Processing NMR Data With OS X/Linux Freeware

Volume 1:
A User’s Guide to NMRPipe

by Josh Kurutz, Ph.D.

Technical Director of the Biochemistry & Molecular Biology NMR Facility

University of Chicago

Last revised: February 16, 2006

Disclaimer: This guide is written in good faith by a user for other users, but the author cannot take responsibility for problems associated with using it. Be advised that the contents may not be error-free or comprehensive, though the author has attempted to make them as useful as practical.

Table of Contents.

Table of Figures.

NMRPipe Processing Guide

1) 1D Spectrum

A. Overview

NMRPipe is not your best choice of program for processing a 1D NMR spectrum. For most purposes, you are better off processing it at the console using VNMR. VNMR also features many tools for analyzing 1D spectra, such as T1 and T2 analysis, and versatile ways of displaying arrayed spectra. NMRPipe is more versatile than VNMR for multi-dimensional spectra.

For the occasions when you do not have ready access to VNMR, follow the first steps in the procedure for processing a 2D spectrum with NMRPipe.

I have just become aware of a free OS X tool for processing 1D data. It's called "transfourier" and was written by Stephen Jones of the College of Charleston, South Carolina. You can download it by going to: http://chem.cofc.edu/ssj/MacOSX/transfourier1-1.dmg.

2) 2D Spectrum

A. Overview

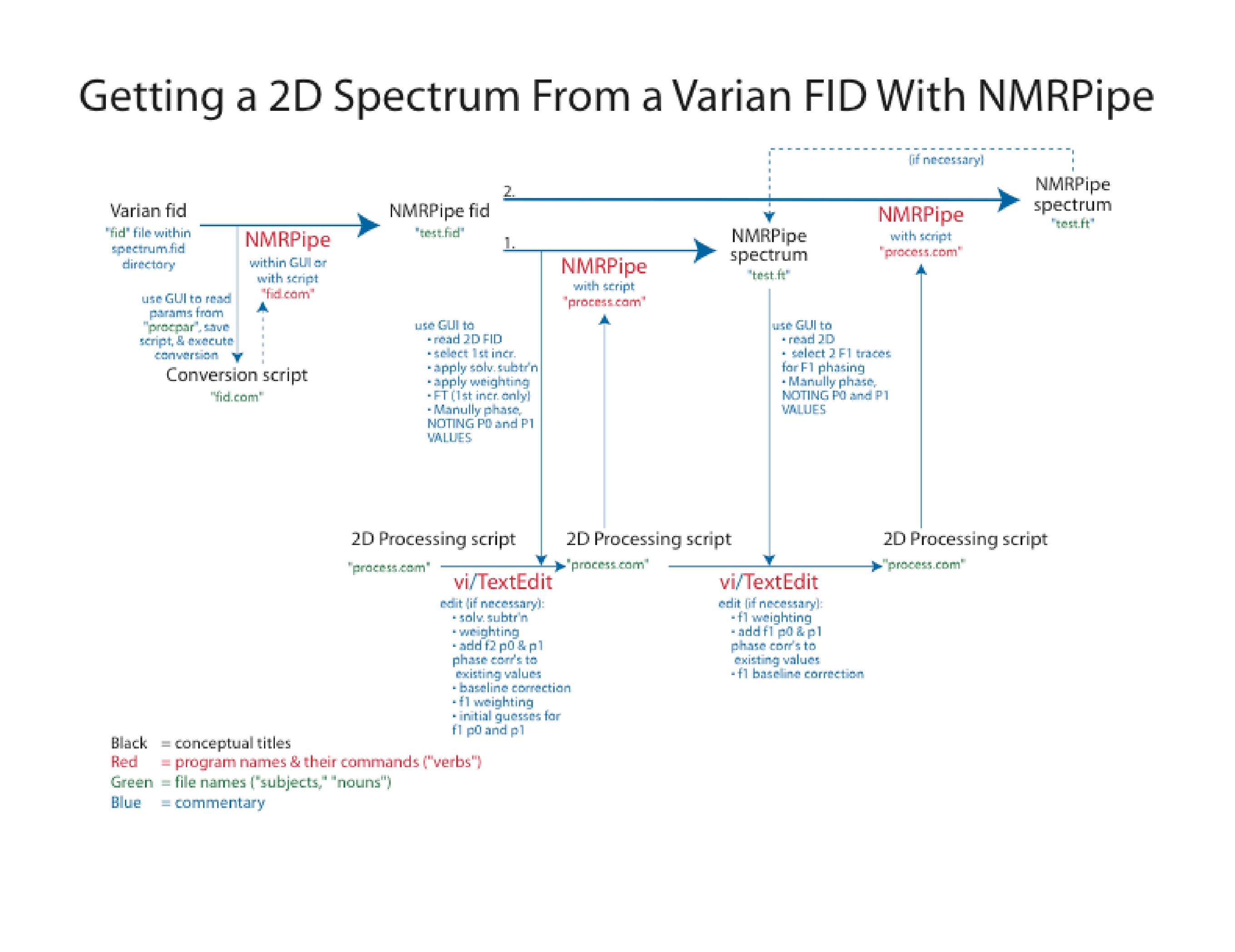
These instructions for processing a 2D NMR spectrum are the basis for processing all other spectra. 1D spectra can be handled with a subset of these instructions. 3+D spectra require initial processing of the first 2D plane, after which addition step, covered later in this guide, are performed.

Figure 1 provides a schematic overview of the procedure for obtaining a high-quality spectrum from raw data provided by the spectrometer.

It is important to realize that processing this data requires you to frequently exit and enter NMRPipe to work with text editors and invoke commands on command lines. This is because much of the data does not actually get processed using the graphical user interface (GUI) that appears when you type "nmrDraw." The function of the GUI is to allow you to view spectra that are already established as independent files and to perform minor manipulations on them.

Most of the computation is conducted by scripts, which are files in your directory; these usually end in ".com" or ".s." Typing the name of a script on a terminal command line gets it to run. A line within the script will tell NMRPipe which file to act upon, and another line will tell it the name a new file to write. Thus, each script will create a new file without destroying the original. In a sense, these files and scripts are like different parts of speech in a sentence (Figure 1). The scripts are the verbs, and act on original files, which are the subjects of the sentences (nouns), to produce new files, which are the objects of the sentences.

You, the user, will edit these scripts to specify file names and adjust processing parameters. (Don't worry – you'll be starting from existing template scripts and modifying them, not creating them from scratch.) You can use various test editors, but you'll be frustrated if you use Microsoft Word, which inserts 6M characters at the end of every line, even when saving as "text only." On the Mac, I recommend TextEdit or, if you're used to it already, vi editor from the command line.



Getting a 2D Spectrum From a Varian FID With NMRPipe

Varian fid
"fid" file within spectrum.fid directory

NMRPipe
within GUI or with script "fid.com"

use GUI to read params from "procpar", save script, & execute conversion

Conversion script
"fid.com"

NMRPipe fid
"test.fid"

2.

1.

NMRPipe
with script "process.com"

use GUI to
- read 2D FID
- select 1st incr.
- apply solv. subtr'n
- apply weighting
- FT (1st incr. only)
- Manully phase, NOTING P0 and P1 VALUES

NMRPipe spectrum
"test.ft"

(if necessary)

NMRPipe
with script "process.com"

NMRPipe spectrum
"test.ft"

use GUI to
- read 2D
- select 2 F1 traces for F1 phasing
- Manully phase, NOTING P0 and P1 VALUES

2D Processing script
"process.com"

vi/TextEdit
edit (if necessary):
- solv. subtr'n
- weighting
- add f2 p0 & p1 phase corr's to existing values
- baseline correction
- f1 weighting
- initial guesses for f1 p0 and p1

2D Processing script
"process.com"

vi/TextEdit
edit (if necessary):
- f1 weighting
- add f1 p0 & p1 phase corr's to existing values
- f1 baseline correction

2D Processing script
"process.com"

Black = conceptual titles
Red = program names & their commands ("verbs")
Green = file names ("subjects," "nouns")
Blue = commentary

Figure 1. Flowchart for processing a 2D NMR spectrum with NMRPipe

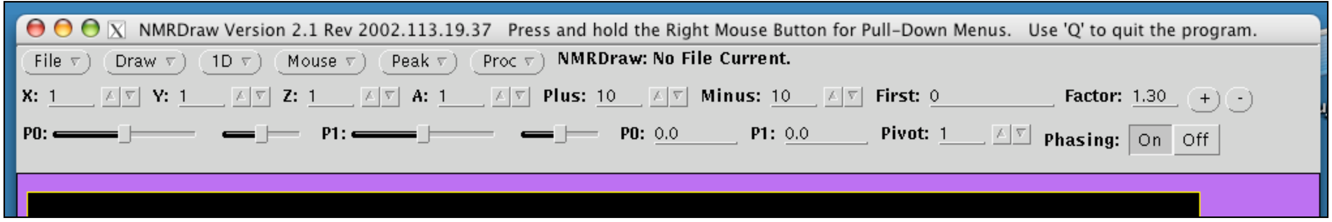
B. Start the program

a) In a terminal window (in X11 if you're on a Mac), change to the directory of your NMR spectrum. If you're processing several spectra, it will be handy to go to the directory where you can view all the .fid directories. If you're looking at just one, switch into the Varian .fid directory. You should see the files fid, procpar, log, text, and maybe some other, optional, files.

b) Type "nmrDraw."

C. Orient yourself

a) Here's the primary toolbar (as it appears on my Mac OS 10.3 desktop).



NMRDraw Version 2.1 Rev 2002.113.19.37 Press and hold the Right Mouse Button for Pull-Down Menus. Use 'Q' to quit the program.

NMRDraw: No File Current.

X: 1 Y: 1 Z: 1 A: 1 Plus: 10 Minus: 10 First: 0 Factor: 1.30

P0: 0.0 P1: 0.0 Pivot: 1

Figure 2. NMRPipe main toolbar

b) Here's what you see when you pull down these menus (accomplished by right-clicking on the menu buttons):

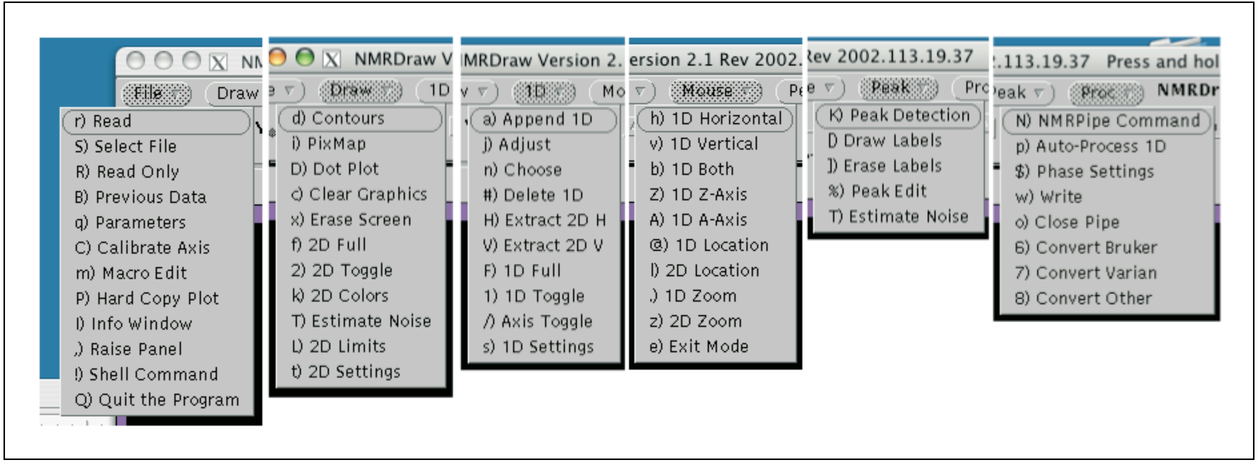
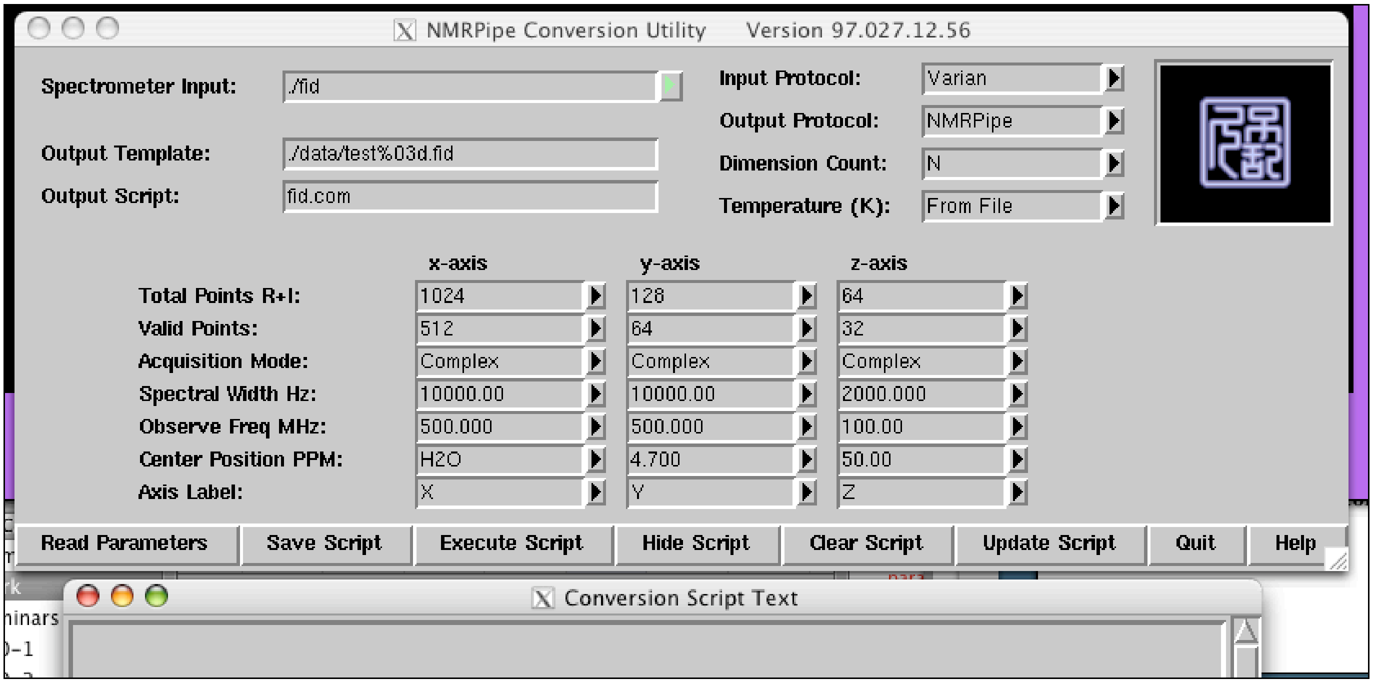


Figure 3. NMRPipe main toolbar pulldown menus

c) Familiarize yourself with the effects of clicking different mouse buttons. Right-clicking opens up menus from menu buttons. Left-clicking selects items from the menu. Left-clicking on a button that activates a menu when right-clicked will select the choice at the top of the menu.

D. Convert Varian's fid to an NMRPipe fid

a) Right-click on the "proc" menu button to pull down the menu.
b) Click "7) Convert Varian." This should call up two windows, one which houses dialog entries for details of the conversion, and one that lets you view the script you are creating. These are shown below:



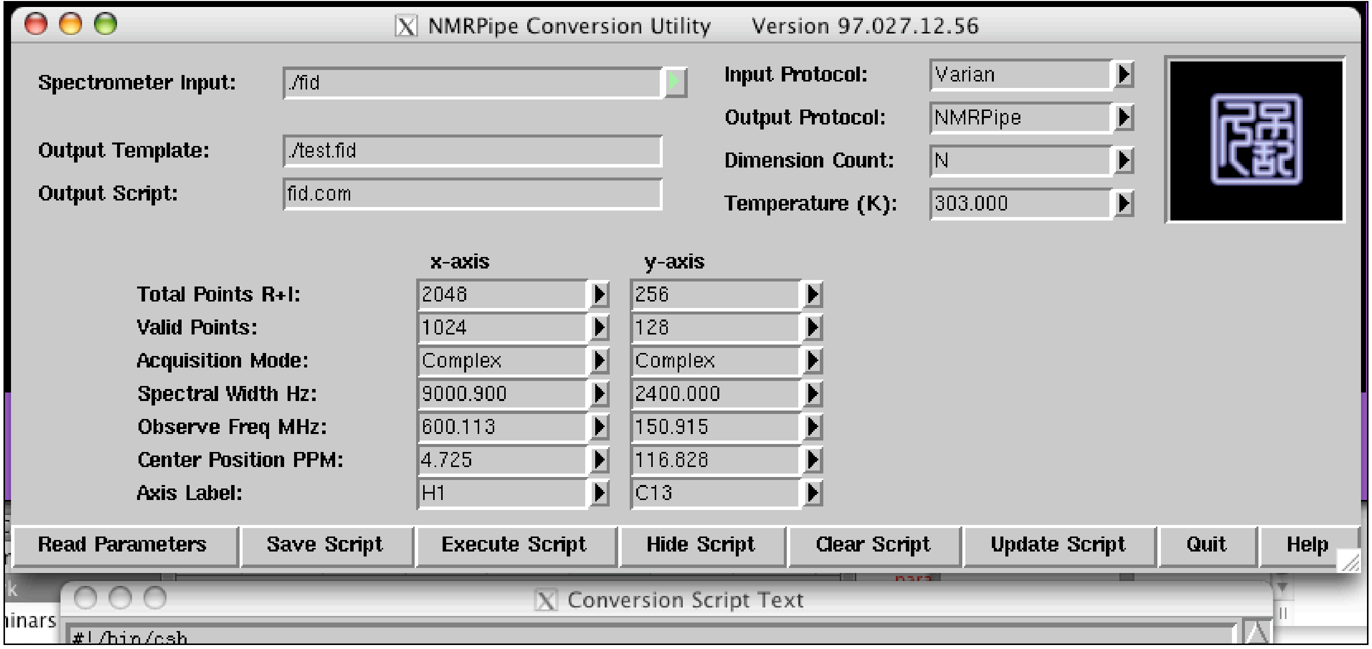
Spectrometer Input: ./fid
Output Template: ./data/test%03d.fid
Output Script: fid.com
Input Protocol: Varian
Output Protocol: NMRPipe
Dimension Count: N
Temperature (K): From File

	x-axis	y-axis	z-axis
Total Points R+I:	1024	128	64
Valid Points:	512	64	32
Acquisition Mode:	Complex	Complex	Complex
Spectral Width Hz:	10000.00	10000.00	2000.000
Observe Freq MHz:	500.000	500.000	100.00
Center Position PPM:	H2O	4.700	50.00
Axis Label:	X	Y	Z

Figure 4. GUI for converting Varian data

Because you started NMRPipe by typing 'nmrDraw" while you are already in a Varian .fid directory, the GUI has identified the input file as "./fid," "./" indicating your current directory. You are going to use this GUI to write a script called "fid.com" that will convert the file "./fid" to an output file, whose name will not be that which currently appears in the "Output Template" box. Indeed, the next step will change the view significantly.

c) Left-click on the "Read Parameters" button. The program looks up the experimental parameters in the Varian file procpar, determines the dimensionality of the experiment, and enters in several of the appropriate numbers required for conversion into their corresponding boxes in the window.



