Introduction to nmrPipe

F.Delaglio, S. Grzesiek, G. Vuister, G. Zhu, J. Pfiefer, and A. Bax.
J. Biomol. NMR, 6 (1995) 277-293

Web site: spin.niddk.nih.gov/bax/software/NMRPipe

Requirements: UNIX operating system (SGI, SUN, Linux, MAC OSX, etc.)

Basic description:
From the manual- "The nmrPipe program is the central part of a system of tools for multi-dimensional spectral processing using UNIX pipes."

The nmrpipe data processing program consists of a number of functions (e.g. FT) that are linked together via unix pipes.
Unix pipe example: ls | more
The vertical line is a 'pipe' ; the output of the list files command (ls) becomes the input of the screen paging command (more).
The functions are usually strung together in a script or macro and executed as a single unix command.

A second program called 'nmrDraw' is used to visualize the processed nmr data. This program provides basic functions such as drawing contours, vertical or horizontal traces, peak picking, integration, etc. A number of functions are available in menus (and mapped to the keyboard) to allow for quick manipulation of the data for analysis.
One can also use other programs to visualize the data (e.g. nmrview).

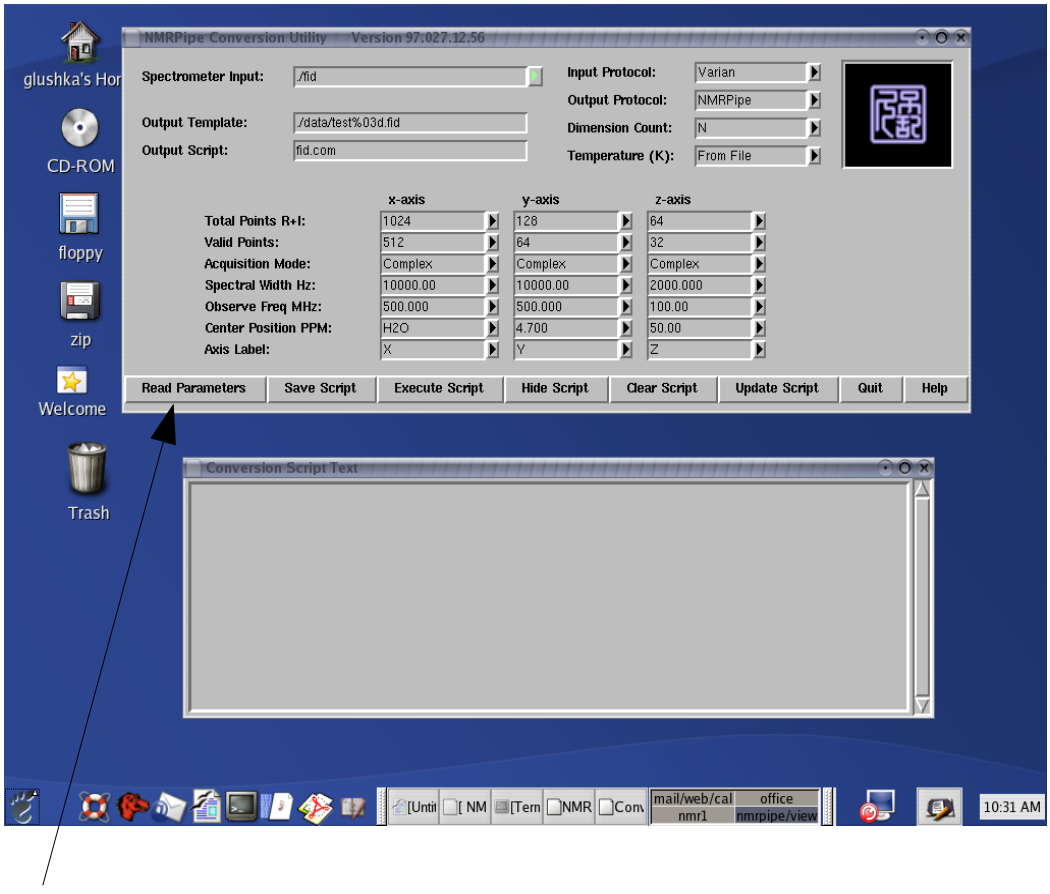
There are also other programs such as 'var2pipe' which are used to convert data into nmrpipe format.

The program manual pages are filled with information and examples. (man nmrPipe)
Brief descriptions of specific functions are accessible via nmrPipe -
'nmrPipe -help ' will list most functions.
'nmrPipe -fn GM -help' will give description of the GM function for example.

