

Protected: Processing 2D ^1H - ^{15}N HSQC

12/11/00 – Steve Hardies, incorporating commentary by Andy Hinck.

Introduction

This document describes processing 2D ^1H ^{15}N HSQC data using the nmrPipe program system. The input is a dataset created by topspin and residing in an nmr data directory on amx-500 on drive /hinck or /nall. Although it is possible to do 2D data processing with topspin, this document describes an alternative program that resides on the NIS network of silicon graphics computers at the Center for Structural Biology. Use of this system does not require scheduling the AMX-500 itself. This document describes how to access the data from the amx-500, apply a series of corrections for anomalies in the data, and Fourier transform the data in both dimensions. The result is a 2D graphic display of nmr peaks — hopefully one peak per each amide hydrogen in your protein. A variety of steps to improve the quality of the display by reprocessing it will be described.

A skeleton of processing steps is given by numbered bold face statements.

Explanation, options, and possible problems are described in regular face. Commands given by typing are in *italic*.

Environment.

b>1. Log on to a computer in the NIS system.

There are currently 5 silicon graphics computers you can access in the computer room adjacent to the faculty office suite on the 5th floor of the Allied Health building. These are

the ones with the blue boxes. You can access the same programs and data directories from any of these computers that is available. The network that allows cross operability of these computers is referred to as the NIS network. There are plans to purchase additional computers to place on the network that will be physically located on the main campus, but at this time you have to go to the Allied Health building.

It is possible to access the system from other unix or linux systems acting as terminals. However, we have experienced occasional lockups in this mode. After logging on to the local system, issue the command *xhost +*, then *telnet instinct.v24.uthscsa.edu* and login as usual. The data transfer rate to the main campus appears to be adequate for the graphics intensive operations of **nmrDraw**. If accessing over a slow link, avoid the fid contour display in **nmrDraw** (don't press "d"), and avoid setting the "first" contour limit low when drawing the 2D plot.

It is also possible to access the system with **Xwin32** from a PC windows platform although we have had marginal to poor success with this method. This link is also prone to lock up and becomes more vulnerable to lock up in the context of a slower connection. If you are unable to minimize the **nmrDraw** window, then your interface is hung up, not just waiting for completion of a graphics command. It usually requires exiting Win32, and rebooting the windows operating system to reestablish the connection. Use version 5.1 or better. Use hostname *instinct.v24.uthscsa.edu*, and session type "XDMBC, query". Set the window mode to <single> and the size to about 1000 x 700. If your screen is too small, click the <maximize> button when the xwin32 session desktop is displayed. This will allow the full Unix desktop to become accessible with scroll bars. Specify <3 button simulation on> in the Configuration "input" window, and use left + right clicks to represent the middle mouse button. If connecting over a modem, it helps to establish the connection with the service provider with another program before starting the Xwin32 session. Over a slow connection, while in **nmrDraw**, scaling or offsetting the 1D spectrum by more than a few increments at a time tends to cause lock ups. Over slow connections, left-clicking in the graphics area fails to abort painting of a requested display, so be especially careful about starting unwanted displays. Avoid using sliders and other features with moving displays when numbers can be typed into boxes instead. To get remote printout, use the <file> <print hard copy> command to make a postscript file with the line that says "echo" by default blanked out. Copy the .ps file to the ~ directory where you can retrieve it by ftp. You can convert a .ps file to a .pdf file within a windows system using Adobe distiller (which requires the purchased copy on Adobe acrobat, not the free Adobe reader).

You will have the same username and password assigned that you used on the amx500. When you interrupt the screen saver with a mouse movement, a box will be displayed for your username and password. Upon entering these, you will get a desktop displaying the unix tool chest.

2. Select <desktop><open unix shell> from the unix tool chest to get a winterm window to work in.

This will start you in /u/people/<your username>, which is your home directory.< You can return to this directory by `cd` or `cd ~`. This directory will not be large enough for processing nmr data. It is used to keep environment files and files from other miscellaneous operations you may choose to do.

You will be assigned a larger quota of space on another disk drive to do the actual data processing. For example my data directory is /instinct2/hardies. Both your home directory and data directory have size limits set by the administrator. To see how much space is left in both of your directories, you have to know which computer your drives are attached too. In my case they are both attached to *instinct*. Type *telnet <computer hosting drive>* & logon with the same username and password. Then type <

The amx_500 disks are known to this system as /amx500_u, /amx500_nall, and /amx500_hinck. You can read from these units but not write to them from the NIS system.

Directory pseudonyms.