Spectrometer Input: ./fid

Output Template: ./test.fid

Output Script: fid.com

Input Protocol: Varian

Output Protocol: NMRPipe

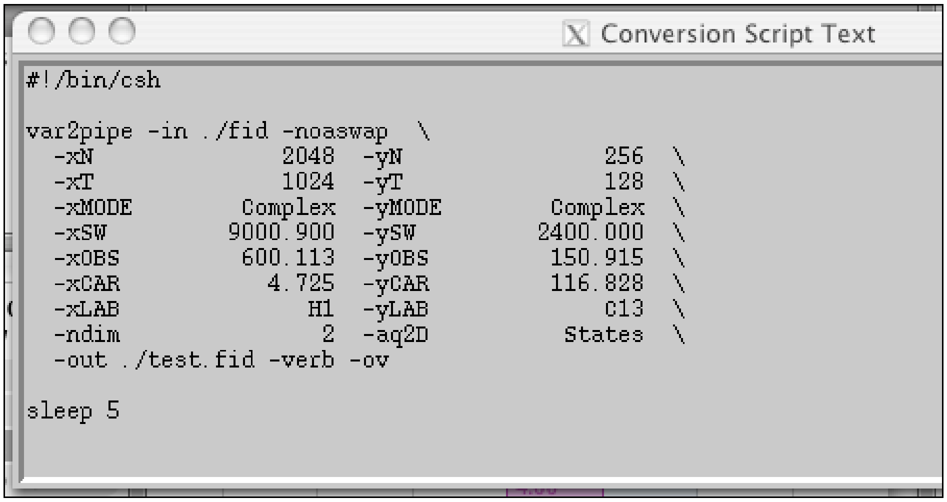
Dimension Count: N

Temperature (K): 303.000

	x-axis	y-axis
Total Points R+I:	2048	256
Valid Points:	1024	128
Acquisition Mode:	Complex	Complex
Spectral Width Hz:	9000.900	2400.000
Observe Freq MHz:	600.113	150.915
Center Position PPM:	4.725	116.828
Axis Label:	H1	C13

Figure 5. Varian conversion window after reading parameters

Reading the parameters also makes a preliminary script appear in the conversion Script Text window:

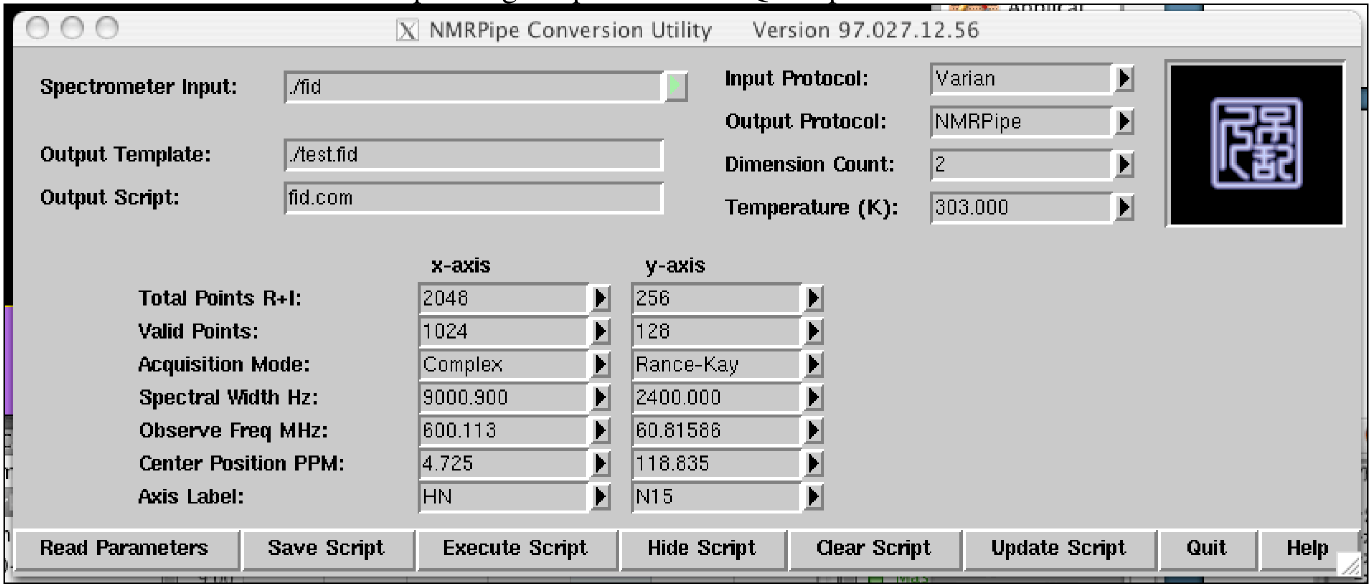


```
#!/bin/csh

var2pipe -in ./fid -noaswap  \
  -xN              2048  -yN             256  \
  -xT              1024  -yT             128  \
  -xMODE        Complex  -yMODE      Complex  \
  -xSW         9000.900  -ySW       2400.000  \
  -xOBS         600.113  -yOBS       150.915  \
  -xCAR           4.725  -yCAR       116.828  \
  -xLAB              H1  -yLAB           C13  \
  -ndim               2  -aq2D        States  \
  -out ./test.fid -verb -ov

sleep 5
```

Figure 6. Preliminary conversion script text

Reading the procpar file is not perfect, and some parameters must be entered manually. For instance, the example data here come from a ^{1}H-^{15}N HSQC spectrum, but the program has defaulted into assuming we're working with a ^{1}H-^{13}C spectrum. Left-click on the > arrows to pull down menu choices or type directly into the boxes to specify values explicitly. Note that the y-axis Acquisition Mode will be set to "Complex," but if sensitivity enhancement was used in the acquisition, you will need to use the "Rance-Kay" option. Otherwise, your spectrum will have its own mirror image superimposed on it. Also, be sure to use the appropriate Axis Label for ^{1}H, the x-axis. Experiments detecting amide ^{1}H's will probably need to have their ^{1}H axis labeled as

"HN." The default output file name is "test.fid;" this can be modified, but because other scripts must use this name to work, it is expeditious to leave it as is. When done, lift-click on "Clear Script," left-click on "Update Script," then left click on "Save script." If you wish, you can click on "Execute Script" also, though you may also execute the script by typing "fid.com" on the command line while in the directory of interest. Here are the windows with the final menu selections and the corresponding script for this HSQC experiment:



Spectrometer Input: ./fid

Output Template: ./test.fid

Output Script: fid.com

Input Protocol: Varian

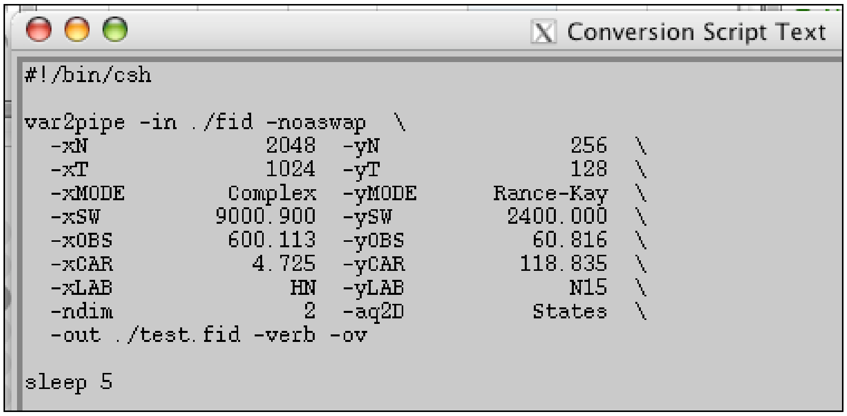
Output Protocol: NMRPipe

Dimension Count: 2

Temperature (K): 303.000

	x-axis	y-axis
Total Points R+I:	2048	256
Valid Points:	1024	128
Acquisition Mode:	Complex	Rance-Kay
Spectral Width Hz:	9000.900	2400.000
Observe Freq MHz:	600.113	60.81586
Center Position PPM:	4.725	118.835
Axis Label:	HN	N15

Figure 7. Conversion GUI with final choices



```
#!/bin/csh

var2pipe -in ./fid -noaswap  \
  -xN              2048  -yN             256  \
  -xT              1024  -yT             128  \
  -xMODE        Complex  -yMODE    Rance-Kay  \
  -xSW         9000.900  -ySW       2400.000  \
  -xOBS         600.113  -yOBS        60.816  \
  -xCAR           4.725  -yCAR       118.835  \
  -xLAB              HN  -yLAB           N15  \
  -ndim               2  -aq2D        States  \
  -out ./test.fid -verb -ov

sleep 5
```

Figure 8. Edited conversion script in GUI window

d) Left-lick "Quit" in the conversion Utility window to exit the conversion GUI.

E. Read in the NMRPipe-format 2D fid.

a) Return to the main GUI.
b) The act of getting the fid into your working space takes two steps: "selecting" and "reading." To start selecting your file, right-click on 'File."
c) Left-click on "test.fid." Double-left-clicking on "test.fid" will also "read" the fid, making it appear in the main window.
d) Left-click "Done."
e) If you single-clicked "test.fid" in step c), then right-click "File" and left click "Read" to display the fid in the main window.

F. Process the first increment

a) Right-click on the “Mouse” button on the main toolbar.

b) Select “h) 1D horizontal. This gives you a straight yellow line and a yellow fid. The left mouse button, when applied in the dark field of the fid, drags the straight yellow line up and down, selecting different increments’ fid’s. Move this cursor line all the way down to select the first increment.
TAKE CARE! As long as you are in “mouse mode,” ANY TIME you click within the black working area, you will select a new fid! So if you go through and process a or partially process a 1D spectrum, but then left-click within the working area, you will abandon the processing and choose a new fid!

c) Familiarize yourself with the functions of the different mouse buttons in the right purple margin. Holding down the right mouse button and moving the mouse moves the fid up and down. Holding down the middle mouse button and moving the mouse increases and decreases the scale of the fid. When you have processed the fid into a spectrum, these mouse buttons will retain these functions.

d) Right-click on the “Proc” button of the main toolbar. Select “N) NMRPipe command.” You will see a new window pop up:

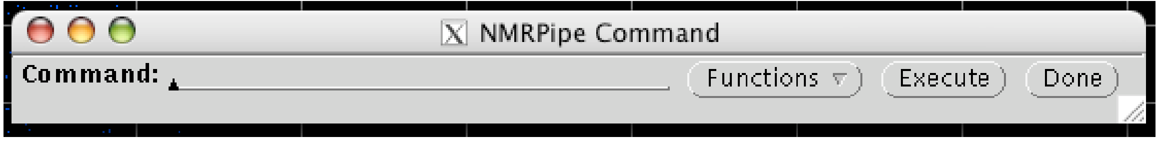


Figure 9. NMRPipe Command window

You can either type commands directly into the line beyond the “Command:” directive, or you can right-click on the “Functions” menu button to see an array of processing choices.

Note that the text you see or type on this command line is exactly what goes into the processing scripts. Recall that the NMRPipe GUI exists primarily to help you write those scripts. This line is where you can test the effects of alternate commands or parameter values.

e) Now start processing your first increment. Here, we’ll go through a standard processing routine, including everything that’s essential, but nothing fancy.

f) Right-clicking on the “Function” button yields the following window:

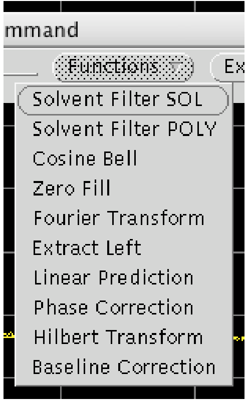


Figure 10. Command "Functions" Menu

g) First, apply solvent subtraction. Under some circumstances, this may be unnecessary or undesirable, but here we’d like it. If you end up with a glitch in your spectrum worse than a giant peak in your spectrum, then reprocess and leave this step out. Left-click on “Solvent Filter SOL” to invoke solvent subtraction. Either left-click in the command line and hit your “return” key, or left-click on “Execute.” To invoke the command. You should see your fid change shape so that it oscillates about a straight horizontal line.

h) Second, apply your apodization function. Right-click on the ‘Functions” button, left click on “Cosine bell,” then left-click on “Execute” to apply a cosine-bell (90°-shifted sine-bell) window function. Please see the appendix on apodizations for a thorough discussion of the meanings of the parameters and the shapes of their corresponding functions. Write down the parameters used for this function because you’ll need to enter them into your script later. After application, you should see the fid with a new envelope shape, but the change may be subtle.

i) Third, apply your zero-filling. Right-click on “Functions,” then left-click on “Zero Fill.” You should see the fid shift to the left half of your working area.

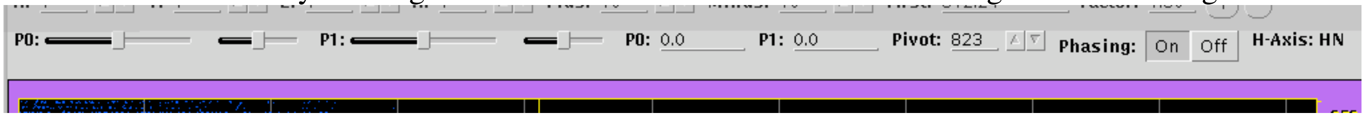
j) Fourth, perform your Fourier transform. Right-click on “Functions”, left click on “Fourier Transform,” then left-click “Execute.” Your fid should become a spectrum now, though you will need to adjust its phase.

k) Left-click “Done” in the NMRPipe Command window.

l) Adjust the vertical scale and position of the spectrum by center-clicking/holding/dragging and right-clicking/holding/dragging in the right-hand purple border region.

m) Adjust the phase.

   a. Orient yourself with the phasing controls at the top of the NMRPipe window (Figure 11). At the left are the zeroth and first order (P0 and P1, respectively) phase controls. Each has a coarse and a fine adjustment bar. Operate these by left-clicking on the gray rectangle and dragging it left and right. The P0 and P1 phase correction values appear in the P0: and P1: displays in the middle. These are very important because you must manually edit the processing macros to reflect the numbers in these displays. The pivot point display informs you which point number you’ve selected at which to pivot. Phasing mode is entered and exited by clicking the “On” or “Off” buttons after the “Phasing:” label at the right.



P0: 0.0 P1: 0.0 Pivot: 823 Phasing: On Off H-Axis: HN

Figure 11. Phasing controls

   b. Select the location of your pivot point. Left-click/hold/drag in the purple border region at the BOTTOM of the NMRPipe window. Two little yellow arrows pointing at each other will appear at the top and bottom of the working area. Place the arrows so they point at the rightmost peak in the spectrum (Figure 12).

