-peaks in a region
K	rpp	Pick raw-peaks in a region
K	rpr	Remove raw-peaks
K	tpf	Find a transposed peak

➢ Peak-list-management commands

Type	Name	Description
K	epl	Export the assigned peak records
M	lpi	Load a defined peak from a peak list Syntax 1: lpi (~X,~Y) Syntax 2: lpi (~X,~Y, ResidueCode1,ResNo1,AtomCode1, ResidueCode2,ResNo2,AtomCode2, LabelDistX, LabelDistY, SpectralFlag, Notes, Intensity) Examples: lpi (7.832, 4.371) Lpi (7.610,3.759, "K",5,"HN", "Q",4,"HA", 40,40, "strong", 1.234e8)
K	lpl	Load a set of three co-related peak lists, including the raw-peak, diagonal peak and cross-peak lists.
K	mlb	Move a peak label
K	sp1	Save peak lists sorted by ResidueNo in D1
K	sp2	Save peak lists sorted by ResidueNo in D2
K	sp3	Save peak lists sorted by D1-coordinate in ppm
K	sp4	Save peak lists sorted by D2-coordinate in ppm
K	spl	(same as spl)

➢ Phasing commands

Type	Name	Description
G	ph	Set zeroth- and first-order phase Syntax: ph (~Ph0,~Ph1,~Pivot) example: ph (88.0, -219.0, 0.0)
G	ph0	Set zeroth-order phase Syntax: ph0 (~Ph0) example: ph0 (-88.0)
G	ph1	Set first-order phase Syntax: ph1 (~Ph1) example: ph1 (-219)
G	phb	Set Bruker ph1 Syntax: phb (~Ph1) example: phb (78)
G	pvt	Set pivot Syntax: pvt (~Pivot) example: pvt (0.0)

➢ Toolbox-management commands

Type	Name	Description
M	g1d	Set geom-pars of the 1D-display Toolbox Syntax: g1d (%L,%T,%W,%H) example: g1d (1172,58,100,540)

M	g1m	Set geom-pars of the 1D-MiscProc Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	g1p	Set geom-pars of the 1D-processing Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	g2d	Set geom-pars of the 2D-display Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	g2p	Set geom-pars of the 2D-processing Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gbc	Set geom-pars of the Baseline-corr. Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gcd	Set geom-pars of the Common-display Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gdr	Set geom-pars of the Drawing Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gft	Set geom-pars of the Filter Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gpa	Set geom-pars of the Peak-assignment Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gpd	Set geom-pars of the Peak-display Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gph	Set geom-pars of the Phasing Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
M	gpp	Set geom-pars of the Peak-picking Toolbox Syntax: (same as that for g1d) example: (see that for g1d)
G	o1d	Open the 1D-display Toolbox Syntax: o1d
G	o1m	Open the 1D-MiscProc Toolbox Syntax: o1m
G	o1p	Open the 1D-processing Toolbox Syntax: o1p
G	o2d	Open the 2D-display Toolbox Syntax: o2d
G	o2p	Open the 2D-processing Toolbox Syntax: o2p

G	obc	Open the Baseline-correction Toolbox Syntax: obc
G	ocd	Open the Common-display Toolbox Syntax: ocd
G	odr	Open the Drawing Toolbox Syntax: odr
G	oft	Open the Filter Toolbox Syntax: oft
G	opa	Open the Peak-assignment Toolbox Syntax: opa
G	opd	Open the Peak-display Toolbox Syntax: opd
G	oph	Open the Phasing Toolbox Syntax: oph
G	opp	Open the Peak-picking Toolbox Syntax: opp
K	s1d	Reset parameters in the 1D-display Toolbox
K	s1m	Reset parameters in the 1D-MiscProc Toolbox
K	s1p	Reset parameters in the 1D-processing Toolbox
K	s2d	Reset parameters in the 2D-display Toolbox
K	s2p	Reset parameters in the 1D-processing Toolbox
G	x1d	Close the 1D-display Toolbox Syntax: x1d
G	x1m	Close the 1D-MiscProc Toolbox Syntax: x1m
G	x1p	Close the 1D-processing Toolbox Syntax: x1p
G	x2d	Close the 2D-display Toolbox Syntax: x2d
G	x2p	Close the 2D-processing Toolbox Syntax: x2p
G	xab	Close all toolboxes Syntax: xab
G	xbc	Close Baseline-correction Toolbox Syntax: xbc
G	xcd	Close Common-display Toolbox Syntax: xcd
G	xdr	Close Drawing Toolbox Syntax: xdr
G	xft	Close Filter Toolbox Syntax: xft
G	xpa	Close Peak-assignment Toolbox Syntax: xpa
G	xpd	Close Peak-display Toolbox Syntax: xpd