3. If you have not already done so, create pseudonyms for your NIS data directory, and amx data directory.

These instructions assume you have created pseudonyms for your NIS data directory, your amx_500 data directory, and the amx_500 pulse program directory. This is done by adding 3 lines to the end of your .cshrc file (a hidden file in your home directory; type `ls -a` to see hidden files). For example, I have used vi to add the following two lines to my .cshrc file:

```
setenv data /instinct2/hardies
setenv mmrdata /amx500_nall/data/hardies/nmr
```

After opening a shell with this .cshrc file in effect, these directories can now be abbreviated as *\$data*, and *\$nmrdata*

Similarly the command *setenv pp /amx500_u/exp/stan/nmr/lists/pp* has been added to a system .cshcc file so that the amx_500 pulse program directory is known to all NIS users as *\$pp*.

Overview of the nmrPipe system.

The overall steps are as follows:

1. Copy the dataset from the amx500
2. Run the program **bruker** to examine the data and fill in parameters needed for conversion to the nmrpipe format and processing. This program will write a conversion script that will actually perform the conversion.
3. Run the conversion script created by the **bruker** program to convert the data.
4. Run **nmrDraw** to test out a sequence of corrections to be made to the data. Accumulate your decisions in the form of a processing script.
5. Execute the processing script to Fourier transform the data including the corrections you have specified.
6. Use **nmrDraw** to examine the final 2D plot, and, if desired, to revise the corrections and try again.
7. Use **nmrDraw** to print or export the plot.

There is detailed documentation on the **nmrPipe** programs on line at:

<http://instinct.v24.uthscsa.edu/~hincklab>

Select <software packages> and <nmrPipe> Summaries of individual **nmrPipe** functions can be obtained by typing *nmrPipe -fn <function name> -help* at the unix command line.

4. Copy the dataset to your NIS data directory:

```
cd $data
```

```
ls $nmrdata (to id the dataset directory name)
```

```
cp -r $nmrdata/<dataset name> . (copies entire subdirectory structure of dataset; the "period" for a destination means to copy to the default directory, which is $data in the sequence above)
```

```
ls (confirm that your dataset is present; this is a subdirectory)
```

```
cd <dataset name>
```

ls (directories 1 2 3 etc. represent different experiment numbers you assigned. These are subdirectories. You must determine from your records which experiment number you intend to process. If you created a title file within topspin, it can be found in *pdata/1/title* and examined with *vi*.)

cd <experiment number>

ls -l (letter l, not number 1; shows directory including file sizes)

You should observe several files. The *acqu** files are parameter files. *ser* is the actual data. The pulse program is saved by the name *pulseprogram* and *pulseprogram.P*. These are somewhat processed by topspin before saving, so you may wish to compare to the original in the *\$pp* directory when seeking some information about the pulse program. However, remember that you may have subsequently edited the version in the *\$pp* directory in conjunction with a different experiment.

5. Write down the size of the *ser* file. It should be 2,867,200 if the data were collected in the usual way by the *hsqc_fb* pulse program (1024 complex points in direct dimension by 175 complex points in the indirect dimension).

6. Delete topspin processed files (if you have processed with topspin):

If you have processed the data on the AMX500, there will be large data files embedded further in the subdirectories that you should delete.

cd pdata

You will see one or more numbered directories. Each one is a separate time that you processed the data (and gave a different process number). For each one *cd <process number>*, *ls*, *rm 2**, *rm dsp**. This deletes the large processed files. Leave other files; they may contain documentation that will be useful for you later. Note: due to space limitations on the amx500, you should also delete the same processed data files from that system when they become obsolete. You must log onto the amx500 itself to do that. You can do that from the NIS system by *telnet amx500*, *logon*, go to your data directory and delete the same files. Current practice is to leave the rest of the dataset intact on the amx500 computer as a backup (specifically including the parameter files and the *ser* files).

Conversion of the *ser* file to the *nmrpipe* format.

The *ser* file contains nothing but a series of encoded numbers representing intensities of the RF signals from the orthogonal RF detectors at various time points during the

acquisition. There are several alternative formats that **nmrPipe** can deal with, so the first step is to tell **nmrPipe** the specifics of the organization of the ser file. Some of these specifics depend on exactly how you did the experiment, so information has to be extracted from the parameter files and the pulse program file and conveyed to **nmrPipe**. The program **bruker** knows how to read Bruker parameter files and pulse programs (the relevant ones of which have also been copied to your data directory). The program **bruker** will attempt to extract the relevant information and show it to you. It generally does not get it all correct, so you have to review and revise the information before attempting the conversion.