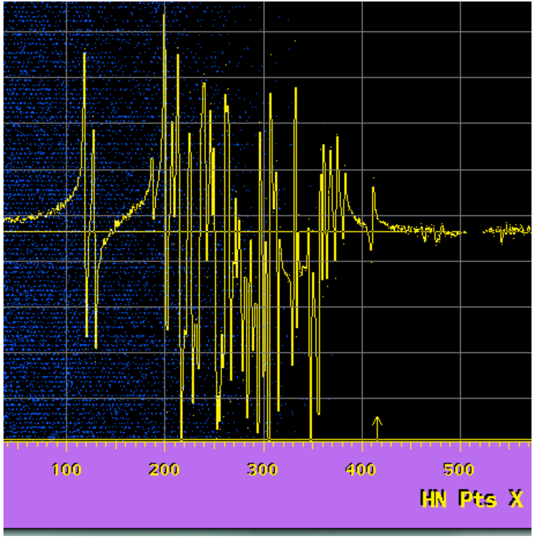


Figure 12. Placement of phasing pivot point

   c. Enter active phasing mode by clicking the “On” button after the “Phasing:” label in the upper right of the NMRPipe window (Figure 12).
   d. Adjust the zeroth-order phase. Use the slider bar labeled “P0:” in the upper left of the NMRPipe window to phase the rightmost peak (Figure 11).
   e. Adjust the first-order phase. Use the slider bar labeled “P1:” in the upper left of the NMRPipe window to phase the rest of the spectrum (Figure 11).
   f. Once you are satisfied with the phasing of your spectrum, write down the P0 and P1 values so you can use them in your processing script.

n) If you wish to see what your spectrum looks like after baseline correction, right-click on “Proc” in the main NMRPipe window toolbar, select “N) NMRPipe Command,” right-click ‘Functions,” then left-click on “Baseline Correction.” The default settings simply adjust for a DC offset, which is highly recommended. Higher order corrections can be achieved by varying the polynomial order number at the end of the script line; e.g. changing the 1 to a 3 sets you up to confer a third-order baseline correction. Left-click the “Execute” button to make the baseline correction happen.

o) It is most common for ^{15}N-edited experiments to have peaks only in the left half of the spectrum. NMRPipe lets you abandon the useless right half of these spectra with the command “Extract Left” in the NMRPipe command window “Functions” menu. This is entirely unexciting to perform on the first increment in the NMRPipe GUI, but it’s useful to include in the script. Go ahead if you’re curious, though.

p) You’re done with the first increment! You’ll find that this is the most labor-intensive portion of 2D spectrum processing.

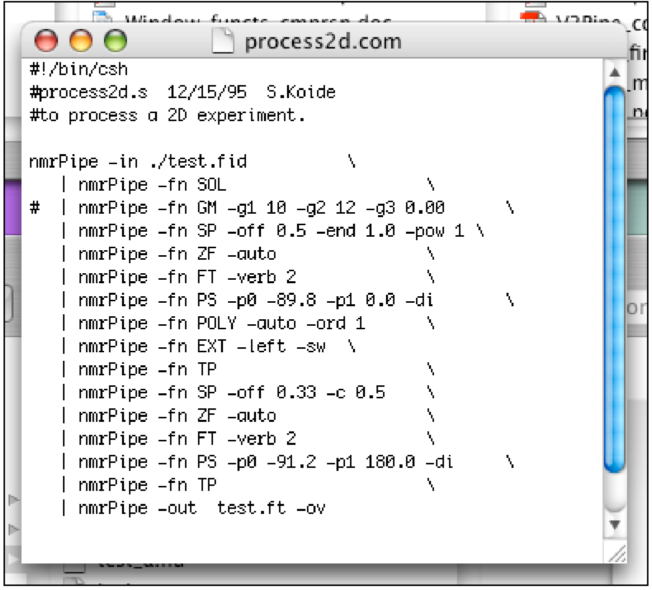
G. Edit the script “process2d.com”

a) You now need to switch to a text editor program so you can edit the script entitled “process2d.com.” You’ll come back to NMRPipe in a minute, so there’s no need to shut it down.

b) Locate a copy of the script “process2d.com,” copy it, and place it in your spectrum’s directory, which should now also contain Varian’s normal files; fid, procpar, text, and log; the script fid.com, and the NMRPipe-format fid, test.fid.

c) Open up a non-MS Word editor program like “TextEdit,” and open up the appropriate file “process2d.com.” On the Mac, double clicking “process2d.com” will probably open it up in TextEdit.

d) Orient yourself. Your script should look something like this when you open it up:



```
#!/bin/csh
#process2d.s  12/15/95  S.Koide
#to process a 2D experiment.

nmrPipe -in ./test.fid          \
   | nmrPipe -fn SOL                    \
#  | nmrPipe -fn GM -g1 10 -g2 12 -g3 0.00      \
   | nmrPipe -fn SP -off 0.5 -end 1.0 -pow 1 \
   | nmrPipe -fn ZF -auto               \
   | nmrPipe -fn FT -verb 2             \
   | nmrPipe -fn PS -p0 -89.8 -p1 0.0 -di       \
   | nmrPipe -fn POLY -auto -ord 1      \
   | nmrPipe -fn EXT -left -sw  \
   | nmrPipe -fn TP                     \
   | nmrPipe -fn SP -off 0.33 -c 0.5    \
   | nmrPipe -fn ZF -auto               \
   | nmrPipe -fn FT -verb 2             \
   | nmrPipe -fn PS -p0 -91.2 -p1 180.0 -di     \
   | nmrPipe -fn TP                     \
   | nmrPipe -out  test.ft -ov
```

Figure 13. Initial "process2d.com" script

These lines should look familiar to you since they have the same format as those that appeared in the NMRPipe command window (Figure 9). The chief difference is the presence of the initiating clause “| NMRPipe –fn,” which simply tells you computer to interpret the following statements in terms of the program NMRPipe. The “#” characters at the left comment out the remainder of the line; this is handy to keep things like GM apodization functions ready in case you want them in future work, but not in the current work.

Here’s what’s happening line-by-line, after the initial commentary:

a. “-in ./test.fid” reads in the input file, specified as the file “test.fid” in the current directory. If you wish, you could specify an entire pathname for the file. Also, the file could have a different name, but processing ends up being much easier if you don’t have to keep switching names of these intermediate files, which requires you to edit those portions of your macro.
b. “SOL” performs solvent subtraction.
c. “GM” will not be used because it’s commented out. GM applies a lorentz-gaussian apodization function. See the appendix for more details.
d. “SP” applies a cosine-bell apodization window with the parameters shown. See the appendix for more details.
e. “ZF” applies zero filling.
f. “FT” applies Fourier transformation.
g. “PS” adjusts the F2 phase with values P0 and P1.
h. “POLY” applies baseline correction, but for “-ord 1,” it is just a DC offset correction.
i. “EXT –left ” extracts the left half of the spectrum.
j. “TP” transposes the spectrum to enable processing of the indirect dimension.
k. The subsequent statements have the same meaning as normal, but they’re applied to the t1/F1 dimension.
l. “-out test.ft” writes the processed spectrum to a new file, “test.ft,” in the current directory.

e) Edit what you need to in the first half of the script (i.e., above the first “TP” statement).
   a. Check the apodization function parameter values. They should match those that you used before when processing the first increment. The values used for GM and SP are pretty standard, and can be safely applied to most spectra. SP is the simplest apodization, and GM is commonly used for resolution enhancement.
   b. Edit the P0 and P1 values so they equal the values you wrote down when processing the first increment. There is a special consideration to note with regard to using the GUI P0 and P1 numbers in the script, but that is better introduced later, when adjusting the phase of the second dimension.
   c. Alter other parameters, such as baseline correction polynomial order, if you feel like it will be useful for improving the spectrum. The script shown above should produce a good spectrum under most circumstances.

f) Edit parameters for processing the indirect dimension. You don’t get to see a 1D spectrum to help you alter these parameters. In the NMRPipe procedure, you just process the 2D spectrum as a whole, read it into the GUI, make observations and phase adjustments in the GUI, then return to this script, alter it, and reprocess the spectrum.
   a. Edit the apodization function. In t1, you shouldn’t use the GM function; just use the SP function. I’d change the SP line to read “SP –off 0.5 –end 1.0 –c 0.5   \” to yield a regular cosine bell function.
   b. Consider adding a line for linear prediction. In this spectrum, it won’t be necessary, but it you have so few t1 increments that you get f1 wiggles on your peaks because of truncation, then insert the line “LP –auto –pred 64” or something like that. Place this line between the

TP line and the SP line. For details on the LP line syntax, type in the terminal command line "NMRPipe –help –fn LP."

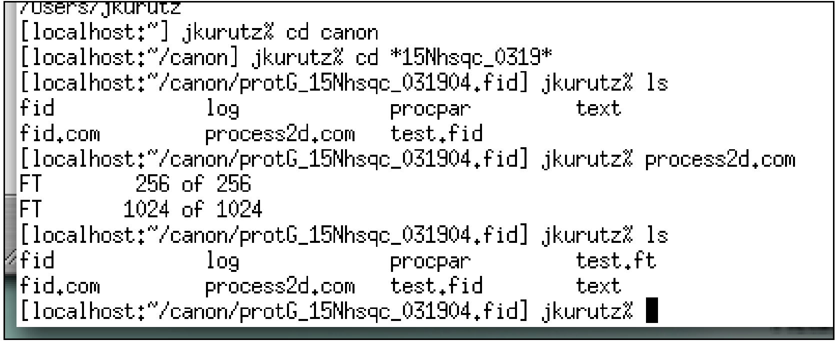
c. Don't worry yet about the f1 P0 and P1 values. You need to see the spectrum, correct the f1 phase, and then add the corrections to the values in this script.

g) Save the script. It will be convenient to retain the name "process2d.com." You'll find it worthwhile to keep the file open in the text editor because you'll need to go back to edit the f1 phase values.

H. Execute the script "process2d.com"

a) This step is trivial. Get to an X11 terminal window and cd to the directory of the spectrum you've been working with.

b) Type "process2d.com." You should first see one counter counting the f1 increments that have been processed, then a second counter showing the f1 dimension being processed. After a couple of seconds (on a modern computer), it will be over, and a new file, "test.ft" should be in the directory. Here is what this whole process should look like:



```
/users/jkurutz
[localhost:~] jkurutz% cd canon
[localhost:~/canon] jkurutz% cd *15Nhsqc_0319*
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% ls
fid             log             procpar         text
fid.com         process2d.com   test.fid
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% process2d.com
FT        256 of 256
FT       1024 of 1024
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% ls
fid             log             procpar         test.ft
fid.com         process2d.com   test.fid        text
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% ▋
```

Figure 14. Execution of the macro "process2d.com"

I. Read the spectrum "test.ft" in the NMRPipe GUI

a) Return to the NMRPipe GUI.

b) Right-click on the "File" button in the NMRPipe main toolbar.

c) Left-click on "Select file."

d) Double click on "test.ft."

e) A beautiful 2D spectrum should appear in your working area.

f) The contour level at which you're viewing the spectrum will probably be suboptimal. Right-click on the "Draw" button in the main toolbar, left-click to select "T) Estimate Noise," then left-click the "continue" button in the new window that pops up. Now that the program knows about the noise level, right-click on the "Draw" button in the main toolbar, and left-click to select "contours." This adjusts your contour level based on the noise in your spectrum, and will give you something that looks reasonable, though perhaps too noisy.

g) Manually adjust the contour level with the "+" and "-" buttons in the upper right corner of the NMRPipe main window. This won't have an effect until you left-click on the "Draw" button in the main toolbar. The convention is that "+" raises the lowest observed contour level;, thus making observation of noise less likely, and that "-" has the opposite effect.

h) Evaluate the spectrum for obviously odd behavior. It is at this point that you'll discover whether the Varian fid was converted to NMRPipe format correctly. If your crosspeaks look like they're duplicated, with the duplicate set being a mirror image symmetric about the center of the indirect dimension, you probably did not specify the correct mode of F1 phasing in the script fid.com; you probably need the "Rance-Kay" treatment rather than the "Complex" or "States" treatments. If it is reflected about the horizontal center line without being duplicated, you will probably need to include the modifier "-neg" to your second FT statement in "process2d.com."

i) If everything looks good, then you are done processing this spectrum and should skip to its conversion to NMRView format.

J. Phase the F1 dimension

a) Right-click on the "Mouse" button in the main toolbar.

b) Left-click on "v) 1D Vertical." This gives you a vertical line and a vertical spectrum corresponding to the trace under the line. Like with the horizontal cursor used in the 1st increment processing, position of the vertical line cursor is determined by left-clicking/holding/dragging in the workspace area, vertical scale is controlled by center-clicking/holding/dragging in the purple margin to the right of the spectrum, and position of the vertical 1D spectrum is governed by right-clicking/holding/dragging in the purple margin to the right of the spectrum.

c) Select a spectrum that contains a crosspeak at the top of the working area. Move it somewhere out of the way, and increase its vertical scale so you can observe the baseline around the top peak.

d) Right-click "1D" in the main toolbar, left-click "a) Append 1D." This will keep the spectrum where you put it while you select another 1D vertical spectrum.

e) Select another 1D vertical spectrum this time choosing one with a crosspeak toward the bottom of the 2D spectrum.

f) Left-click in the right-hand purple margin to bring up a pair of small yellow arrows pointed at one another. Left-click/hold/drag them so that they point at the topmost peak. This is about what things should look like in your working area:

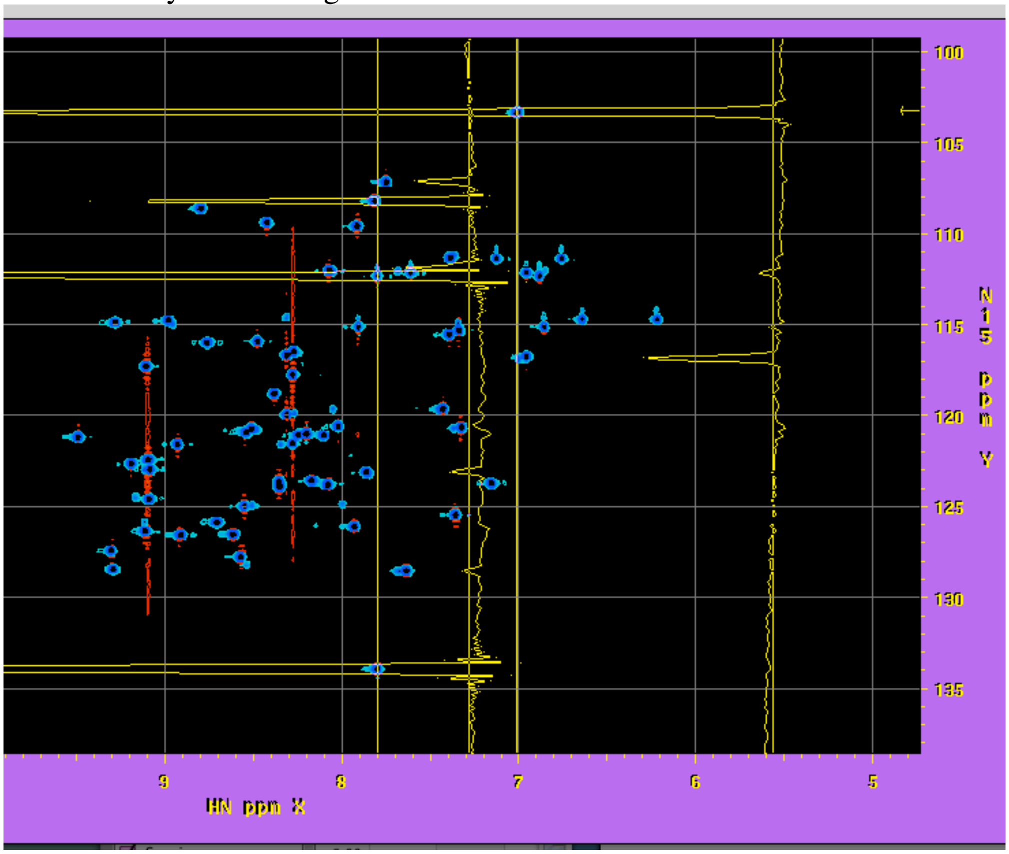


Figure 15. Phasing F1, two vertical traces selected

This spectrum happened to be phased quite well by the initial "process2d.com" script.

g) Enter active phasing mode by left-clicking the “On” button at the right side of the phase controls toolbar (Figure 11).
h) Phase the topmost peak with P0 and the bottom-most peak with P1. Note that the 2D spectrum will not reflect these changes; only the 1D traces will respond.
i) Record the P0 and P1 values appearing in their windows in the phase controls toolbar.

K. Edit the script “process2d.com” again

a) Return to your editor with “process2d.com” open.
b) ADD the values of P0 and P1 to those in the lower “PS” line. For example, if your original script line was: | NMRPipe –f PS –p0 3.2 –p1 0.0 –di \
and your new P0 and P1 corrections are –85.0 and 177.0, respectively, then your new line should read: | NMRPipe –f PS –p0 –81.8 –p1 177.0 –di \
c) Make other alterations you find beneficial, such as adding a 2nd order baseline correction to f1:
| NMRPipe –fn POLY –auto –ord 2 \
d) Save the modified script as”process2d.com” again.

L. Re-run the script “process2d.com”

a) Like with step H, return to an X11 terminal and cd to the directory of your spectrum.
b) Type “process2d.com.”

M. View the reprocessed 2D spectrum in the NMRPipe GUI

a) As in step I, open the spectrum “test.ft.”
b) Study the spectrum and its 1D vertical traces. If all is OK, then exit NMRPipe. If not, go back and edit process2d.com, execute the script, and re-view it until it looks right.