Converting spectrometer data into nmrPipe format:

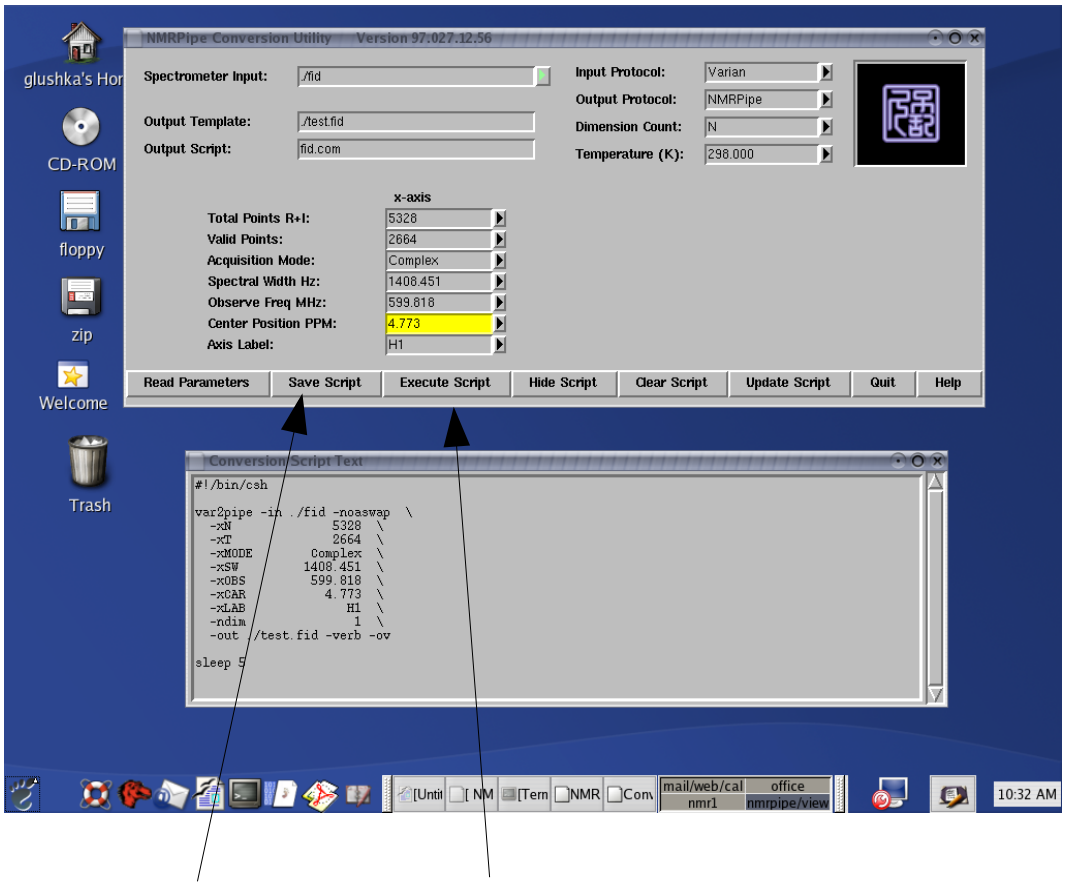
First convert data from spectrometer format (Varian, Bruker, etc.) to nmrpipe format using conversion scripts. This is most easily done using a built-in program called 'varian' for Varian data (or 'bruker' for Bruker data). Our first example will be a 1d proton spectrum using a Varian dataset called 'sample1d.fid'.
-open a terminal window and copy this dataset into your home directory.

1) cd into sample1d.fid, then type 'varian' – this starts a tcl/tk script and brings up these two windows:



2) Click on the 'Read Parameters' button. The script will read the parameter file ('procpar') from inside the sample.fid directory and update the parameters to give:

Note that the template now shows only a single dimension file, with the appropriate paramters. At this stage, one can update the center position PPM value if known – this can also be done later.