7. Set your directory to the relevant experiment-numbered subdirectory with the ser file you wish to process.

eg. `cd $data/hsqc_fb_l1.sch/1`

8. Run **bruker**

bruker

You will see two new windows. One contains a table of parameters for your review, and the other the conversion script that **bruker** will build from them. The spectrometer reading box should say `./ser`

Change the name of the converted output file if you like (The default name, `test.fid`, is assumed in the instructions that follow). The arrows on many of the boxes give drop down menus. You may select an option or directly type something in the box.

- **1. Set the dimension box to 2D to get the correct format for the table.**
- **2. Click <read parameters> to update the displayed parameters from the parameter files in this directory.**

The table will be updated with information extracted from the parameter files and the pulse program files saved in the directory with the ser file. The column labeled "x-axis" is also called the direct dimension or the proton dimension. The y-axis is also called the indirect dimension, or in this case the nitrogen dimension. The values

highlighted in yellow are particularly likely to require correction; however, you should check them all. Note: for a 3 dimensional experiment, the identities of the 2nd (y) and 3rd (z) columns are less intuitive, but must be correctly identified. The isotope whose delay times are changed in the inner loop of the pulse program is the y-axis. The one changed in the outer loop is the z-axis.

This is an example of a properly set up table for a $^1\text{H}/^{15}\text{N}$ HSQC experiment:

	<i>x-axis</i>	<i>y-axis</i>
<i>Total points</i>	2048	350
<i>Total valid complex points</i>	1024	175
<i>Mode</i>	complex	complex
<i>Spectral width</i>	6024.096	2000.00
<i>Observe frequency</i>	500.134	50.684
<i>Center position</i>	4.757	118.100
<i>Axis label</i>	HN	N

And the script:

```
#!/bin/csh
```

```
bruk2pipe -in ./ser -bad 0.0 -nosqap \
-xN          2048      -yN          350      \
```

-xT	1024	-yT	175	\
-xMODE	Complex	-yMODE	Complex	\
-xSW	6024.096	-ySW	2000.000	\
-xOBS	500.134	-yOBS	50.684	\
-xCAR	4.757	-yCAR	118.100	\
-xLAB	HN	-yLAB	N	\
-ndim	2	-aq2D	States	\

-out ./test.fid -verb -ov

The entries most likely to need changed are:

- 3. Particularly check that the y-axis observe frequency is 50.684 for ^{15}N .
- 4. The x and y center positions values are temperature dependent. The reported temperature from the temperature controller is not completely accurate. Optimally, you should have performed a temperature calibration experiment to have allowed setting your temperature more accurately, and a chemical shift referencing experiment to determine the appropriate center position values to enter in these boxes. See the document on [temperature calibration and chemical shift referencing](#). If you have not done this, then inquire as to a recent measurement of the bias of the temperature controller unit, and estimate values for center positions accordingly.
- 5. The axis labels can be whatever you want printed on the axes.

Explanation of the parameters.

This table corresponds to your experiment as follows:

x-axis. The protons in the protein are excited by a pulse sequence that involves an interaction with adjacent nitrogens. The two detectors then take a series of paired readings at successive time points to define the RF signal that is emitted by those protons. This is the data that would define one complete 1-dimensional nitrogen-edited proton spectrum.

The parameters describing one such series are listed in the x-axis column.

y-axis. There are a series of nitrogen-edited proton spectra recorded in the ser file, each taken with a different delay time in the pulse sequence that affects how the nitrogens influence the intensity of the bound protons. The y axis column has parameters related to retrieving the chemical shifts of the bound nitrogens by processing this series.

MODE: In the ser file, the signal from the two detectors are recorded as ordered pairs in the format of a complex number. Therefore the mode row should be set to "complex" in the x column. The series of proton fids is taken in pairs so that the nitrogen dimension will also be composed of complex numbers. So the mode in the y column should also be "complex".

Total points and Valid complex points: The total number of complex numbers in each complete proton spectrum is listed as number of valid points. This should be twice the number of total points tabulated, since for complex numbers there are two points per complex number. The subroutine called by the pulse program to collect the data retrieves 1024 complex points per call.. The proton scans are stacked up as pairs used to construct a complex number for each point in the indirect dimension. The number of such pairs is controlled by parameter L3=175. You can check on the value of L3 by looking at the messages given in the winterm window as **bruker** runs. You can also use a specialized version of grep to retrieve it from the parameter files: Open a new shell, set the directory to this data directory, type `uxgrep l` (lower

case L). `uxgrep <string>` can be used to search the parameter files for other parameters.[Note: when you change the acquisition time by altering parameter NS, you do not change these parameters. You just change the number of replicate readings that get averaged into a single recorded reading. NS doesn't appear in the pulse program. It is used by the subroutine called by the pulse program.]You can check that the number of points inferred by **bruker** are correct by noting that total points in x * total points in y * 4 should equal the size of the ser file in bytes (which you wrote down earlier). In this case, $2048 * 350 * 4 = 2,867,200$.**NB!** The above parameters must be correctly set in order to process the data. The parameters below influence the accuracy with which the axes are labeled on the 2D plot that results.