N. Convert the NMRPipe file “test.ft” to NMRView format

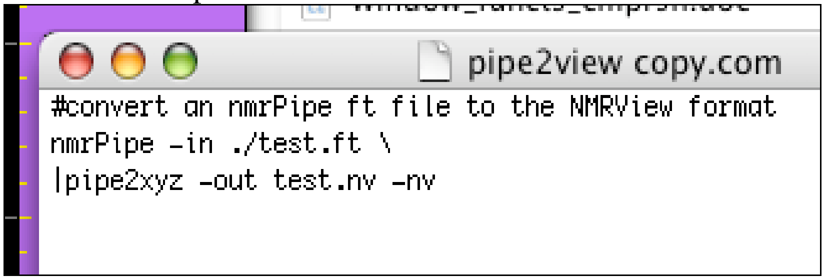
a) Your NMRPipe spectrum can be converted to NMRView format in either of two convenient ways: using a one-line command on the command line, or using a one-word script which must be created or copies from somewhere else.

b) Go to an X11 terminal window and cd to the directory of your spectrum.

c) Do either of the following:

   a. Type “pipe2xyz –in test.ft –out ./test.nv -nv”

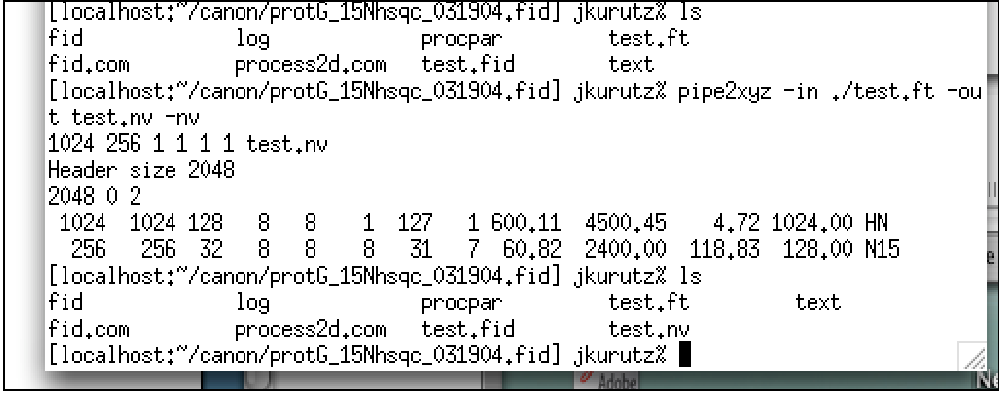
   b. Copy the script “pipe2view.com” into your working directory, then type “pipe2view.” Here is what this script looks like inside:



```
#convert an nmrPipe ft file to the NMRView format
nmrPipe -in ./test.ft \
|pipe2xyz -out test.nv -nv
```

Figure 16. The pipe2view.com script

d) This is what your terminal should show you after the conversion:



```
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% ls
fid             log             procpar         test.ft
fid.com         process2d.com   test.fid        text
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% pipe2xyz -in ./test.ft -ou
t test.nv -nv
1024 256 1 1 1 1 test.nv
Header size 2048
2048 0 2
 1024  1024 128    8   8    1  127   1 600.11  4500.45    4.72 1024.00 HN
  256   256  32    8   8    8   31   7  60.82  2400.00  118.83  128.00 N15
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% ls
fid             log             procpar         test.ft         text
fid.com         process2d.com   test.fid        test.nv
[localhost:~/canon/protG_15Nhsqc_031904.fid] jkurutz% 
```

Figure 17. Results of pipe2view conversion

O. You’re done!

3) Processing 3D Data

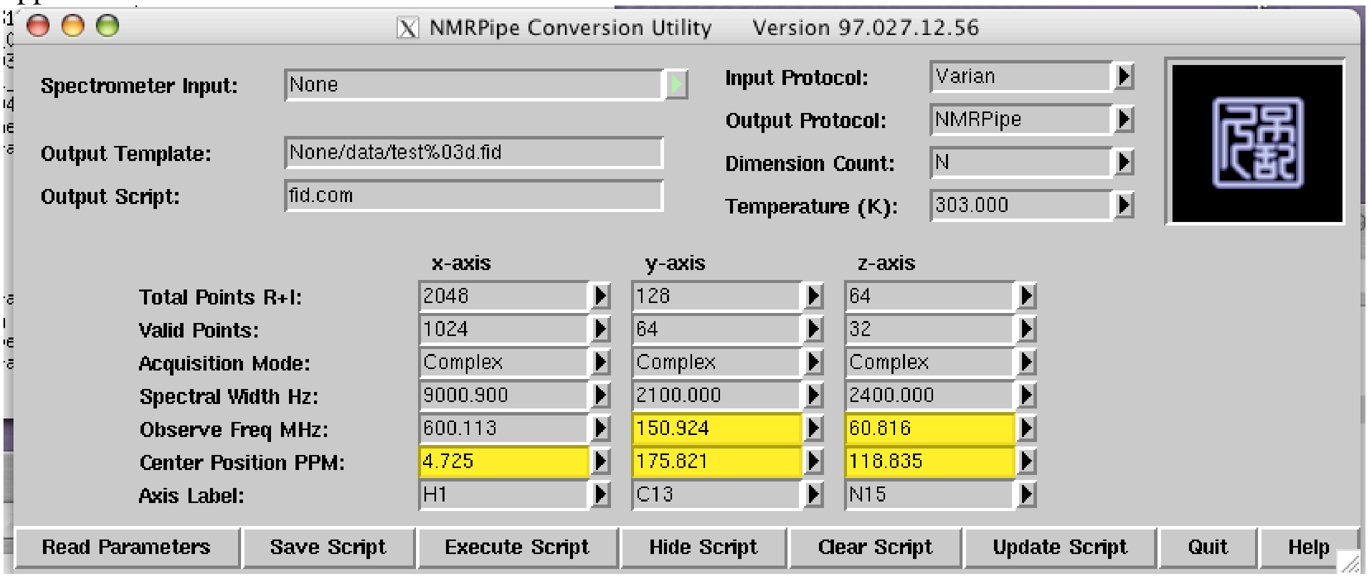
3D Datasets present no great challenge when processing. Just as one might suspect, the spectroscopist converts the data to nmrPipe format, looks at the first increment to determine F3 phasing, edits a script that performs the processing, then executes the script. Note that the data manipulation takes significantly more time to execute than for 1D and 2D data, but this imposes no great burden you.

A. Convert the data

First, cd to your spectrum’s .fid file and ensure that you see the file “fid”. Sometimes 3D data is large enough that people compress it, in which case you’ll see a file like “fid.gzip,” in which case you’ll need to uncompress your fid. Before starting, you should ensure you have the following scripts (or ones like them) in your directory:

process2D3.com	(processs 1st 2D plane of a 3D spectrum)
process3d1.com	(processes t1xt2xt3 data into t1xF2xF3 data)
process3d3.com	(processes t1xF2xF3 data into F1xF2xF3 spectrum)
process3d_lp2.com	(optional - processes t1xt2xt3 data into F1xF2xF3 spectrum with linear prediction in t1 and t2 dimensions – if using this, the process3d1 and 3d3 scripts are unnecessary.)
pipe2view_3d.com	(converst 3D nmrPipe data into 3D nmrView data)

Like converting 2D data to nmrPipe format (Section 2)D), you can either type "varian" to call up the conversion interface, or you can type "nmrDraw" to start the larger nmrPipe program, right-click on the "Proc" menu button (Figure 3), then select "Convert Varian." You'll see the familiar initial GUI (Figure 4). This time, when you click "Read Parameters," all three columns will remain and meaningful data will appear:



NMRPipe Conversion Utility Version 97.027.12.56

Spectrometer Input: None

Output Template: None/data/test%03d.fid

Output Script: fid.com

Input Protocol: Varian

Output Protocol: NMRPipe

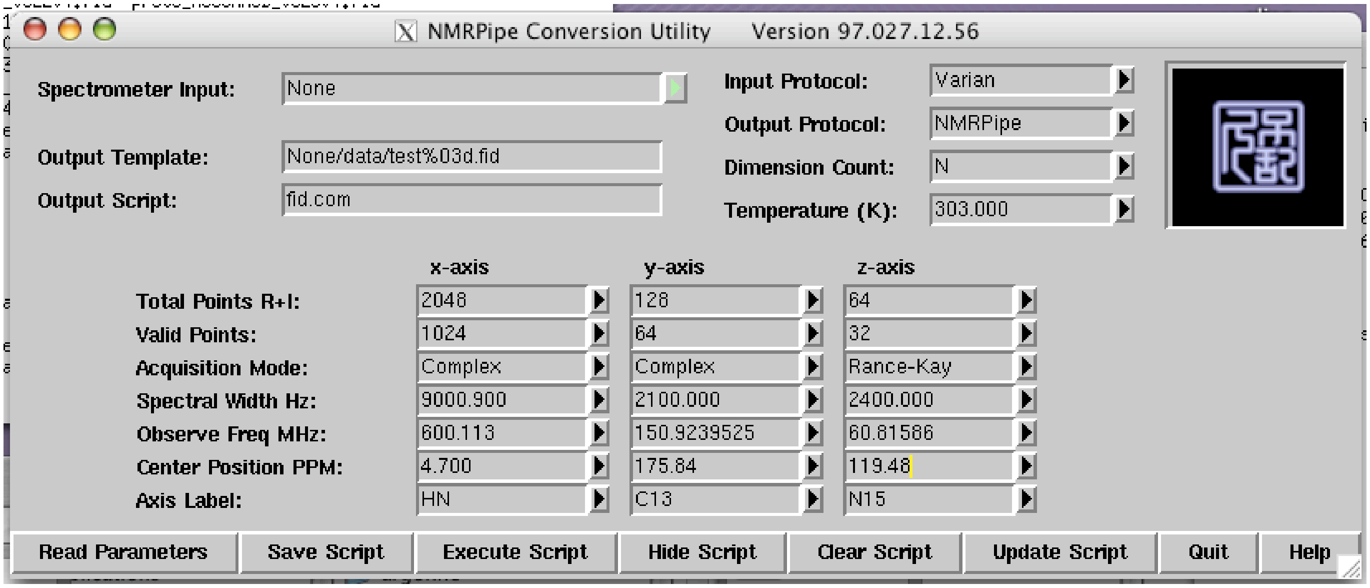
Dimension Count: N

Temperature (K): 303.000

	x-axis	y-axis	z-axis
Total Points R+I:	2048	128	64
Valid Points:	1024	64	32
Acquisition Mode:	Complex	Complex	Complex
Spectral Width Hz:	9000.900	2100.000	2400.000
Observe Freq MHz:	600.113	150.924	60.816
Center Position PPM:	4.725	175.821	118.835
Axis Label:	H1	C13	N15

Figure 18. "Varian" conversion utility after reading parameters for a 3D dataset (hnco).

Then change the relvant parameters as you normally would, making sure your Center Position frequencies, in ppm, are correct and consistent with the others in your dataset:



Spectrometer Input: None

Output Template: None/data/test%03d.fid

Output Script: fid.com

Input Protocol: Varian

Output Protocol: NMRPipe

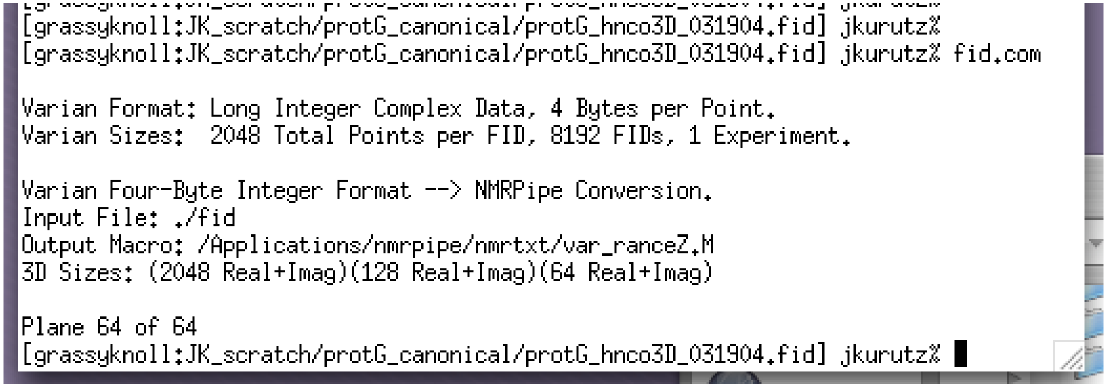
Dimension Count: N

Temperature (K): 303.000

	x-axis	y-axis	z-axis
Total Points R+I:	2048	128	64
Valid Points:	1024	64	32
Acquisition Mode:	Complex	Complex	Rance-Kay
Spectral Width Hz:	9000.900	2100.000	2400.000
Observe Freq MHz:	600.113	150.9239525	60.81586
Center Position PPM:	4.700	175.84	119.48
Axis Label:	HN	C13	N15

Figure 19. Varian conversion utility, after entering correct parameters for this dataset (hnco, whose 13C range center on C'=O carbons ~175 ppm)

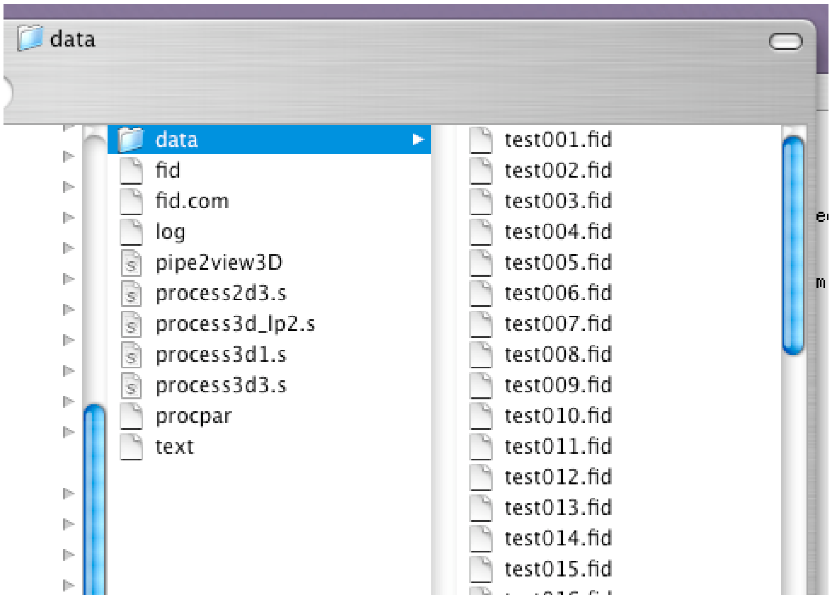
Once your parameters are entered, go through the same procedure you would for a 2D spectrum: clear the script, update it, save it as “fid.com,” then execute it.



```
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz%
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% fid.com

Varian Format: Long Integer Complex Data, 4 Bytes per Point.
Varian Sizes:  2048 Total Points per FID, 8192 FIDs, 1 Experiment.

Varian Four-Byte Integer Format --> NMRPipe Conversion.
Input File: ./fid
Output Macro: /Applications/nmrpipe/nmrtxt/var_ranceZ.M
3D Sizes: (2048 Real+Imag)(128 Real+Imag)(64 Real+Imag)

Plane 64 of 64
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% █
```

Figure 20. Xwindow after converting varian data to nmrPipe format

This will create a folder full of 2D fids. There will be as many files here as there are complex (R+I) points on your z-axis (usually ^{15}N) :



```
data
fid
fid.com
log
pipe2view3D
process2d3.s
process3d_lp2.s
process3d1.s
process3d3.s
procpar
text
```

```
test001.fid
test002.fid
test003.fid
test004.fid
test005.fid
test006.fid
test007.fid
test008.fid
test009.fid
test010.fid
test011.fid
test012.fid
test013.fid
test014.fid
test015.fid
```

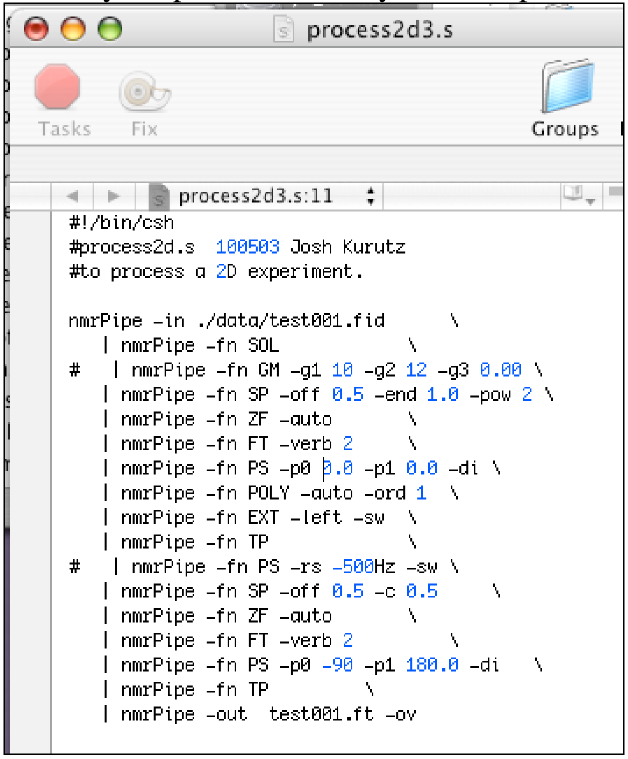
Figure 21. Folder and 2D FIDs created by conversion process

B. Process & phase the first plane.

You need to edit your scripts so the phase will be correct, just as you did with 2D spectra. To get your phase information, just investigate the first plane (test001.fid). You can either

A) start nmrPipe, read in the file test001.fid, and process it there, or

B) use a script to process the spectrum, then read the processed spectrum, test001.ft, into nmrPipe. Here's a handy script to do this for you called process2d3.s:



```
#!/bin/csh
#process2d.s  100503 Josh Kurutz
#to process a 2D experiment.

nmrPipe -in ./data/test001.fid      \
   | nmrPipe -fn SOL                \
#   | nmrPipe -fn GM -g1 10 -g2 12 -g3 0.00 \
   | nmrPipe -fn SP -off 0.5 -end 1.0 -pow 2 \
   | nmrPipe -fn ZF -auto           \
   | nmrPipe -fn FT -verb 2         \
   | nmrPipe -fn PS -p0 0.0 -p1 0.0 -di \
   | nmrPipe -fn POLY -auto -ord 1   \
   | nmrPipe -fn EXT -left -sw   \
   | nmrPipe -fn TP                 \
#   | nmrPipe -fn PS -rs -500Hz -sw \
   | nmrPipe -fn SP -off 0.5 -c 0.5      \
   | nmrPipe -fn ZF -auto           \
   | nmrPipe -fn FT -verb 2              \
   | nmrPipe -fn PS -p0 -90 -p1 180.0 -di    \
   | nmrPipe -fn TP           \
   | nmrPipe -out  test001.ft -ov
```

Figure 22. process2d3.com, a script for processing the first plane of a 3D dataset.