G	xph	Close Phasing Toolbox Syntax: xph
G	xpp	Close Peak-picking Toolbox Syntax: xpp

➢ Window-management commands

Type	Name	Description
K	clr	Clear the display in a draw-window
K	cws	Copy the structure of a draw-window
K	g1w	Set geom-pars of 1D draw-windows
K	g2v	Set geom-pars of 2D view-windows
K	g2w	Set geom-pars of 2D draw-windows
G	o1w	Open a 1D draw-window Syntax: o1w
G	o2v	Open the 2D view-window Syntax: o2v
G	o2w	Open a 2D draw-window Syntax: o2w
K	sbg	Set the background colour of a draw-window Syntax: sbg (%Red, %Green, %Blue) Syntax: Green example: sbg (228, 228, 228)
K	scm	Set colour map
G	sw1	Set the 1D active window to be the current window Syntax: sw1
G	sw2	Set the 2D active window to be the current window Syntax: sw2
G	x1w	Close a 1D Draw-window Syntax: x1w
G	x2v	Close the 2D view-window Syntax: x2v
G	x2w	Close a 2D Draw-window Syntax: x2w

➢ Miscellaneous commands

Type	Name	Description
M	lci	Read the Chemical-shift Database Syntax: lci (@Residue, $Atom, ~Helix, ~Beta, ~Coil, $Note) Example: lci (G, "HN", 8.07,8.44,8.36, "Wishart 91")
K	rsc	Reset initial colour setting

12.4 The Command List in alphabetical order

Type	Name	Description
G	add	Add a constant to 1DBuf0
M	afs	Alter FileStruct to current setting
G	avr	Produce an average curve from 1DBuf0
G	axy	Add 1DBuf#Y to 1DBuf#X
G	bc	Correct baseline
M	bmb	Start a build-menu block
M	bme	End a build-menu block
M	bmm	Build main menu
M	brk	= 'break' that breaks a condition block
M	bsm	build sub-menu
G	ch1	Change the size of 1D buffers by entry
G	chk	Check if there is data loaded for a draw-window
K	clr	Clear the display in a draw-window
K	cpa	Assign cross-peaks in a group
K	cpc	Copy one cross-peak
K	cpe	Edit one cross peak
K	cpl	Line up cross-peaks
K	cpv	Calculate cross-peak volumes from a 2D spectrum
K	cpp	Pick cross-peaks in a region
K	cpr	Remove cross-peaks in a region
G	cr1	Create 1D buffers
G	cr2	Create a 2D buffer
G	cs	Apply a cosine filter
G	cs2	Apply a squared-cosine filter
G	ctc	Convert 1DBuf0 to its complex conjugate
G	ctp	Produce a power spectrum from 1DBuf0
K	cty	Switch the [CenterY] on/off
K	cws	Copy the structure of a draw-window
K	d1r	Switch the 1D [DspRealOnly] on/off
K	d2r	Switch the 2D [DspRealOnly] on/off
K	dal	Draw a line
K	dar	Draw a rectangle
K	dbf	Switch the [DspBuf] on/off
K	dbl	Switch the [DspBsline] on/off
K	dch	Draw a crosshair
K	dft	Switch the [DspFilter] on/off
K	dgr	Switch the [DspGrid] on/off
K	dlg	Switch the [DspInLog] on/off
K	dpl	Display peak lists
K	dpp	Pick diagonal peaks in a region
K	dpr	Remove one diagonal peak
G	dr	Redraw 1D plot(s)
G	dr1	Draw 1D plot(s)
G	dr2	Redraw 2D plot(s)