Observe frequency. – the carrier frequency for each of the nuclei observed. The drop-down menu lists appropriate values for each of the following nuclei (H = 500.134, N=50.684, C = 125.764). More accurate values should be used for calculating center position values below. To get more accurate values look at the pulse program to see which channel was used as the carrier for each kind of nucleus and use `uxgrep` to see the corresponding SFOx parameter. For example, in `hsqc_fb`, nitrogen was stimulated on channel f3, and SFO3 reveals the carrier frequency to have been 50.683840. For ^{13}C , there may be several frequencies used in the pulse program, specified by a parameter F2LIST. In this case, only one of the frequencies is the relevant observe frequency for ^{13}C , and a comment in the pulse program should reveal this frequency.

Spectral width – the maximum frequency that can be correctly measured given the time intervals between the digital RF

measurements. The pulse program specifies the sweep width for the proton axis as parameter SW_h. The sweep width for the indirect dimension is set up in the pulse program as $1/(2 \cdot IN0)$. For other pulse programs, the correct formulation for spectral width should similarly be revealed by a comment in the pulse program. This formula can be picked out of the drop-down menu. Note that signals outside the range appear at false positions within the range (said to be “folded”). These are identifiable by having an inconsistent phase with the other peaks in the final 2D plot.

Center position – The chemical shift of the point in the middle of the spectrum ($N/2 + 1$). As of this writing, it is 4.7396 for ^1H and 118.05 for ^{15}N at 300 K. The temperature controller currently reads 1.6 degrees higher than the true temperature. See [temperature calibration and chemical referencing](#).

Axis label – is whatever you want as a label for this axis on the graph. “NH” for example, means amide hydrogens.

Executing the conversion.

8. Execute the script by <save script>, exit the bruker program, and run the script by typing *fid.com*:

Alternatively, you may click on <save script> and <execute script> to execute the script from within the **bruker** program. Check that the output file (test.fid, or otherwise if you renamed it in **bruker**) has been created .

Processing the data

Overview

The processing is also done by a script executed by the program **nmrPipe**, and there is an interactive graphical program named **nmrDraw** to help you set up the processing script and to view your data.

An example of a script that I have used on $^1\text{H}/^{15}\text{N}$ HSQC data is as follows:

```
#!/bin/csh

#
# Basic 2D Phase-Sensitive Processing
# Cosine-Bells are used in both dimensions.
# Use of "ZF -auto" doubles size, then rounds to power of 2.
# Use of "FT -auto" chooses correct Transform mode.
# Imaginaries are deleted with "-di" in each dimension.
# Phase corrections should be inserted by hand.

nmrPipe -in test.fid \
| nmrPipe -fn SOL \
| nmrPipe -fn SP -off 0.30 -end 1.00 -pow 1 -c 0.5 \
| nmrPipe -fn ZF -auto \
| nmrPipe -fn FT -auto \
| nmrPipe -fn PS -p0 167.0 -p1 0.00 -di -verb \
| nmrPipe -fn EXT -left -sw \
| nmrPipe -fn POLY -ord 1 -nl 20 40 60 80 100 120 140 180 750 800 850 900 950 \
| nmrPipe -fn TP \
| nmrPipe -fn SP -off 0.35 -end 1.00 -pow 1 -c 1.0 \
| nmrPipe -fn ZF -auto \
| nmrPipe -fn FT -auto \
| nmrPipe -fn PS -p0 -90.00 -p1 180.00 -di \
  -ov -out test.ft2
```

The general form of this script is that the first line reads in the raw data file (*-in test.fid*), each subsequent line performs a processing function (like *-fn SOL*) with a variety of parameters set by switches (like *-auto*), and the last line includes a specification to write

an output file (*-out test.ft2*). You will have to customize several parts of the script for your data.

The \ at the end of each line is a continuation mark. **Do not put any characters (including blanks) past the \.**

The | at the beginning of each line is a unix "pipe" operator. The output of one function is passed as input to the next without writing a file. You must put a *-out <filename>* specification to get any output. You may also put this specification elsewhere in the option list to save intermediate results. The *-ov* specification means to overwrite preexisting files of the same name. If you put a *-outs* specification at an intermediate position, you should comment out the remainder of the script (with a leading #), or follow the statement with the *-out* with a *nmrPipe -in <intermediate filename>* statement to restart the pipe.