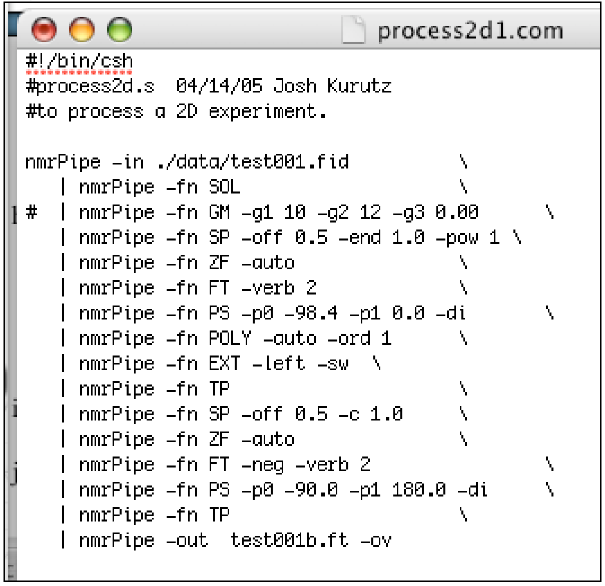
I usually choose option B because it involves less clicking. Once you have a processed spectrum read into nmrPipe, phase it in F3 (the ^{1}H or HN dimension, which was F2 in a 2D spectrum), and record your p0 and p1 phase values, just as you would for a 2D spectrum (Section 2)F.m). Edit your script to add the numbers to the p0 and p1 values already in the script (which is why it's handy to start with p0 and p1 of 0.0). Reprocess the spectrum with your new script and make sure the 2D spectrum looks OK. (i.e, like a normal 2D spectrum). If it all looks OK, then go ahead and enter your phasing parameters into your 3D script(s).

SPECIAL NOTE: The script above (Figure 22) includes a line, shown commented out:

```
nmrPipe –fn PS –rs –500Hz –sw \
```

This line is used to shift the spectrum in the F2 (^{13}C) dimension by 500 Hz. When processing the spectrum originally, I had found that some peaks had "folded over", i.e., they lay outside the sw of the acquired spectrum and were showing up as negative peaks near one edge of the spectrum. By "right-shifting" the FID before performing the FT, effected by using "PS –rs –500Hz", the spectrum was shifted "downward" so that the peaks were all moved downfield by 500 Hz in the ^{13}C dimension. In theory, "-rs –500Hz" should mean "right shift by negative 500 Hz," which would be synonymous with "-ls 500Hz," "left shift by 500 Hz." In practice, the "-ls 500Hz" thing didn't work for some reason, but the "-rs –500Hz" line did. Go figure.

SPECIAL NOTE: for Varian's HNCACB spectrum (and possibly others): The way this experiment is implemented, the ^{13}C axis is flipped, but NMRPipe does not know this *a priori*. This requires that the processing script include a "-neg" modifier on the ^{13}C FT line:



```
#!/bin/csh
#process2d.s  04/14/05 Josh Kurutz
#to process a 2D experiment.

nmrPipe -in ./data/test001.fid          \
   | nmrPipe -fn SOL                    \
#  | nmrPipe -fn GM -g1 10 -g2 12 -g3 0.00      \
   | nmrPipe -fn SP -off 0.5 -end 1.0 -pow 1 \
   | nmrPipe -fn ZF -auto               \
   | nmrPipe -fn FT -verb 2             \
   | nmrPipe -fn PS -p0 -98.4 -p1 0.0 -di       \
   | nmrPipe -fn POLY -auto -ord 1      \
   | nmrPipe -fn EXT -left -sw  \
   | nmrPipe -fn TP                     \
   | nmrPipe -fn SP -off 0.5 -c 1.0     \
   | nmrPipe -fn ZF -auto               \
   | nmrPipe -fn FT -neg -verb 2                \
   | nmrPipe -fn PS -p0 -90.0 -p1 180.0 -di     \
   | nmrPipe -fn TP                     \
   | nmrPipe -out  test001b.ft -ov
```

Figure 23. Process2d3.com script for an HNCACB spectrum, which otherwise will flip the ^{13}C axis.

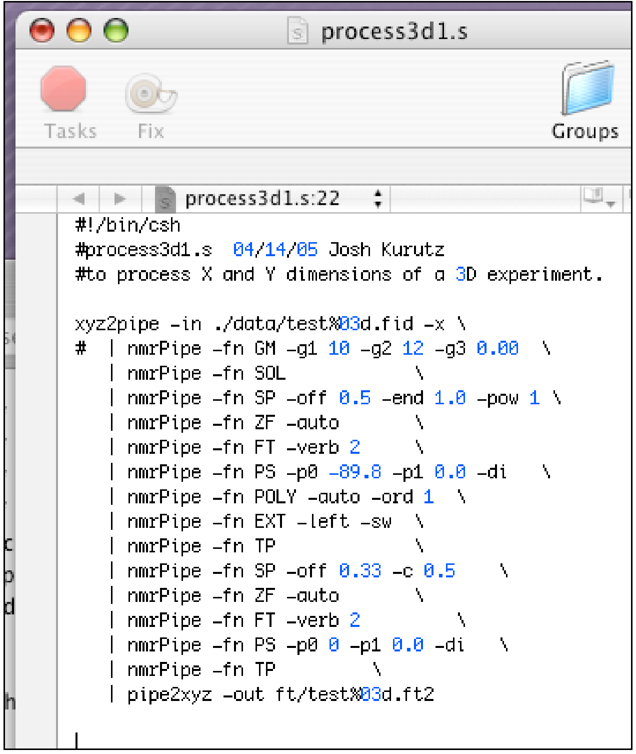
The serine and threonine beta carbons should be negative and in a narrow band at the bottom (high frequency), alpha carbons positive in the center and bottom, and beta carbons negative and at the top. If the ^{13}C axis if flipped, the 2nd dimension's FT transform line should read "FT –neg".

C. 3D processing with 2 scripts, LP in ^{15}N only

The computationally simplest mechanism to generate a workable 3D spectrum is to process the 2D datasets with one script, then handle the third dimension with another script. Typically, you'll be applying linear prediction to the ^{15}N dimension to enhance resolution there. It is computationally trickier, but easier and more useful for the spectroscopist to use a single script that applies linear prediction on both ^{13}C and

^{15}N dimensions. This method is described in the following section and you may wish to skip ahead to it if you've got a fast computer with a capacious hard drive.

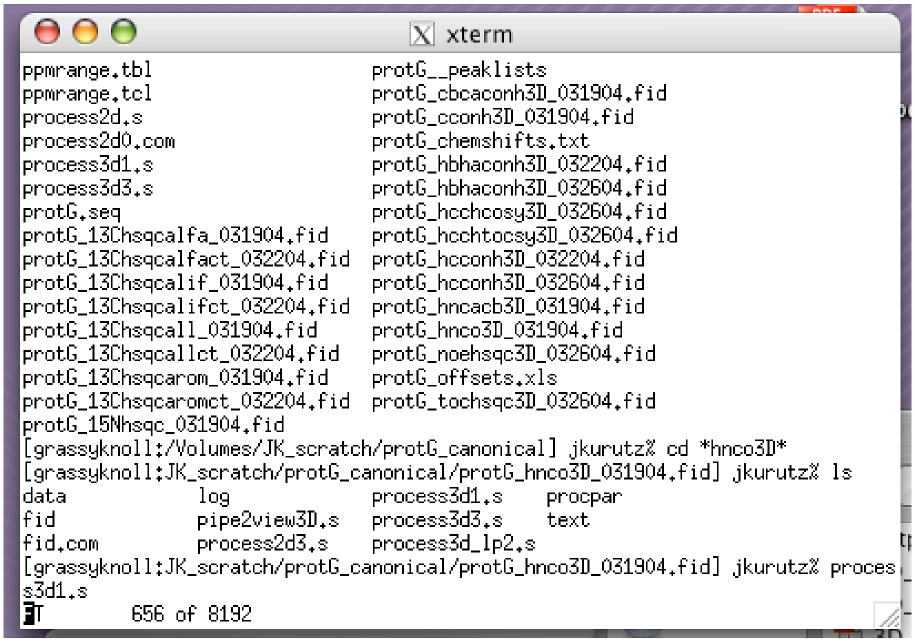
i) Edit your first script, typically called "process3d1.s", to include your F3 phase parameters:



```
#!/bin/csh
#process3d1.s  04/14/05 Josh Kurutz
#to process X and Y dimensions of a 3D experiment.

xyz2pipe -in ./data/test%03d.fid -x \
#   | nmrPipe -fn GM -g1 10 -g2 12 -g3 0.00   \
    | nmrPipe -fn SOL               \
    | nmrPipe -fn SP -off 0.5 -end 1.0 -pow 1 \
    | nmrPipe -fn ZF -auto          \
    | nmrPipe -fn FT -verb 2        \
    | nmrPipe -fn PS -p0 -89.8 -p1 0.0 -di    \
    | nmrPipe -fn POLY -auto -ord 1   \
    | nmrPipe -fn EXT -left -sw   \
    | nmrPipe -fn TP                \
    | nmrPipe -fn SP -off 0.33 -c 0.5     \
    | nmrPipe -fn ZF -auto          \
    | nmrPipe -fn FT -verb 2            \
    | nmrPipe -fn PS -p0 0 -p1 0.0 -di    \
    | nmrPipe -fn TP            \
    | pipe2xyz -out ft/test%03d.ft2
```

Figure 24. A process3d1.s script, which transforms the t1xt2xt3 dataset into a t1xF2xF3 series of 2D spectra.

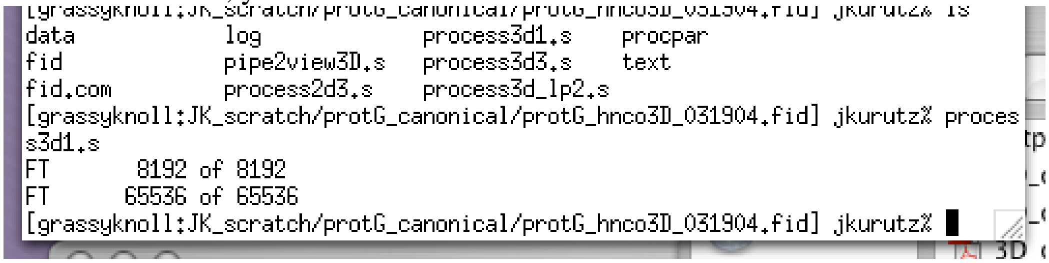
ii) Execute the script by typing "process3d1.s". You should see something like this going on in your X window:



```
ppmrange.tbl                    protG__peaklists
ppmrange.tcl                    protG_cbcaconh3D_031904.fid
process2d.s                     protG_cconh3D_031904.fid
process2d0.com                  protG_chemshifts.txt
process3d1.s                    protG_hbhaconh3D_032204.fid
process3d3.s                    protG_hbhaconh3D_032604.fid
protG.seq                       protG_hcchcosy3D_032604.fid
protG_13Chsqcalfa_031904.fid    protG_hcchtocsy3D_032604.fid
protG_13Chsqcalfact_032204.fid  protG_hcconh3D_032204.fid
protG_13Chsqcalif_031904.fid    protG_hcconh3D_032604.fid
protG_13Chsqcalifct_032204.fid  protG_hncacb3D_031904.fid
protG_13Chsqcall_031904.fid     protG_hnco3D_031904.fid
protG_13Chsqcallct_032204.fid   protG_noehsqc3D_032604.fid
protG_13Chsqcarom_031904.fid    protG_offsets.xls
protG_13Chsqcaromct_032204.fid  protG_tochsqc3D_032604.fid
protG_15Nhsqc_031904.fid
[grassyknoll:/Volumes/JK_scratch/protG_canonical] jkurutz% cd *hnco3D*
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% ls
data            log             process3d1.s    procpar
fid             pipe2view3D.s   process3d3.s    text
fid.com         process2d3.s    process3d_lp2.s
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% proces
s3d1.s
XT        656 of 8192
```

Figure 25. X window during the execution of the process3d1.s script

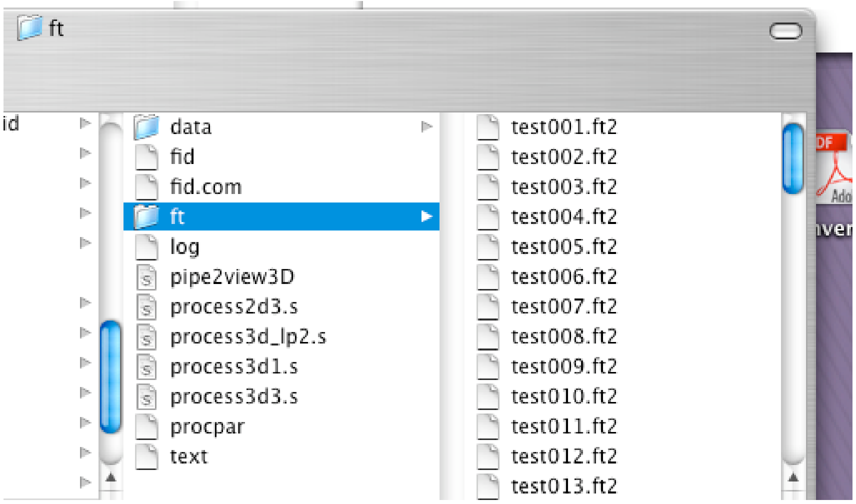
When it's over, your window should look like this:



```
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% ls
data              log               process3d1.s    procpar
fid               pipe2view3D.s     process3d3.s    text
fid.com           process2d3.s      process3d_lp2.s
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% proces
s3d1.s
FT       8192 of 8192
FT      65536 of 65536
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz%
```

Figure 26. X window after execution of script process3d1.s

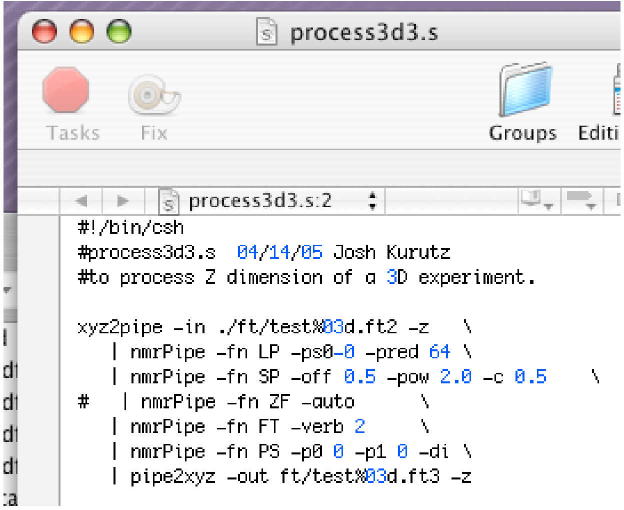
You should now have a new folder entitled "ft" in your directory, and it should be full of processed 2D spectra:



```
data                 test001.ft2
fid                  test002.ft2
fid.com              test003.ft2
ft                   test004.ft2
log                  test005.ft2
pipe2view3D          test006.ft2
process2d3.s         test007.ft2
process3d_lp2.s      test008.ft2
process3d1.s         test009.ft2
process3d3.s         test010.ft2
procpar              test011.ft2
text                 test012.ft2
                     test013.ft2
```

Figure 27. The folder & files created by process3d1.s

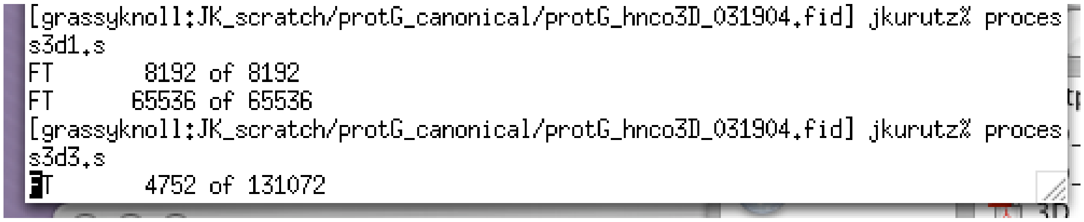
iii) Processing the ^{15}N dimension should require no editing of the process3d3.s script unless you wish to change the weighting function or linear prediction. Here's a good script:



```
#!/bin/csh
#process3d3.s  04/14/05 Josh Kurutz
#to process Z dimension of a 3D experiment.

xyz2pipe -in ./ft/test%03d.ft2 -z   \
   | nmrPipe -fn LP -ps0-0 -pred 64 \
   | nmrPipe -fn SP -off 0.5 -pow 2.0 -c 0.5     \
#    | nmrPipe -fn ZF -auto       \
   | nmrPipe -fn FT -verb 2      \
   | nmrPipe -fn PS -p0 0 -p1 0 -di \
   | pipe2xyz -out ft/test%03d.ft3 -z
```