K	dtr	Draw a transposed rectangle
K	dvl	Draw a vertical line
K	dz1	Set a 2DzmPtern to Zm#1
K	dz2	Set a 2DzmPtern to Zm#2
K	dz3	Set a 2DzmPtern to Zm#3
K	dz4	Set a 2DzmPtern to Zm#4
K	dz5	Set a 2DzmPtern to Zm#5
K	dz6	Set a 2DzmPtern to Zm#6
K	dz7	Set a 2DzmPtern to Zm#7
K	dz8	Set a 2DzmPtern to Zm#8
G	dzf	Define ZmFactor for zo and zi operations
K	dzp	Define a 2DZmPternNo
M	eif	= 'else-if' that starts an else-if-condition block
M	els	= 'else' that starts an else-condition block
M	end	= 'end' that ends a block
K	epl	Export the assigned peak records
K	eql	Switch the [Equalise] on/off
G	esb	RunSetEnhSineBell
G	exp	Apply a exponential filter
M	fcl	Close a file
G	fft	Apply a fast Fourier transform
G	flp	Reverse the order of the contents in 1DBuf0
M	fop	Open a file
M	for	= 'for' that starts a for-loop
M	frd	Read a file
M	fsk	Jump to a location of a file
G	ftg	Fit 1DBuf0 with Gaussian(s)
G	fws	f-domain water-signal suppression
M	g1d	Set geom-pars of the 1D-display Toolbox
M	g1m	Set geom-pars of the 1D-MiscProc Toolbox
M	g1p	Set geom-pars of the 1D-processing Toolbox
K	g1w	Set geom-pars of 1D draw-windows
M	g2d	Set geom-pars of the 2D-display Toolbox
M	g2p	Set geom-pars of the 2D-processing Toolbox
K	g2v	Set geom-pars of 2D view-windows
K	g2w	Set geom-pars of 2D draw-windows
G	gau	Apply a Gaussian filter
M	gbc	Set geom-pars of the Baseline-corr. Toolbox
M	gcd	Set geom-pars of the Common-display Toolbox
M	gdr	Set geom-pars of the Drawing Toolbox
M	gft	Set geom-pars of the Filter Toolbox
M	gpa	Set geom-pars of the Peak-assignment Toolbox
M	gpd	Set geom-pars of the Peak-display Toolbox
M	gph	Set geom-pars of the Phasing Toolbox
M	gpp	Set geom-pars of the Peak-picking Toolbox
G	gsp	Generate a 1D spectrum
G	hft	Apply a Hilbert transform
G	hm	Apply a Hamming filter