Meanings of the functions in the example script are listed below, as well as whether they usually need customized..

- **SOL** – corrects fids for water distortions. Not usually customized.
- **SP** – window function applied to direct dimension to govern signal-to-noise ratio and peak resolution. Will probably need to be customized after seeing fully processed data.
- **ZF** – zero fill; extends fids with zeros; improves peak resolution; Not usually customized.
- **FT** – Fourier transform 1st dimension. Not usually customized.
- **PS** – set phase for 1st dimension. You will have to process to this point, determine phases manually, and edit them into the script. After the data is fully processed, you will probably further adjust the phase values.
- **EXT** – cut off the right part of the 1st dimension; You can use more specific left and right limits to cut out a noisy water region. eg. *-fn EXT -sw -x1 10.5PPM -xn 6.0PPM*. The left limit is x "one" not x "el". Don't leave out the *-sw*; it updates a header stored with the data that is essential for further processing.
- **POLY** – smooth the frequency domain baseline. Process to this point and extensively customize. May need readjusted after seeing fully transformed data.
- **TP** – transpose the data so that functions now apply to y-axis instead of x-axis. Not usually customized.
- **SP** – window function applied to indirect dimension to govern signal-to-noise ratio and peak resolution. Will probably need to be customized after seeing fully processed data.

- **ZF** – zero fill y-dimension. Not usually customized.
- **FT** – Fourier transform y-dimension. Not usually customized.
- **PS** – Set phase for y-dimension. These values are fixed by the pulse program, and should be stated as a comment in any 2D pulse program.

The script will be kept in the same directory with the input data by the name `nmrproc.com`.

You could copy a script like the one above to this directory and name it `nmrproc.com` if you wanted to use it as a template. Otherwise, you can set up the script from a template provided by **nmrDraw**.

One explores the steps necessary to customize the script by running the graphical program **nmrDraw**. The program **nmrDraw** allows you to execute a whole script on the full data, or only certain steps in the script on selected fids. There are two ways to execute the functions from within **nmrDraw**. 1) You can edit a script to comment out steps you don't want to do (by adding a `#` to the beginning of the line), and add a `-ov -out <filename>` to steps for which you want to recall and review partially processed data. Or 2) you can directly load the unprocessed file, a partially processed file, or the fully processed file, and then pick a particular fid or frequency domain slice and transiently apply functions to it. For first-pass adjustments, one partially processes to the step prior to which customization is needed, and then uses transient applications of the next function to settle on the desired parameters by trial and error. One then edits these parameters into the script and uncomments down to the next stopping point (remembering to move the `-ov -out <filename>` specification to the new point. Once one gets to a fully processed spectrum, additional adjustments may be made by editing the script and fully processing the data to see the end result of the modification.

The program is very general, and different users will develop their own strategies for working through the data. However, a step-by-step example is given below to help new users get started.

9. Run **nmrDraw**.

*With the directory set to contain your raw data as `test.fid`,
type `nmrDraw`*

***nmrDraw** menus are expanded by right mouse clicks, but functions in the menus are activated by left mouse clicks.*

10. Load a template script by <file><Macro edit><process 2D><Basic 2D>.

An editor window will come up with a basic script with most of the processing steps you can expect to perform already filled in. If you want to use a script that you've used before, copy it to the directory and rename it nmrproc.com prior to running nmrDraw. It will then appear in the macro edit window after <file><Macro edit>.

11. Explore the solvent distortion correction.

- **1. Load your raw data by <file><select file> and pick test.fid out of the menu. Click <done>.**
- **2. Press "d" on the keyboard and then "h".**
- **3. Left click near the bottom of the drawing area and drag the mouse to the bottom of the drawing area. The box in the upper left display area should say y=1.**

"d" (or <draw><contour>) shows a view from the top of the rows of fid oscillations. "h" (or <mouse><1 D horizontal>) shows one of the fids; the row on the bottom (y=1) is the first serial fid with no time evolution from nitrogen. Until the 2nd dimension is Fourier transformed, you should always look at this slice when judging the operation of a function to avoid being confused by the effects of nitrogen evolution.

On a slow connect, skip the 2D display by loading with <read> <done>. Then set the y coordinate box to 1 followed by a < carriage return> and choose the horizontal 1D display with "h".

Available mouse operations are indicated on the upper window bar. When the mouse is inside the drawing area, the left, middle, right buttons select an fid, horizontal pan, and horizontal zoom, respectively. When the mouse is over the purple borders the buttons set the phase pivot, vertical scale, and vertical offset, respectively, of the chosen fid.