Figure 28. A good process3d3.s script

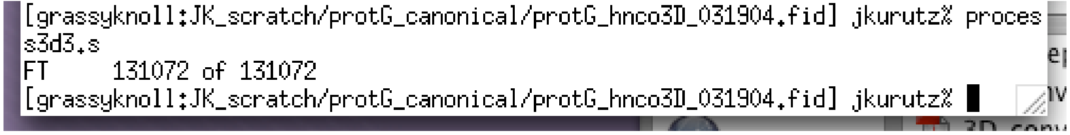
iv) In your X window, type "process3d3.s" to execute this script. This one can take a while to run: 5-20 minutes, depending on your computer. You should see A screen that looks like this during the processing:



```
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% proces
s3d1.s
FT       8192 of 8192
FT      65536 of 65536
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% proces
s3d3.s
FT       4752 of 131072
```

Figure 29. An X window screed during execution of process3d3.s

When it's done, you should see something like:



```
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% proces
s3d3.s
FT     131072 of 131072
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% █
```

You should also see new files ending in .ft3 in the ft folder:

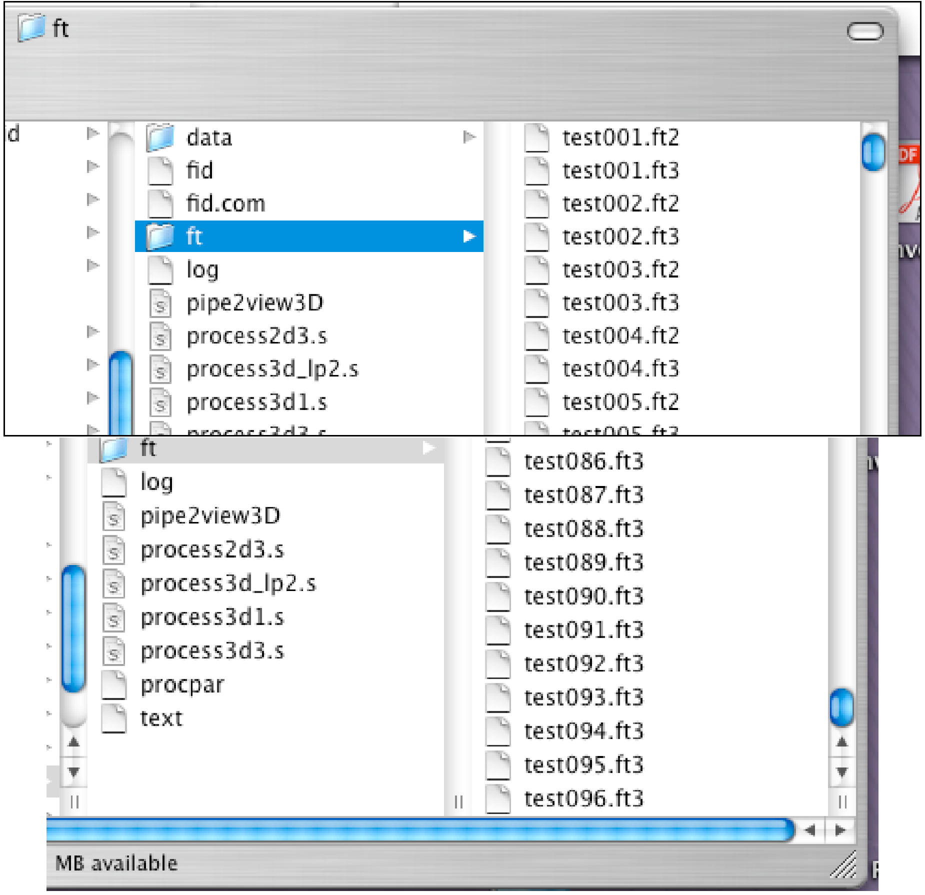


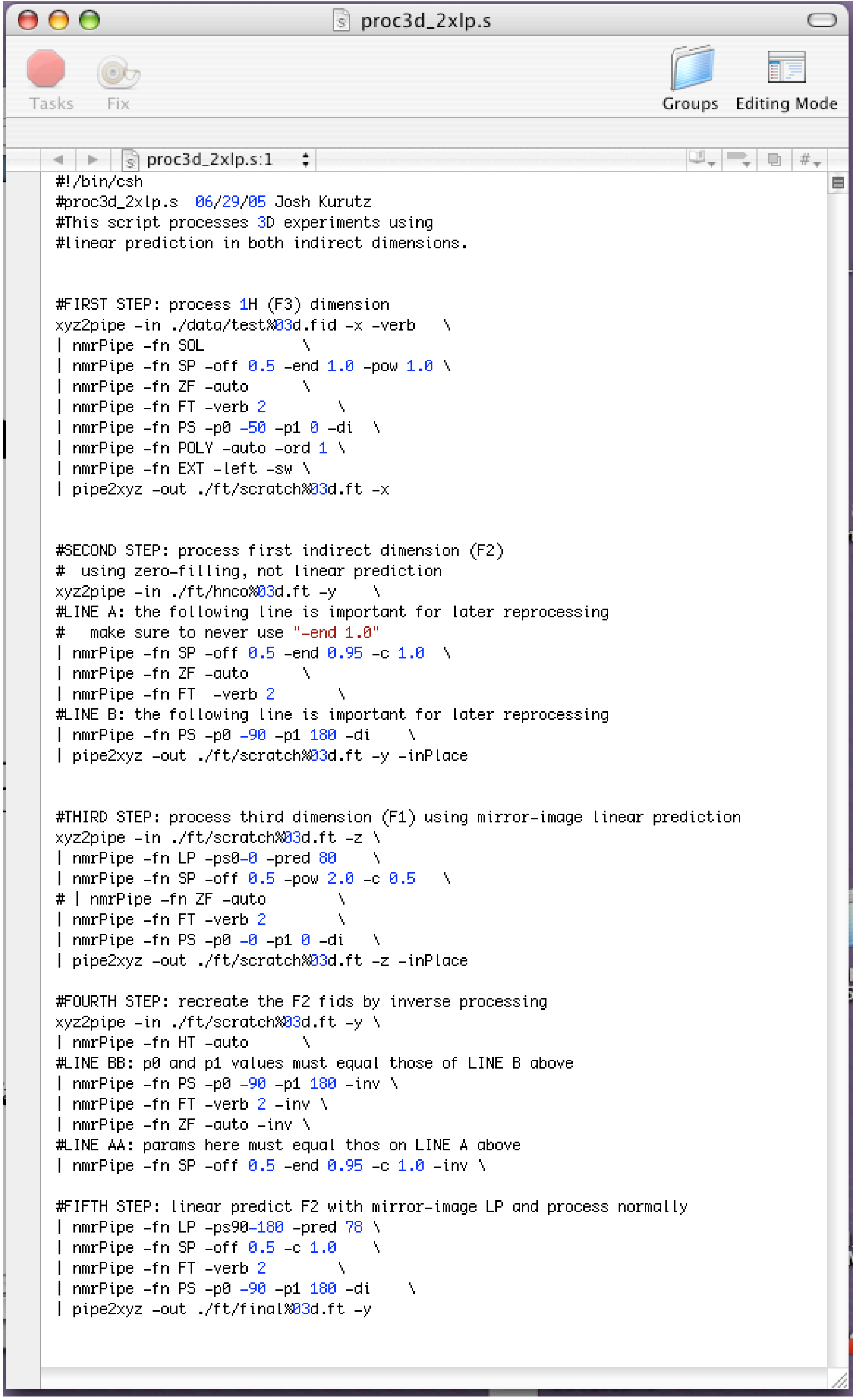
Figure 30. *.ft3 files created by process3d3.s

Each of these .ft3 files is a 2D plane (F2xF3, ^{1}Hx^{13}C) of the 3D spectrum. Note that there are more .ft3 planes than .ft2 spectra because of linear prediction in ^{15}N.

D. Processing 3D data using one script with LP in ^{13}C and ^{15}N

One drawback of the procedure above is you can only linear predict points in one dimension, usually ^{15}N. Here, we employ a trick that enables you to perform LP in both ^{13}C and ^{15}N, thus giving you better resolution in both dimensions. The tradeoff is that it requires slightly more time to process, gives you larger datasets, and may leave you open to some artifacts if your spectrum is not well-resolved.

Here is the script, proc3d_2xlp.s, that performs this operation:



```
#!/bin/csh
#proc3d_2xlp.s  06/29/05 Josh Kurutz
#This script processes 3D experiments using
#linear prediction in both indirect dimensions.


#FIRST STEP: process 1H (F3) dimension
xyz2pipe -in ./data/test%03d.fid -x -verb   \
| nmrPipe -fn SOL           \
| nmrPipe -fn SP -off 0.5 -end 1.0 -pow 1.0 \
| nmrPipe -fn ZF -auto      \
| nmrPipe -fn FT -verb 2        \
| nmrPipe -fn PS -p0 -50 -p1 0 -di  \
| nmrPipe -fn POLY -auto -ord 1 \
| nmrPipe -fn EXT -left -sw \
| pipe2xyz -out ./ft/scratch%03d.ft -x


#SECOND STEP: process first indirect dimension (F2)
#  using zero-filling, not linear prediction
xyz2pipe -in ./ft/hnco%03d.ft -y    \
#LINE A: the following line is important for later reprocessing
#   make sure to never use "-end 1.0"
| nmrPipe -fn SP -off 0.5 -end 0.95 -c 1.0  \
| nmrPipe -fn ZF -auto      \
| nmrPipe -fn FT  -verb 2       \
#LINE B: the following line is important for later reprocessing
| nmrPipe -fn PS -p0 -90 -p1 180 -di    \
| pipe2xyz -out ./ft/scratch%03d.ft -y -inPlace


#THIRD STEP: process third dimension (F1) using mirror-image linear prediction
xyz2pipe -in ./ft/scratch%03d.ft -z \
| nmrPipe -fn LP -ps0-0 -pred 80    \
| nmrPipe -fn SP -off 0.5 -pow 2.0 -c 0.5   \
# | nmrPipe -fn ZF -auto        \
| nmrPipe -fn FT -verb 2        \
| nmrPipe -fn PS -p0 -0 -p1 0 -di   \
| pipe2xyz -out ./ft/scratch%03d.ft -z -inPlace

#FOURTH STEP: recreate the F2 fids by inverse processing
xyz2pipe -in ./ft/scratch%03d.ft -y \
| nmrPipe -fn HT -auto      \
#LINE BB: p0 and p1 values must equal those of LINE B above
| nmrPipe -fn PS -p0 -90 -p1 180 -inv \
| nmrPipe -fn FT -verb 2 -inv \
| nmrPipe -fn ZF -auto -inv \
#LINE AA: params here must equal thos on LINE A above
| nmrPipe -fn SP -off 0.5 -end 0.95 -c 1.0 -inv \

#FIFTH STEP: linear predict F2 with mirror-image LP and process normally
| nmrPipe -fn LP -ps90-180 -pred 78 \
| nmrPipe -fn SP -off 0.5 -c 1.0    \
| nmrPipe -fn FT -verb 2        \
| nmrPipe -fn PS -p0 -90 -p1 180 -di    \
| pipe2xyz -out ./ft/final%03d.ft -y
```

Figure 31. The process3d_lp2.s script, which accomplshes 3D processing and linear prediciton in t2 and t1

The first section you should edit as you would any 2D processing script, entering your phase and apodization parameters. The other sections may be edited, but there are some caveats, discussed below.

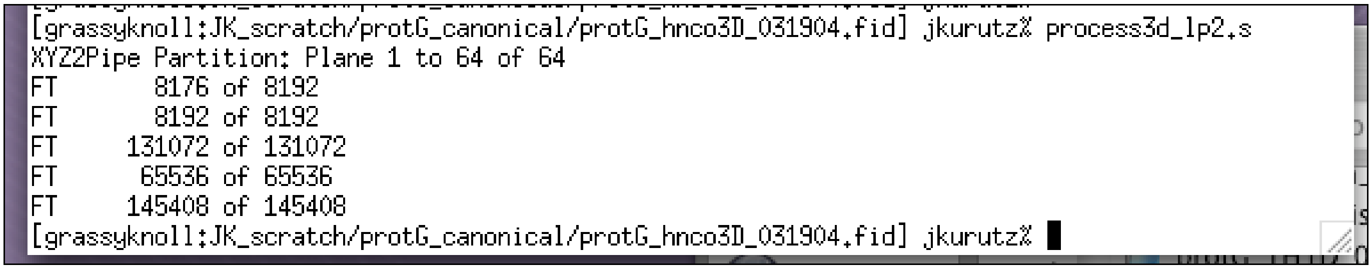
In the second section, the ^{15}N dimension is apodized, transformed, and phased. Zero filling is applied, but will be undone later.

In the third section, linear prediction is applied to the ^{13}C dimension, adding 78 points. (This spectrum was acquired with 50 complex ^{13}C increments, so this will yield 128 points across the ^{13}C axis.) The spectrum is then weighted, transformed, and phased.

In the fourth section, the ^{15}N dimension is made to revert back to its fid's, then linear prediction is applied and normal processing carried out. The "HT" line specifies a Hilbert transform, which prepares the spectrum for inverse fourier transform. The next PS, FT, ZF, and SP lines all end with a "-inv" qualifier, signifying that their operation will be the inverse of what they normally do. For instance, a normal "SP –off 0.5 –end 1.0" statement would multiply an fid by a sine-bell curve. "SP –off 0.5 –end 1.0" divides its object by a sine-bell curve. This point is important, because we discovered we cannot use a normal sine-bell curve to weight ^{15}N when using this script. Using "SP –end 1.0" multiplies the last point by 0, so "SP –end 1.0 –inv" divides the last point by zero, making everything in the spectrum infinitely large. Thus, we use "-end 0.95" instead. After undoing the ^{15}N dimension, then LP is applied, and so are the normal SP, FT, and PS.

One other item of note here is that the dimension that gets undone must be undone in exactly the same way it was constructed. For instance, "SP –off 0.45 –end 0.95 –pow 1 –c 0.5" is applied in section two. Processing will go terribly awry if in section four a SP … -inv line is applied with any parameters different from those in section two.

To execute the script, simply type "process3d_lp2.s" and go get a cup o' coffee or something. this will take awhile, but it won't involve any input from you. When it's done, your X window should look like:

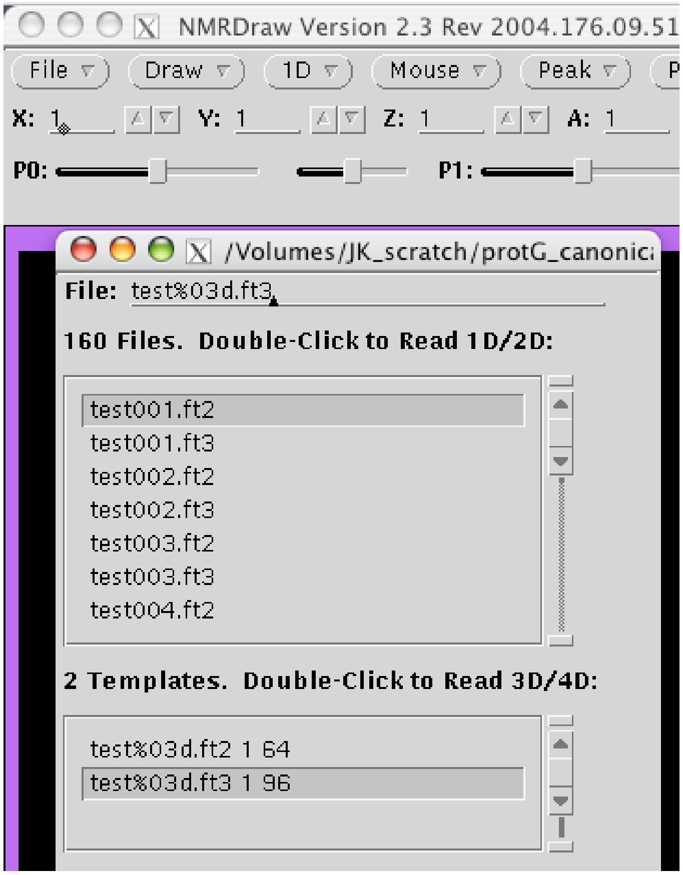


```
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% process3d_lp2.s
XYZ2Pipe Partition: Plane 1 to 64 of 64
FT       8176 of 8192
FT       8192 of 8192
FT     131072 of 131072
FT      65536 of 65536
FT     145408 of 145408
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% ▮
```

Figure 32. On-screen results of executing the proces3d_lp2.s script

E. Examine your 3D spectrum

At this point, you must be dying of curiosity to see what your 3D spectrum looks like. Spark up nmrPipe and read in the spectrum. Remember to choose the ".ft3 3D" selection:



NMRDraw Version 2.3 Rev 2004.176.09.51

/Volumes/JK_scratch/protG_canonic

File: test%03d.ft3

160 Files. Double-Click to Read 1D/2D:

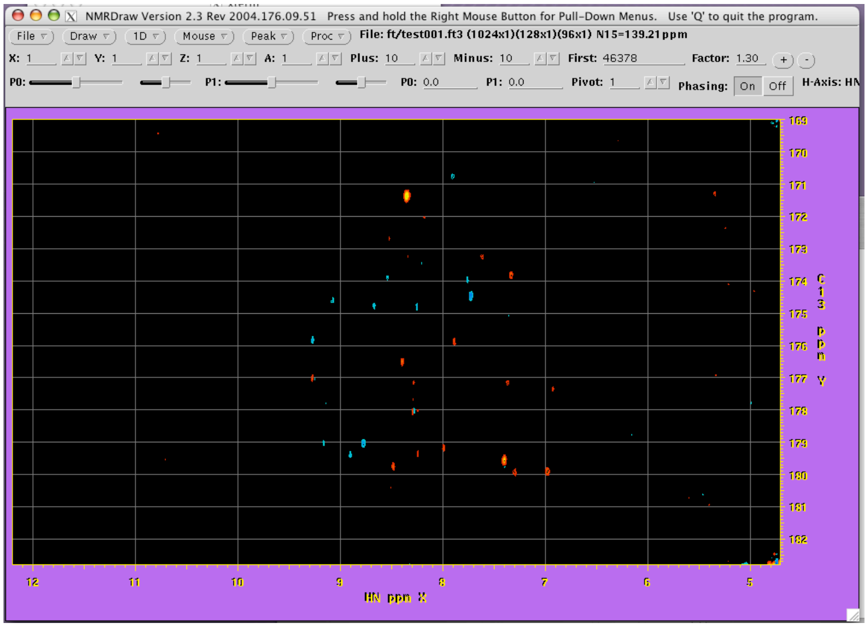
test001.ft2
test001.ft3
test002.ft2
test002.ft3
test003.ft2
test003.ft3
test004.ft2

2 Templates. Double-Click to Read 3D/4D:

test%03d.ft2 1 64
test%03d.ft3 1 96

Figure 33. nmrPipe selection of the 3D dataset

With the *.ft3 dataset highlighted, click the Read/Draw button. You may get something that looks like this:



NMRDraw Version 2.3 Rev 2004.176.09.51 Press and hold the Right Mouse Button for Pull-Down Menus. Use 'Q' to quit the program.