G	hn	Apply a Hanning filter
G	htf	Apply a Hilbert transform (fast)
G	hts	Apply a Hilbert transform (slow)
K	hz	Switch the [Hz] on/off
M	if	= 'if' that starts an if-condition block
G	ift	Apply an inverse fast Fourier transform
G	imz	Assign zeros to the imaginary part
G	lb0	Set 1dLoadingBuf = 0
G	lb1	Set 1dLoadingBuf = 1
G	lb2	Set 1dLoadingBuf = 2
G	lb3	Set 1dLoadingBuf = 3
G	lb4	Set 1dLoadingBuf = 4
G	lb5	Set 1dLoadingBuf = 5
G	lb6	Set 1dLoadingBuf = 6
G	lb7	Set 1dLoadingBuf = 7
G	lb8	Set 1dLoadingBuf = 8
G	lc	Load a column from a 2D buffer
K	lc0	Load a column from a 2D buffer to 1DBuf0
K	lc1	Load a column from a 2D buffer to 1DBuf1
K	lc2	Load a column from a 2D buffer to 1DBuf2
K	lc3	Load a column from a 2D buffer to 1DBuf3
K	lc4	Load a column from a 2D buffer to 1DBuf4
K	lc5	Load a column from a 2D buffer to 1DBuf5
K	lc6	Load a column from a 2D buffer to 1DBuf6
K	lc7	Load a column from a 2D buffer to 1DBuf7
K	lc8	Load a column from a 2D buffer to 1DBuf8
M	lci	Read the Chemical-shift Database
G	ld1	Load 1D data
G	ld2	Load 2D data
G	ldc	(same as lc)
G	ldr	(same as ld)
M	lli	Load a defined line from a line list
G	lp	Apply linear prediction
K	lp2	Load and process 2D NMR data
M	lpi	Load a defined peak from a peak list
K	lpl	Load a set of peak lists
G	lr	Load a row from a 2D buffer
K	lr0	Load a row from a 2D buffer to 1DBuf0
K	lr1	Load a row from a 2D buffer to 1DBuf1
K	lr2	Load a row from a 2D buffer to 1DBuf2
K	lr3	Load a row from a 2D buffer to 1DBuf3
K	lr4	Load a row from a 2D buffer to 1DBuf4
K	lr5	Load a row from a 2D buffer to 1DBuf5
K	lr6	Load a row from a 2D buffer to 1DBuf6
K	lr7	Load a row from a 2D buffer to 1DBuf7
K	lr8	Load a row from a 2D buffer to 1DBuf8
M	lri	Load a defined rectangle from a rectangle list
M	lti	Load a defined text-string from a text-string list

K	mlb	Move a peak label
G	mul	Multiply 1DBuf0 by a constant
G	mxy	Multiply dot product of data in two 1D buffers
G	ncb	Normalise current 1D buffers to 1.0e6
G	neg	Negate 1DBuf0
M	nex	= 'next' of a for-loop
G	o1d	Open the 1D-display Toolbox
G	o1m	Open the 1D-MiscProc Toolbox
G	o1p	Open the 1D-processing Toolbox
G	o1w	Open a 1D draw-window
G	o2d	Open the 2D-display Toolbox
G	o2p	Open the 2D-processing Toolbox
G	o2v	Open the 2D view-window
G	o2w	Open a 2D draw-window
G	obc	Open the Baseline-correction Toolbox
K	ocb	Open the Calibration Dialog
G	ocd	Open the Common-display Toolbox
K	ocv	Open the File-conversion Dialog
G	odr	Open the Drawing Toolbox
G	oft	Open the Filter Toolbox
G	opa	Open the Peak-assignment Toolbox
G	opd	Open the Peak-display Toolbox
G	oph	Open the Phasing Toolbox
G	opp	Open the Peak-picking Toolbox
K	ovl	Switch the [Overlap] on/off
K	pc2	process 2D NMR data
G	ph	Set zeroth- and first-order phase
G	ph0	Set zeroth-order phase
G	ph1	Set first-order phase
G	phb	Set Bruker ph1
K	pl1	Open the Plot1D Dialog
K	pl2	Open the Plot2D Dialog
K	plc	Define 2D contour parameters
K	pnt	Switch the [Pnt] on/off
K	ppm	Switch the [ppm] on/off
M	prb	Print a 1d buffer
M	prn	Print the result of a mathematical expression
M	prs	Print a text-string
G	ptc	Convert a power spectrum into complex
G	pvt	Set pivot
K	r1l	Remove one line
K	r1r	Remove one rectangle
K	rad	Remove all drawing objects
K	ral	Remove all lines
K	rar	Remove all rectangles
G	rft	Apply a real Fourier transform
G	rmo	Remove an offset calculated from sbs
K	rp1	Pick one raw peak