Examine this fid (y=1). It should be a complex of interfering high frequency sine waves which decay in intensity as they proceed from left to right. The axis that the waves oscillate around should be a straight line. Solvent distortion will cause the axis to slowly undulate itself as it progresses from left to right. This is an artifact due to imprecision in water suppression, and the first issue requiring a processing step. If you do not remove this effect, the right side of your 2D plot will be overcome with a residual water signal centered on the water (carrier) frequency. The recommended correction function is named SOL. It takes an average of 30 points around each point to estimate the offset of the undulating axis from zero. It then subtracts that offset from each point. To see if SOL improves the fid try it out as follows.

- **4. Right click <proc> <function> <SOL solvent correction>. SOL should appear as the proposed function**

to transiently execute. Then left click <execute>. Make the pop-up window go away with <done>.

Your fid should become straightened out. If not, then 1) make a resolution to set up your solvent suppression better the next time you do an acquisition. 2) You can remove the effect of the last processing by pressing "h" on the keyboard, and then repeat step 4 above using some of the options available for SOL or the alternative function POLY, solvent correction. Information about how to use these options can be found by consulting the nmrPipe web page cited at the top of this document. A listing of options can be found for any nmrPipe function by typing (in a separate shell window) `nmrPipe -fn <function name> -help`. If the solvent can not be completely suppressed, then process through to the Fourier transformed data and see if the noise is confined to a region away from your peaks. If so, you may use something like `-fn EXT -x1 10.5PPM -xn 6.0PPM -sw` to cut the water region out of the dataset. You could go on to explore how the SP, ZF and FT functions affect the look of the 1st direct slice by transiently applying them in turn. However, these don't need adjustment at this time.

12. Edit the script to process through the first Fourier transform.

Your script should look something like this:

```
nmrPipe -in test.fid |  
| nmrPipe -fn SOL |  
| nmrPipe -fn SP -off 0.50 -end 1.00 -pow 1 -c 0.5 |  
| nmrPipe -fn ZF -auto |
```

```
| nmrPipe -fn FT -auto -ov -out test.ft1 |  
#| nmrPipe .... rest of lines commented out
```

Note that you will have added the line with -fn SOL (or whatever alternative you settled upon).

You also should add the -c 0.5 switch to the SP function. This prevents a mathematical artifact in the Fourier transform that will shift your baseline for the transformed spectra up from zero. You will probably later modify the -off parameter and maybe the -powparameter, but leave them alone for now.

Also, don't forget to add the -ov -out <filename> specification after the 1st Fourier transform.

13. <save> and <execute> the script.

<save> puts the edited script in nmrproc.com. <execute> executes the script in nmrproc.com.

If you do not <save> before <execute> you will inadvertently execute the previous version of the script (if there was one).

14. Load the partially processed file by <file><select><test.ft1><done>. Use "d" and "h" and pick the 1st slice (y=1) as before. "c" followed by "h" will remove the contour display.

15. Phase the spectrum.

- 1. Put the phasing button "on" in the upper right of the control panel.
- 2. Use the large slider at the left of the control panel labeled P0 to do course phase adjustment, then use the smaller slider next to that for fine adjustment.

- **3. Note the phase value in the p0 box. Edit it into the script replacing the 0.00 next to the -p0 parameter in the PS function.**

Phases are modulo 360. -200 and +160 are the same thing.

It will be possible to adjust phase more precisely after baseline correction is made and you separate the peaks more cleanly in the 2D display.

16. Explore the baseline correction step.

- ***1. Edit the script to uncomment the PS step and the EXT step. Move the output statement to after the EXT step.***

```
nmrPipe -in test.fid \  
  
| nmrPipe -fn SOL \  
  
| nmrPipe -fn SP -off 0.50 -end 1.00 -pow 1 -c 0.5 \  
  
| nmrPipe -fn ZF -auto \  
  
| nmrPipe -fn FT -auto \  
  
| nmrPipe -fn PS -p0 167.0 -p1 0.00 -di -verb \  
  
| nmrPipe -fn EXT -left -sw -ov -out test.ft1 \  
  
#| nmrPipe ... rest commented out...
```

Notice that we will process from the beginning to the new stopping point. Instead we could input the first intermediate file and just do the new processing steps. But in the way illustrated above, the script always corresponds to exactly what processing was done to make the latest intermediate file.

The `-verb` switch can be put on any function. It displays a popup window showing how the process is going during execution. If the process fails, unfortunately, the pop up tends to disappear before you can read the messages. If a process fails, minimize **nmrDraw** to an icon. There will be error messages in the winterm window.