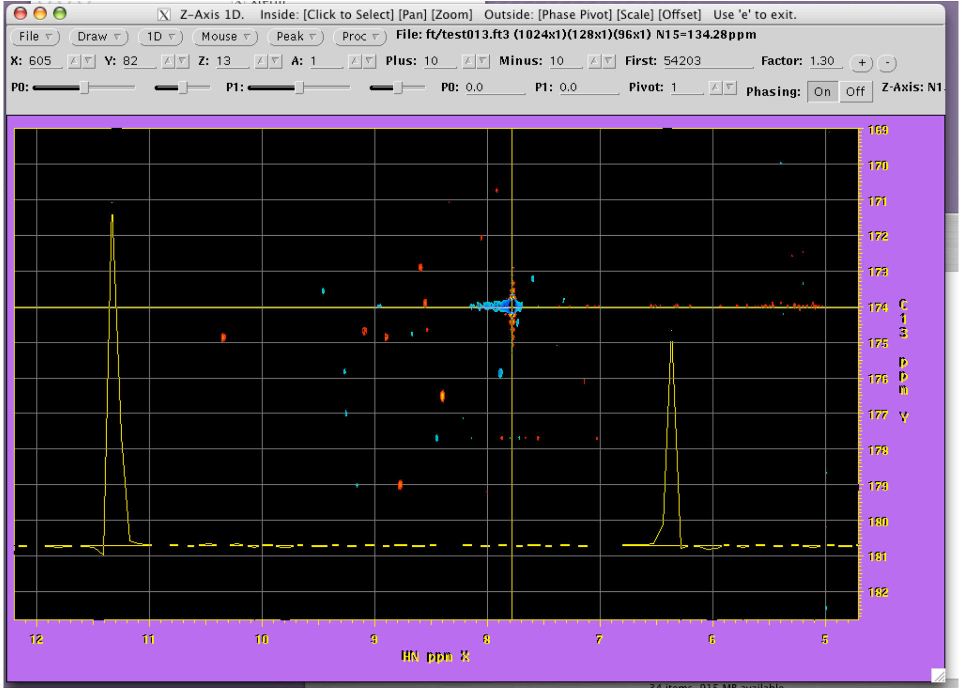
File: ft/test001.ft3 (1024x1)(128x1)(96x1) N15=139.21ppm

X: 1 Y: 1 Z: 1 A: 1 Plus: 10 Minus: 10 First: 46378 Factor: 1.30

P0: 0.0 P1: 0.0 Pivot: 1 Phasing: On Off H-Axis: HN

Figure 34. 1st plane of a 3D hnco spectrum, as seen when first opening the file

Yes, this looks funny. There are indeed both positive and negative peaks here, in this first ^{15}N plane of the 3D spectrum. Don't despair! These are just the tails of your ^{15}N peaks, which ring positive and negative a little bit. To get a better idea of your relevant vertical scale, scroll up and down the ^{15}N planes by clicking the up and down triangles next to the capital Z at the top of your window; they're just below the "mouse" and "peak" buttons. Watch how the number next to Z changes as you scroll. This number specifies the plane you are in. Scroll up until you find a plane with a much stronger signal (Figure 35). You can view the ^{15}N 1D trace for this peak by putting your mouse into Z-axis mode and placing the crosshairs on the peak. Do this by left-clicking "Mouse," then selecting "Z) 1D Z-axis."



Z-Axis 1D. Inside: [Click to Select] [Pan] [Zoom] Outside: [Phase Pivot] [Scale] [Offset] Use 'e' to exit.

File: ft/test013.ft3 (1024x1)(128x1)(96x1) N15=134.28ppm

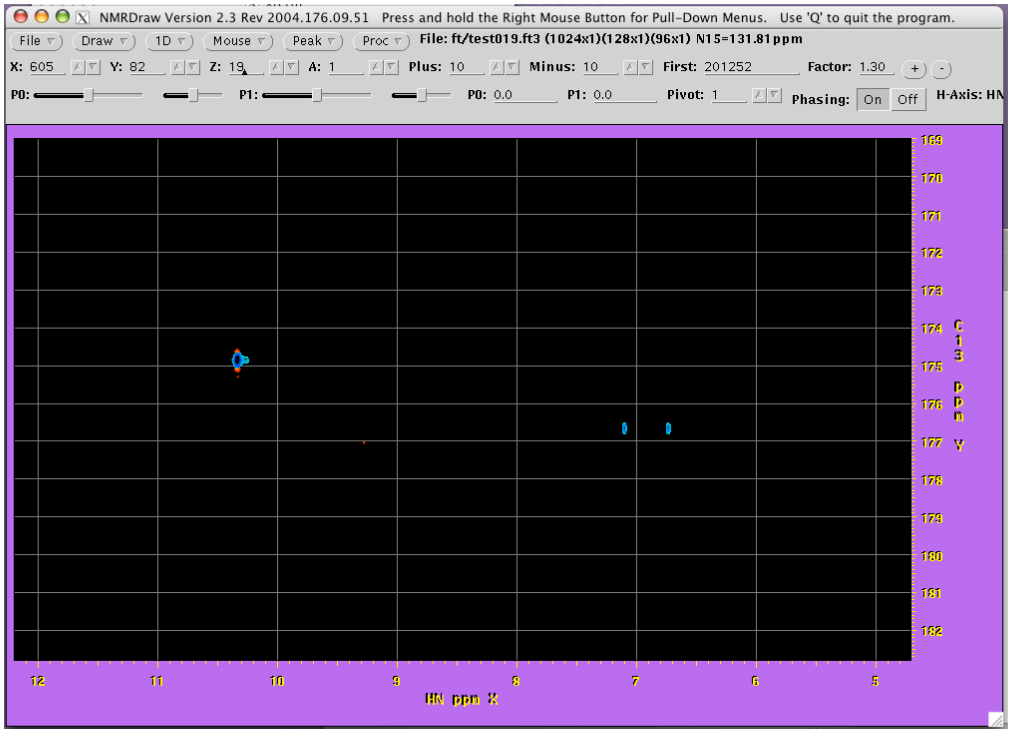
X: 605 Y: 82 Z: 13 A: 1 Plus: 10 Minus: 10 First: 54203 Factor: 1.30

P0: 0.0 P1: 0.0 Pivot: 1 Phasing: On Off Z-Axis: N1

Figure 35. Processed 3D spectrum (hnco) exhibiting a strong, relevant peak and several weaker, irrelevant peaks

The yellow spectrum you observe is the ^{15}N trace of the 3D spectrum at that ^{1}H/^{13}C frequency combination. In (Figure 35), the peak shown in the 2D ^{1}H/^{13}C spectrum show two strong ^{15}N frequencies, indicating overlap in the ^{1}H/^{13}C spectrum that is resolved by going to the 3D spectrum. Note that the signal is actually quite strong and consistently positive when viewed along the ^{15}N axis.

You can adjust the vertical scale by clicking the "+" button in the upper right corner a few times and re-displaying the spectrum. Scroll around, and you'll see that you can clearly distinguish the real signals from the ^{15}N tail peaks:



NMRDraw Version 2.3 Rev 2004.176.09.51 Press and hold the Right Mouse Button for Pull-Down Menus. Use 'Q' to quit the program.

File: ft/test019.ft3 (1024x1)(128x1)(96x1) N15=131.81ppm

X: 605 Y: 82 Z: 19 A: 1 Plus: 10 Minus: 10 First: 201252 Factor: 1.30

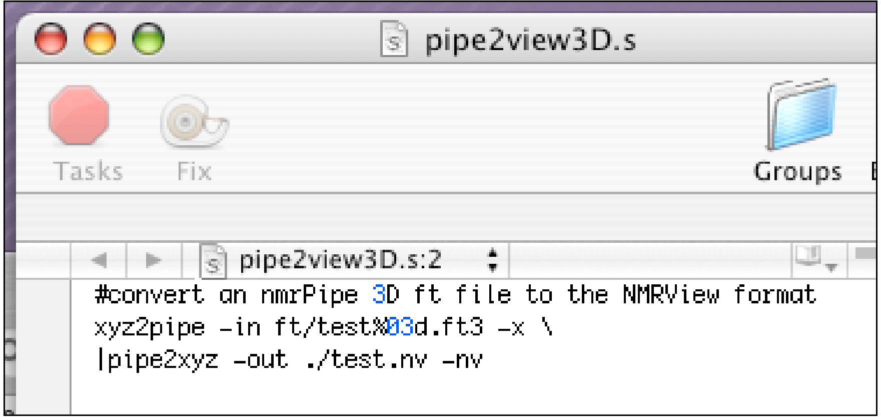
P0: P1: P0: 0.0 P1: 0.0 Pivot: 1 Phasing: On Off H-Axis: HN

Figure 36. One 15N plane of the hnco depicted after adjusting the vertical scale

If this all looks OK, then go ahead and convert the file to nmrview format and go on your merry way.

F. Convert the 3D nmrPipe spectrum to nmrView format.

This step is trivial. All you need is a new script for the conversion:



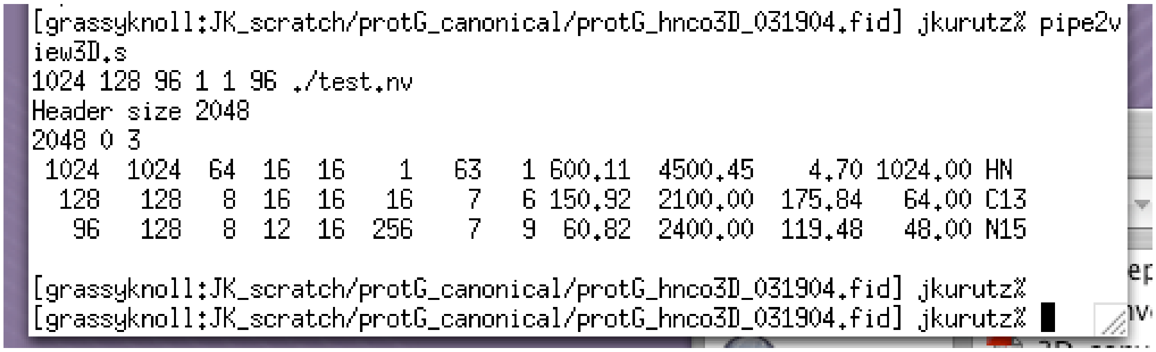
pipe2view3D.s

pipe2view3D.s:2

```
#convert an nmrPipe 3D ft file to the NMRView format
xyz2pipe -in ft/test%03d.ft3 -x \
|pipe2xyz -out ./test.nv -nv
```

Figure 37. The script pipe2view3D.s, which converts an nmrPipe 3D dataset into nmrView format

Go in to your X window spectrum directory and type “pipe2view3D.s”. This will effect the conversion in 20-90 seconds. At the end, you should see something like this:



```
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% pipe2v
iew3D.s
1024 128 96 1 1 96 ./test.nv
Header size 2048
2048 0 3
 1024  1024  64  16  16    1   63   1 600.11  4500.45    4.70 1024.00 HN
  128   128   8  16  16   16    7   6 150.92  2100.00  175.84   64.00 C13
   96   128   8  12  16  256    7   9  60.82  2400.00  119.48   48.00 N15

[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz%
[grassyknoll:JK_scratch/protG_canonical/protG_hnco3D_031904.fid] jkurutz% █
```

Figure 38. Your X window after executing the pipe2view3D.s conversion script.

G. You are done, done done!

4) Appendix 1: Phasing

A. 1D phasing

Adjusting the phase on your spectrum is a skill best learned at the spectrometer console. The practical goal of phasing is very simple: to make sure that the baselines on either side of your peak or group of peaks line up with one another. Here are some simple examples to illustrate the principle.

(1D simple mol. – not phsed)

(1D simple mol. – phased)

(1D protei spec – not phased)

(1D protein spec – phased)

In VNMR, you phase peaks by clicking "phase," selecting the right-most region of the spectrum, click/drag your left button up and down to adjust the phase to the baseline(s) look good, then select the opposite end of the spectrum and adjust its phase. In this manner, you go back and forth a few times until thee whole spectrum looks good. Note that this adjusts two parameters, "rp" and "lp," which stand for "right phase" and "left phase."

In NMRPipe, you achieve the same ends in a similar manner. (See section 2)F on page 8 of this manual.) Once you have a 1D spectrum on display, place a "pivot point" under your rightmost peak by left-clicking in the purple border between your black spectrum display and the edge of your window:

Then adjust the "right phase" by sliding the rough and fine "p0" slider bars, found in the upper left of the nmrPipe window. The goal here is *only* to phase the peak or peak region at the pivot point, *not* the whole spectrum. If the rest of the spectrum looks properly phased after adjustment of p0 only, that's fine, but it's not required. This is how the spectrum shown above looks after proper adjustment of p0:

Next, adjust the “left phase” by sliding the rough and fine “p1” slider bars. The phasing at the pivot point will not change when you adjust p1 – that’s the purpose of the pivot point. Adjust p1 so that the leftmost peaks. This is how the above p0-phased spectrum looks after proper adjustment with p1:

Write down your p0 and p1 values!

B. 1D Phasing theory

You need to make phase adjustments in your directly-detected dimension because of the method by which the spectrometer acquires your data. Signal from the probe is split into two channels that are exactly 90° out of phase. Without this detection method, signals would appear on both sides of the spectrum’s center because there would be no way to distinguish positive from negative frequencies. When these data are combined when processing, what we see is a blend of signal from the two channels. We cannot, however, say that one detection channel corresponds to the “real” part of the spectrum, which gives “absorptive lineshapes,” and the other corresponds to the “imaginary” part, which gives rise to “dispersive” peaks. All we can know (with our conventional hardware – this issue is irrelevant in the new Varian NMR “Direct Drive” system, which debuted at the 2005 ENC) is that the real and imaginary parts are 90° out of phase, and our two detected channels are also 90° out of phase. By adjusting our phasing parameters, we get the two to agree so what all of what we see is the “real” component, and all of what we don’t (but is still stored) is the imaginary component.

(real/absorptive vs imaginary/dispersive)

(Discussion of the relationship between phasing and the delay between the last pulse and acquisition.)

C. Phasing indirectly-detected dimensions

It is important to have phase information in indirectly-detected dimensions, but it is impossible to split the signal into two channels since it is entirely encoded in the single directly detected signal. Instead, multidimensional spectra are acquired so that phases of certain pulses have one value when the “real” component is desired, and a different value when the “imaginary” component is desired. On Varian instruments, this is accomplished by setting “phase=1,2.” Thus, for 128 desired points in an indirect dimension, one acquires 128 fid’s with “real” encoding, and 128 fid’s with “imaginary” encoding.