K	rpc	Count the number of raw-peaks in a region
K	rpp	Pick raw-peaks in a region
K	rpr	Remove raw-peaks
K	rsc	Reset initial colour setting
G	run	run a macro
K	rzp	Zoom to a defined 2DzmPtern by entry
K	s1d	Reset parameters in the 1D-display Toolbox
G	s1l	Set 1D base level
K	s1m	Reset parameters in the 1D-MiscProc Toolbox
K	s1p	Reset parameters in the 1D-processing Toolbox
K	s2d	Reset parameters in the 2D-display Toolbox
G	s2l	Set 2D base level
K	s2p	Reset parameters in the 1D-processing Toolbox
G	sb	Apply a sinebell filter
G	sb2	Apply a squared-sine-bell filter
K	sbg	Set the background colour of a draw-window
G	sbs	Set 1D baseline segments and run s1l
K	scm	Set colour map
G	sd1	Save 1D data
G	sd2	Save 2D data
G	shc	Shift data circularly
G	shl	Shift data to the left
G	shr	Shift data to the right
K	sp1	Save peak lists sorted by ResidueNo in D1
K	sp2	Save peak lists sorted by ResidueNo in D2
K	sp3	Save peak lists sorted by D1-coordinate in ppm
K	sp4	Save peak lists sorted by D2-coordinate in ppm
K	spl	(same as spl)
G	sri	Swap the real and imaginary part in 1DBuf0
G	sw1	Set the 1D active window to be the current window
G	sw2	Set the 2D active window to be the current window
K	swb	Save the workbench to WBench_new.mcr
G	sxy	Swap 1DBufX with 1DBufY
G	szd	half the size of 1D buffers
G	szu	double the size of 1D buffers
K	tpe	Edit a transposed cross-peak
K	tpf	Find a transposed peak
G	tws	t-domain water-signal suppression
G	wrc	Write a column to a 2D buffer
G	wrr	Write a row to a 2D buffer
G	wxy	Write the contents of 1DBufX to 1DBufY
G	x1d	Close the 1D-display Toolbox
G	x1m	Close the 1D-MiscProc Toolbox
G	x1p	Close the 1D-processing Toolbox
G	x1w	Close a 1D Draw-window
G	x2d	Close the 2D-display Toolbox
G	x2p	Close the 2D-processing Toolbox
G	x2v	Close the 2D view-window

G	x2w	Close a 2D Draw-window
G	xab	Close all toolboxes
G	xbc	Close Baseline-correction Toolbox
G	xcd	Close Common-display Toolbox
G	xdr	Close Drawing Toolbox
G	xft	Close Filter Toolbox
G	xp	(same as exp)
G	xp0	Apply a exponential filter that ended with zero
G	xpa	Close Peak-assignment Toolbox
G	xpd	Close Peak-display Toolbox
G	xph	Close Phasing Toolbox
G	xpp	Close Peak-picking Toolbox
K	xsh	Set XShift
K	ysc	Set YScale
K	ysh	Set YShift
K	z1	Zoom to a pattern set by Zm#1
G	z1d	Zoom into a 1D region
K	z2	Zoom to a pattern set by Zm#2
G	z2d	Zoom into a 2D region
K	z3	Zoom to a pattern set by Zm#3
K	z4	Zoom to a pattern set by Zm#4
K	z5	Zoom to a pattern set by Zm#5
K	z6	Zoom to a pattern set by Zm#6
K	z7	Zoom to a pattern set by Zm#7
K	z8	Zoom to a pattern set by Zm#8
K	zd	Zoom down
G	zf	Zoom current window to full-view plot(s)
G	zf1	Zoom to 1D full-view plot(s)
G	zf2	Zoom to 2D full-view plot(s)
K	zi	Zoom in
K	zl	Zoom left
K	zm	Zoom current window to mutiple-region plot(s)
K	zm1	Zoom to 1D mutiple-region plot(s)
K	zm2	Zoom to 2D mutiple-region plot(s)
K	zn	Zoom to next pattern
K	zo	Zoom out
K	zp	Zoom previous pattern
K	zr	Zoom right
K	zs	Zoom current window to single-region plot(s)
K	zs1	Zoom to 1D single-region plot(s)
K	zs2	Zoom to 2D single-region plot(s)
K	zsp	Display the same pattern for next active draw-window
K	zss	Zoom to a multi-region plot that shows all peaks of a spin-system in an NMR spectrum
K	zt	Zoom to the transposed region
K	zu	Zoom up