- 2. <Save> and <execute> the script.
 - 3. Load the test.ft1 file and set up to observe the first horizontal slice (y=1) as before.
 - 4. See if *POLY -auto*, or *POLY -auto -ord 1*, or *POLY -auto -ord 2* or *POLY -auto -ord 3* will flatten out drift in the baseline. Note: do not try to remove high frequency noise by this method.
- • • 1. In each case select <proc><functions><baseline>, fill in the specific switches, and <execute>
 - 2. To remove one corrections and try another press "h".

Poly does a polynomial fit of points judged to be on the baseline, and then subtracts this function from the spectrum to try to flatten and zero the baseline. The `-auto` switch allows the program to automatically choose the points that it considers to be on the baseline for the fitting. `-ord` specifies the order of the polynomial, ie. `-ord 1` fits a straight line. The default is `-ord 4`. Because there is a big region in the middle of the spectrum that does not come down to baseline, lower order polynomials may be a better choice.

- • • 5. If *POLY -auto* seems unsatisfactory, try *POLY -ord 1 -nl <point list>*, or then *-ord 2*.

The `-auto` function may be unsatisfactory if your baseline is particularly noisy, or if there are short interpeak regions that you want to force onto the baseline. Then you can specify a list of points to use in the fitting after a `-nl` switch (that's letter l). The positions have to be in points, not ppm.

- • • 1. Remove all other corrections with "h".
- 2. If the bottom axis is labeled in ppm, then go to <draw> <2D settings> and pick <points> for the x-axis units.
- 3. Select <draw><toggle> or type "/" at the keyboard.

This last operation adds a second horizontal scale to the drawing area. Some operations (EXT, ZF, zoom, and pan) change the horizontal scaling but do not update the scale in the purple panel once the file is loaded. The new scale placed in the drawing area is dynamically updated to keep it correct. You will read the coordinates for the point list from this new axis.

- • **4. Use <proc><functions><baseline> to try out a variety of point lists until you are satisfied with the baseline correction.**

17. Process to a fully processed 2-D plot

- **1. Add your baseline correction to the script.**
- **2. Add a -c 1.0 switch to the indirect dimension SP function and make -off = 0.5**
- **3. Fill in -90.0 for -p0 in the indirect PS function and -p1 180.0**
- **4. <save> and <execute> the script.**

The script should look something like this:

```
nmrPipe -in test.fid \  
  
| nmrPipe -fn SOL \  
  
| nmrPipe -fn SP -off 0.5 -end 1.00 -pow 1 -c 0.5 \  
  
| nmrPipe -fn ZF -auto \  
  
| nmrPipe -fn FT -auto \  
  
| nmrPipe -fn PS -p0 167.0 -p1 0.00 -di -verb \  
  
| nmrPipe -fn EXT -left -sw \  
  
| nmrPipe -fn POLY -ord 1 -nl 20 40 60 80 100 120 140 180 750 800 850 900 950 \  

```

```
| nmrPipe -fn TP \
```

```
| nmrPipe -fn SP -off 0.5 -end 1.00 -pow 1 -c 1.0 \
```

```
| nmrPipe -fn ZF -auto \
```

```
| nmrPipe -fn FT -auto \
```

```
| nmrPipe -fn PS -p0 -90.00 -p1 180.00 -di \
```

```
-ov -out test.ft2
```

18. Do a refined zero and first order phase correction.

- **Load the processed file.**
- **"d"**
- **Use the + and – buttons to adjust the level of the first displayed countour to see all the peaks (as spots) but not much noise.** You have to redraw the screen ("d") to visualize each + or – adjustment. A more quantitative approach is to set the first contour level box relative to the level of noise. <peak><estimate noise> gives a pop-up window with an estimate of rms noise. By default, the first countour is set to 6 times the noise level. This tends to leave out smaller peaks. 4 times the noise is good for looking for faint peaks. 3 times the noise is good for displaying patterns in the noise, including phase errors.

On a slow connect, read the 2D file by <read><done>, and use <peak> <estimate noise> to set the first contour box as desired before issuing the draw command. This will avoid waiting for an unnecessary display to be drawn.

- **Click the "phasing" button on.**
- **"h"**
- **Select a horizontal slice running through a well separated peak far to the left of the plot.**

- Put the phase pivot on the center of this peak. (move arrow in lower purple border with left mouse button).
- Adjust the P0 sliders to optimally phase this peak. (make it symmetrical with a flat baseline on either side.)
- Select a horizontal slice running through a well separated peak far to the right of the plot.
- Move the P1 sliders to optimally phase this peak. Note that only the activated 1D display (in this case the direct dimension) is updated from the slider. The contour display remains unaltered, as does the 1D display of the other dimension. These will not be updated until the file is reprocessed as below:
- Note the values in the P0 and P1 boxes and add these numbers to the phase corrects in the first dimension PS function in the script.
- <save> and <execute> the script. Reload the fully processed file.