One benefit of this scheme is that the phasing parameters for indirectly-detected dimensions are entirely predictable, so they need no manual adjustment. Shown below are phasing parameters for many common experiments. This table also includes other parameters, such as first point multipliers and

experiment	2nd dim p0	2nd dim p1	2nd dim fp mult	2nd dim LP	3rd dim p0	3rd dim p1	3rd dim fp mult	3rd dim LP
Homonuclear								
DQ-COSY								
TOCSY								
NOESY								
ROESY								

CT-COSY								
Heteronuclear 2D								
HSQC (^{15}N or ^{13}C)	-90	180	1.0	90-180				
Heteronuclear 3D (f1180='n')								
HCNO	0	0	-c 0.5	ps0-0	0	0	-c 0.5	-ps0-0
CCONH	0	0	-c 0.5	ps0-0	0	0	-c 0.5	-ps0-0
HNCACB (FT –neg required)	0	0	-c 0.5	-ps0-0	0	0	-c 0.5	-ps0-0
CBCACONH	0	0	-c 0.5	-ps0-0	0	0	-c 0.5	-ps0-0
HCACOCANH								
HBHACONH								
HCCONH								
NOESY-N-HSQC								
NOESY-C-HSQC								

D. Relationship between phasing and first point multiplication

NMRPipe apodization functions frequently contain arguments that look like "-c 0.5." The "-c" option specifies that the first point of an fid will be multiplied by the number following it, in this case, 0.5. This is handy in cases where the first point is artificially large, but it can result in subtle problems, such as T1 ridges, if it is misapplied.

The rule is to apply "-c 0.5" to fids whose spectra will be phased with p1 approximately 0. If p1 is approximately 180, then use "-c 1.0."

E. Relationship between linear prediction and phasing

Linear prediction can clearly elevate the quality of your spectrum by extending your fid instead of just adding zeroes to it. However, LP involves manufacturing data to resemble existing data, so one must be very careful about its application.

The most general means of LP involves analysis of the FID, determining its dominant frequencies, and creating data at the end of the fid that matches the experimentally acquired fid. This technique is best applied only to fids representing a few different frequencies. To simply linear predict, say, 78 points to an fid, simply include the line: "LP –pred 78". Note that this ADDS 78 points. If you start with 50 points and LP 78 more, you'll get 128. Using "LP –pred 128" will net you 178 total points.

An alternative means of linear prediction involves creating a mirror image of your fid. Here, new points are less “manufactured” than those derived with standard linear prediction because they are simply the actual data, just reflected. The reflection is done around the first point in the FID and extends backward in time. It is done backward instead of forward because all frequencies will be at a maximum or minimum at the first point, whereas they are no longer aligned at the final point.

If the fid results in a spectrum properly phased with p0 = -90 and p1 = 180, you invoke mirror-image linear prediction with “LP –pred90-180”. With these FID’s, the first point is a maximum, and the FID can simply be reflected about the zero time point.

If the fid’s spectrum requires phasing of p0 = 0 and p1 = 0, use “LP –pred0-0.” In this case, the first point of the fid has zero intensity, and the mirror image must be multiplied by –1 in order to achieve a rational fid.

Even in mirror image linear prediction

9Discussion f the relat. bet. LP and phasing.)

5) Appendix 2: Installing NMRPipe on a Mac

The instructions supplied in the README and INSTALL files associated with NMRPipe are not quite complete, and they are not as convenient as they could be for the mac environment. Here are detailed instructions you can follow to install NMRPipe on your mac, including modifications of UNIX files you might not even know existed.

A. Prepare your terminal windows to emulate “t c-shell” windows.

This will enable certain commands that are otherwise misunderstood. To do this, you must modify the settings of your account using a little-known Apple system program called “NetInfo Manager.” Open NetInfo Manager; it’s found under “/Applications/Utilities:”

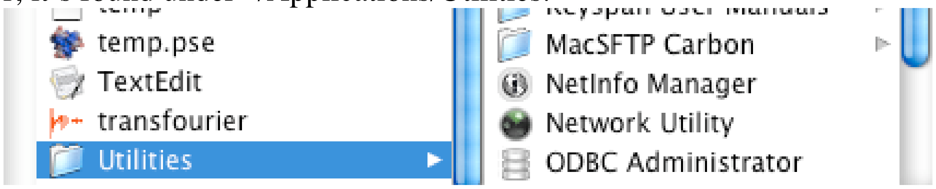
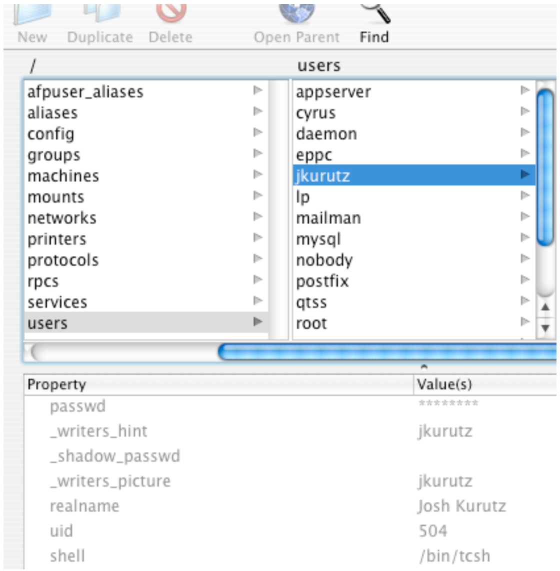


Figure 39. Where to find "NetInfo Manager"

Once your program is open, select /…/Users…/(your username). In the lower pane, scroll down until you see a line that reads “shell” on the left. Your need this line to read “tcsh” on the right. Edit if necessary. Don’t if not necessary.



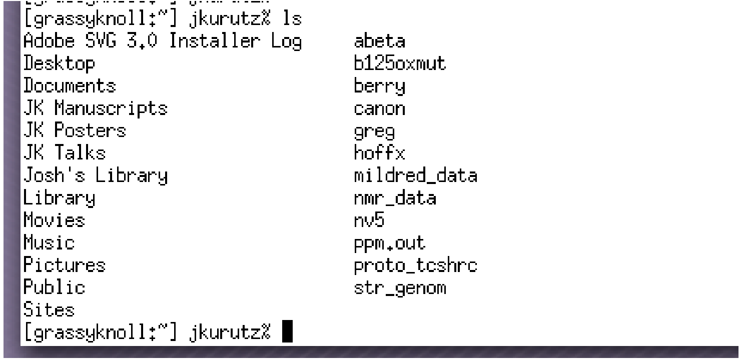
Property	Value(s)
passwd	********
_writers_hint	jkurutz
_shadow_passwd	
_writers_picture	jkurutz
realname	Josh Kurutz
uid	504
shell	/bin/tcsh

Figure 40. Mac user info, including your shell type (needs to be "tcsh".

Once you've modified your shell type, close the NetInfo Manager program.

B. Edit your "tcshrc" file.

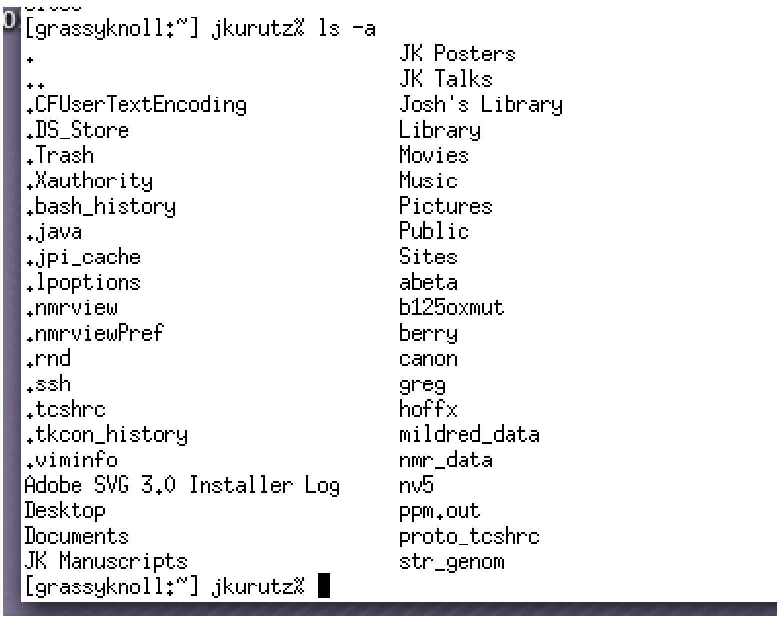
If you are new to UNIX, you might find it surprising that your working environment contains a number of files that are invisible to you under normal circumstances. To see what I mean, you should start up X11 and get a command line prompt. To see the files in your home directory, type "ls" or "list." Here's what mine look like (if you don't mind seeing my personal junk):



```
[grassyknoll:~] jkurutz% ls
Adobe SVG 3.0 Installer Log     abeta
Desktop                         b125oxmut
Documents                       berry
JK Manuscripts                  canon
JK Posters                      greg
JK Talks                        hoffx
Josh's Library                  mildred_data
Library                         nmr_data
Movies                          nv5
Music                           ppm.out
Pictures                        proto_tcshrc
Public                          str_genom
Sites
[grassyknoll:~] jkurutz%
```

Figure 41. Results of the "ls" command in a home directory.

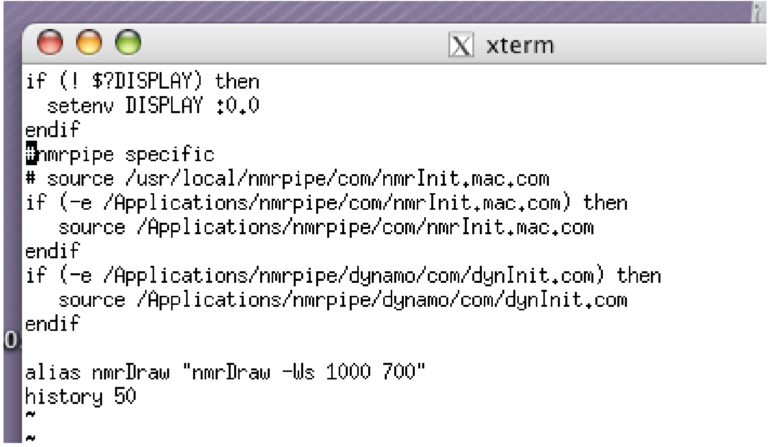
If you type "ls –a" ("-a" for "all"), you see a number of hidden files, which begin with ".".



```
[grassyknoll:~] jkurutz% ls -a
.                               JK Posters
..                              JK Talks
.CFUserTextEncoding             Josh's Library
.DS_Store                       Library
.Trash                          Movies
.Xauthority                     Music
.bash_history                   Pictures
.java                           Public
.jpi_cache                      Sites
.lpoptions                      abeta
.nmrview                        b125oxmut
.nmrviewPref                    berry
.rnd                            canon
.ssh                            greg
.tcshrc                         hoffx
.tkcon_history                  mildred_data
.viminfo                        nmr_data
Adobe SVG 3.0 Installer Log     nv5
Desktop                         ppm.out
Documents                       proto_tcshrc
JK Manuscripts                  str_genom
[grassyknoll:~] jkurutz%
```

Figure 42. Invoking "ls -a" reveals hidden UNIX files

You need to edit the file “.tcshrc” to set your shell preferences correctly. If this is the first time you ever did anything with your UNIX environment, you might not een have this file. Not to worry, you can create it here. If you can use the editor program “vi,” it’ll be quick to do so here and now; type “vi .tcshrc.” If you’re not vi-savvy and you have a .tcshrc file, you can copy the .tcshrc file to a temporary file that TextEdit can see; the normal Apple Finder environment and normal applications don’t see files starting with “.”. In this case, type: “cp .tcshrc temp.txt,” then open temp.txt with TextEdit. If you don’t have a .tcshrc file, then create a new plain text (RTF rich text format will NOT work) file called “temp.txt” with TextEdit. I any of the above cases, the contents of your .tcshrc file should look like mine



```
if (! $?DISPLAY) then
  setenv DISPLAY :0.0
endif
#nmrpipe specific
# source /usr/local/nmrpipe/com/nmrInit.mac.com
if (-e /Applications/nmrpipe/com/nmrInit.mac.com) then
   source /Applications/nmrpipe/com/nmrInit.mac.com
endif
if (-e /Applications/nmrpipe/dynamo/com/dynInit.com) then
   source /Applications/nmrpipe/dynamo/com/dynInit.com
endif

alias nmrDraw "nmrDraw -Ws 1000 700"
history 50
~
~
```

Figure 43. Contents of the .tcshrc file.

If you edited your file in TextEdit, save it as something like “temp.txt.” Then, back in your UNIX window, give it the name “.tcshrc” by typing “cp temp.txt .tcshrc” on your command line. Note that is essential that your .tcshrc file reside in your home directory.

If you have to modify your user settings and/or your .tcshrc file, you should exit X11 now. To be sure, you may want to log out/in or reboot just to make certain your shell settings have stuck.

C. Install NMRPipe

The instructions for installing NMRPipe are in the file called INSTALL. These tell you to install the program in a directory called /usr. I recommend installing it in /Applications instead. Your Finder environment cannot see anything in /usr, so if you ever want to access your NMRPipe files outside of X11, you shouldn’t put them in /usr. Here are detailed instructions for installing NMRPipe in /Applications.

i) Start X11

ii) Either in X11 or in Finder, create a directory called “nmrpipe” in /Applications.

iii) Put all the files you downloaded (or were supplied): nmrPipe.mac.tar.Z, binval.com and install.com, pdbH.tar.Z, into your new nmrpipe directory. You may wish to add pdbH,tar.Z and dyn.tar.Z to run the program DYNAMO (which isn’t necessary for running NMRPipe).

iv) Change your directory: “cd /Applications/nmrPipe”

v) To be sure you have execution permissions on all your files, type “chmod +x *”

vi) Start the installation script by typing “./install.com /Applications/nmrpipe”

vii) That should be it. If everything went OK, you should have a file in the directory /Applications/nmrpipe/com called “nmrInit.mac.com.” Its contents should make lots of references to your new directory, /Applications/nmrpipe and its subdirectories. If things did not work as planned, please consult the INSTALL file to see if everything was in order; e.g., check to make sure you started with all the right files to start with.

viii) You may need to log out and back in if you just set your default shell to “tcshrc” from something like “bash.” Restarting of logging out/in is the first thing to try if you type “nmrDraw” and nothing productive happens.

6) Appendix: Common 3D experiments’ nucleus/axis correspondence table

OK, so you should be able to figure this out by reading Cavanagh, and, moreover, you should’ve written this information in your lab notebook when you were at the console. But now it’s late and night, your committee meeting is tomorrow, you’re just now processing your spectra, and you’ve realized that you don’t know if the axes of your HBHACONH experiment are H/C/N, H/H/C, or H/H/N or what. Here’s the table that will guide you.

experiment	**F1**	**F2**	**F3**
Homonuclear			
DQ-COSY	^{1}H	^{1}H	-
TOCSY	^{1}H	^{1}H	-
NOESY	^{1}H	^{1}H	-
ROESY	^{1}H	^{1}H	-
CT-COSY	^{1}H	^{1}H	-
Heteronuclear 2D			
HSQC (^{15}N or ^{13}C)	^{15}N or ^{13}C	^{1}H	-
Heteronuclear 3D			
HNCO	^{15}N	^{13}C’	^{1}H
HNCACO	^{15}N	^{13}C’	^{1}H
HCACOCANH	^{15}N	^{13}C’	^{1}H
HNCOCO	^{15}N	^{13}C’	^{1}H
HNCA	^{15}N	$^{13}C_{\alpha}$	^{1}H
HNCOCA	^{15}N	$^{13}C_{\alpha}$	^{1}H
CCONH	^{15}N	$^{13}C_{aliph}$	^{1}H
HNCACB	^{15}N	$^{13}C_{aliph}$	^{1}H
CBCACONH	^{15}N	$^{13}C_{aliph}$	^{1}H
HNHA	^{15}N	^{1}H	^{1}H
HBHACONH	^{15}N	^{1}H	^{1}H
HCCONH	^{15}N	^{1}H	^{1}H
HCCH-COSY	^{13}C	^{1}H	^{1}H
HCCH-TOCSY	^{13}C	^{1}H	^{1}H
NOESY-N-HSQC	^{15}N	^{1}H	^{1}H
NOESY-C-HSQC	^{13}C	^{1}H	^{1}H
NOESY-CN-HSQC	^{13}C	^{1}H	^{1}H

7) Appendix: Troubleshooting

Even with your best efforts, you may end up with spectra that don't look quite right. Here is a guide to some of the more common problems.

- i) F1 in your HSQC doesn't phase correctly no matter what you do and it still seems close to being phased.
  - (1) This probably arises from not setting your F1 acquisition more to "Rance-Kay" instead of "States." Check out your fid.com script and, if necessary, change it and reprocess your spectrum.
- ii) None of the phasing changes I enter into my processing script makes any difference in the resulting spectrum.
  - (1) You may not be operating on the fid you want. Check your script to see if it's reading in the file "./test…fid," not some file with a full pathname like "/Users/jkurutz/Data/protG/protG_Nhsqc_032404/fid.com."
- iii) My HNCACB looks like its ^{13}C dimension is inverted. The few negative peaks for ser and thr are on top, and the bulk of the negative peaks are on the bottom.
  - (1) For reasons unknown to this author, the HNCACB spectrum requires that the spectrum be processed with a "-neg" argument for the FT function: "nmrPipe -fn FT -neg -verb 2."
- iv) I just processed my HNCO, and the ^{1}H and ^{13}C dimensions look fine, but all my ^{15}N resonances are in the middle of their spectrum, but there's ^{15}N noise throughout the spectrum.
  - (1) Your HNCO is toast. You must be running a relatively new HNCO, which looks for the parameter "pra," which invokes "projection reconstruction acquisition." This fancy parameter was not included in the HCNO procpar file when it was distributed, so it needed to be created by typing "create('pra','real')." This enables acquisition without error messages, but it sets pra=''. When the HNCO sees this, it simply fails to increment the ^{15}N dimension. Fix the problem by retaking the HCNO after setting pra='0'.