First order phase correction (P1) means that a different phase correction is applied to each peak as a linear function of its frequency. Ideally there would be no need for a first order phase correction. It appears because a delay parameter "d7" in the pulse program hasn't been fine tuned yet. Right now I get about -45 degrees. It may be considerably reduced in future releases of the hsqc pulse program. The first order phase correction may interact with the -c switch in the SP window function. Ideally it would be small and -c 0.5 would then be correct for the first dimension SP function. High first order phase correction (as in the 2nd dimension) should use -c 1.0 (which is the default). The slightly high P1 correction in the first dimension may cause a baseline shift, which may be complicating the baseline correction. For example, it may be why POLY -auto tends to fail.

19. Evaluate the signal to noise and the resolution.

A high value of -off in the SP function more severely emphasizes the beginning of the fid in order to decrease the noise. A lower value (say 0.35) gives more emphasis to the end of the fid, giving greater peak resolution at the expense of more noise. If your peaks are not very intense, or if you are looking for faint side peaks that may be conformational

variants, then you will want to stay with good signal to noise. If your signal is strong and without faint spots, and you would like to improve peak separation, you should try a lower *-off* setting for the SP function in one or both dimensions. You may observe that by decreasing the value too severely (particularly in the indirect dimension) that you cause the fid to fall off as too much of a step function rather than smoothly. This causes wiggles on the sides of each peak in that dimension. On the 2D plot, these would appear as red fringes surrounding the more intense blue peaks or as one or more satellite peaks. If trying to optimize resolution, you might try

-pow 2 in the SP function to overcome this effect. There are several other window functions you could try listed on the **nmrPipe** web page cited at the top of this document.

The truncation problem mainly affects the ^{15}N dimension. Peaks differ in their vulnerability to this problem based on their individual relaxation times. Peaks that have long relaxation times also tend to be intense, thus increasing the visibility of the artifact. You may choose to tolerate this artifact on certain well separated peaks, in exchange for better resolution in other areas of the plot.

- **1. For each trial, edit the script, <save> and <execute>, reload the final file, and examine the critical regions with the "h" and "v" (vertical slice) functions.**

20. Reevaluate the baseline correction.

In the 2D plot, look at the baselines of several of the transformed slices. If there is a consistent curvature to them, then you may return to the POLY baseline correction and try harder to remove that trend from the data. It may be helpful to return SP -off to 0.5 to suppress noise while reworking the baseline.

21. Plot the final processed data.

To create a plot, select <print hard copy> from the <file> menu. The the window which appears, supply an appropriate name for a postscript

file to contain the plot. To get hard copy at this time, replace the word "echo" with `lp -dlaser1` (that's "el" pee – dlaser "one"). This directs the plot to the printer in the back of the structure center computer lab. As currently configured, neither the title specified in the `<print hard copy>` window, nor labels assigned to peaks through `<peak detection>` will display on the printed copy. When creating a .PS file from a remote connection, blank out the field that contains the word "echo". There is a program named `showps` that can be used to print postscript files at a later time.

22. Record your final noise and signal to noise ratio.

- **1. Use `<peak><estimate noise>` to get an estimate of you noise level.**
 - Noise should scale linearly with receiver gain (parameter RG, usually 256 for `hsqc_fb`), and as the square root of parameter NS. Typical values after using `SP -off 0.5` are:
 - RG=256, NS=16, rms noise = 9000.
 - RG=512, NS=16, rms noise = 18000
 - RG=1024, NS=32, rms noise = 42000
 - Much greater noise levels would suggest some problem in the execution of the experiment or in the processing.
- **2. Measure representative peak heights to record signal strength.**
 - Select `<peak> <peak detect> <detect>` to measure peaks.
 - Select `<variables>` and select `<peak height>` from the list.
 - Select `<draw>` to draw the labels on the display.
 - Note the heights of representative well isolated peaks.
 - To unclutter the display, use `<Mouse><2D zoom>` and left click to move the corners of the rectangle over a small area you wish to examine. Right click to zoom. Then `<peaks>` and `<draw>` to put back the labels.
 - Signal/noise for my experiment was 50-100 at NS=16.
 - The more critical sensitivity parameter is T2 relaxation time, which is measured in a separate experiment. The HSQC is usually qualitatively evaluated for approximately the right number of peaks, mostly resolved, and not clustered in the 8 ppm region (which an unfolded random coil would do).