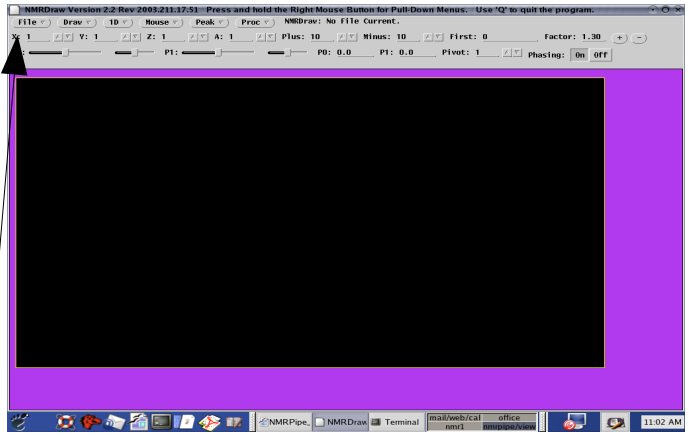


3) Click on 'Save Script', and the 'Execute Script'.
These actions have created an executable unix shell script called 'fid.com' which is displayed in the lower window, and executed it. The script shows that a program called 'var2pipe' is run with a variety of arguments; the input file is 'fid' , the generic name for the raw varian binary data, and the output is 'test.fid', which is now in nmrpipe format.
Since we started the 'varian' program inside the sample1d.fid directory, all these files remain inside this directory.
One can also keep libraries of these scripts, and edit them with a text editor. You can also run them from different directories as long as you keep track of the input and output names.

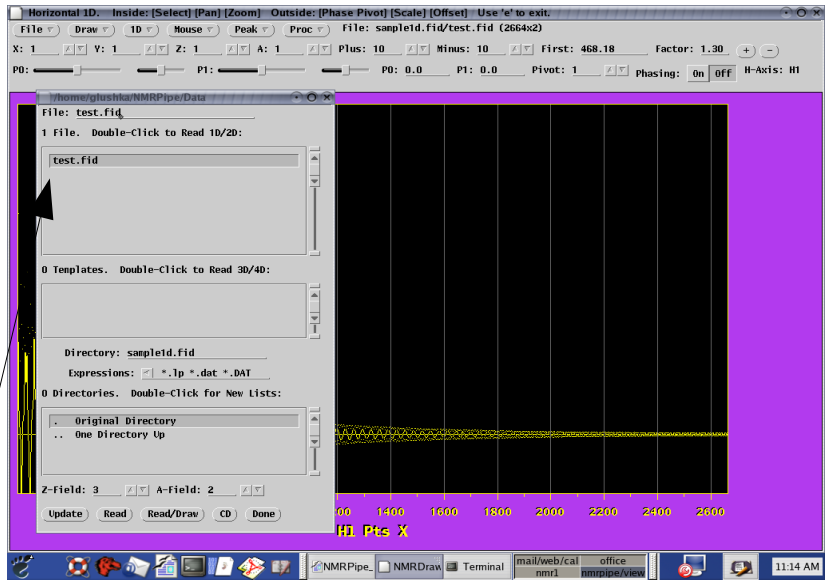
Processing 1D data using nmrDraw.

Normally we will processing data using scripts (see later) but for a quick look at a 1D you can use menu functions found in nmrDraw.

From the terminal window, type 'nmrDraw &' ; you should get this screen.



1) Right mouse button drag on 'File' menu and choose 'Select file', and right click.
   -a second window showing directories and files appears.
2) Left double click on 'sample1d.fid' listed in bottom panel – the top panel should then list the 'test.fid' file.



3) Left double click on test.fid – this will load the fid into the display:

4) click on 'Done' to remove directory window.

5) Right click the 'Proc' menu, and then right click on 'Auto-process 1D' – this will perform a fourier transform. The spectrum that appears is only available in memory and is not stored.
6) Turn 'Phasing' On, and use the P0 and P1 sliders to adjust the phase of the spectrum.
   -Note at the top of the window in the window frame, are listed the mouse functions. e.g. put cursor outside of the spectrum display into the purple panel on the right. Hold down the middle button and slide to adjust the vertical scale.
   -expand the 1D by selecting '1D Zoom' in the 'Mouse' menu.
   NOTE: some of these functions do not work – if you lose the spectrum, a left click on the 'File' menu button will re-read the fid. You can then reprocess it.

7) You can experiment with processing options by using the 'NMRPipe Commands' selection in the 'Proc' menu. This opens a small window where you can type in an explicit function (e.g. ft) if you know them, or choose from a short list of common functions. Redraw the original fid by left clicking on the 'File' menu button, then try some functions;
   - type ft or choose 'Fourier Transform', phase spectrum (first turn phasing On)
   -refresh fid, then try 'Cosine Bell' , then ft, then phase.
   -refresh fid, then apply a decaying exponential function by typing 'em -lb 20' – notice how the fid changes shape. Follow with ft and phasing to see the resulting spectrum. How is it different than with the cosine bell? (recall mestre-c tutorial)

Other functions to try in the 'NMRPipe Commands' command line:

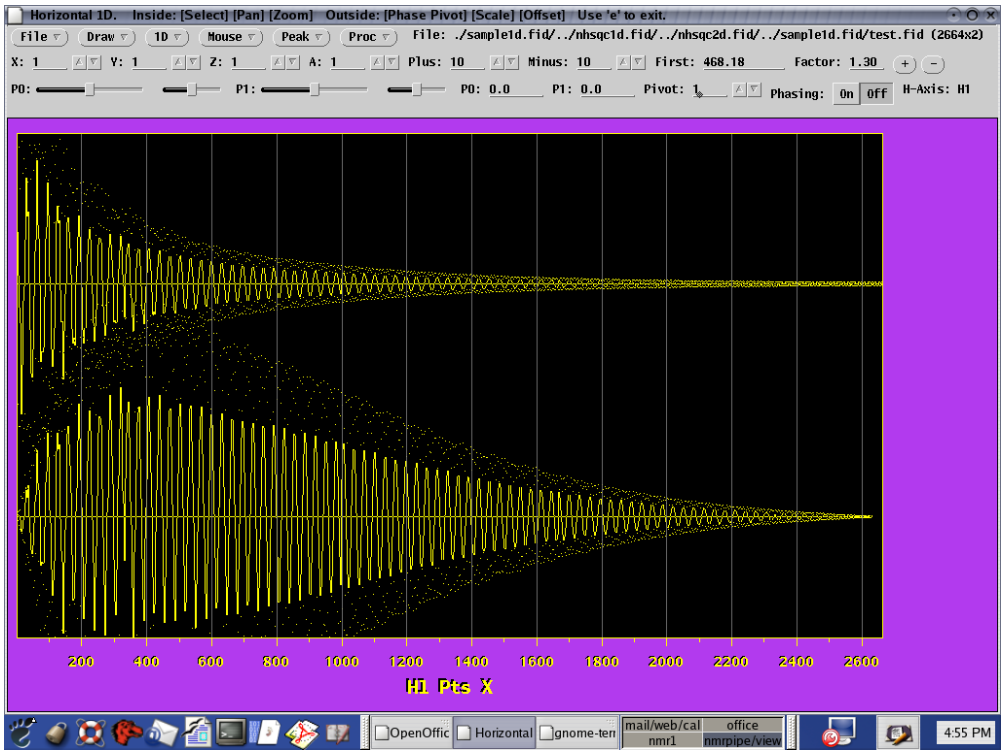
i)compare window functions (examine the manual pages e.g. 'nmrPipe -fn em -help')

- exponential line broadening: em -lb 1 ; em -lb 10; 30 etc.
- sinebell: sb (default is unshifted, i.e. maximum is halfway along the fid); sb -off 0.1 (offset by 18 deg); sb -off 0.5 (offset by 90 degrees, equivalent to cosine)
- lorentz to gauss: this function combines the exponential and gaussian and results in a gaussian lineshape that can be used for resolution enhancement. It first multiplies the fid by an inverse exponential to remove the natural decay (try 'em lb -0.7' to see this effect), then multiplies the fid by a gaussian. There is also an option to shift the maximum of the gaussian.
  gm -g1 0.8 -g2 0.8 -g3 0.0 [-g1 exponential value -g2 gaussian value -g3 shift]

ii) zero fill the data: One can add zeros to the end of the fid so that there are more points in the final fourier transformed spectrum. This will smooth the data and increase the resolution to some degree. BEFORE executing an 'ft' command, try 'zf -auto' which rounds up the number of points to the nearest power of 2, or 'zf -zf 2' which doubles the size, or 'zf -size 16384' which zero fills up to 16384 points. Zoom in to a region of fine structure to examine the effect.

iii) Linear prediction: Instead of adding zeros, one can calculate additional points based on a mathematical analysis of the existing fid. 'lp -fb' uses default values and doubles the size of the fid. This will result in real increase in digital resolution. It is used mostly in multidimensional processing. See help for other options.

iv) Baseline correction: 'POLY -auto' automatically generates a polynomial baseline correction. See help for the many other options.



Top: starting fid ; Bottom: after multiplying with an unshifted sinebell ('sp')

Processing with macros.

This is the preferred method of processing data, since it creates a record of all steps involved and the output is a file of the spectrum.
Like the conversion scripts, macros are shell scripts that are created with a text editor, and look like this:

```
#!/bin/csh
nmrPipe -in test.fid                \
| nmrPipe -fn EM -lb 1.0             \
| nmrPipe -fn FT                      \
| nmrPipe -out test.ft2 -ov -verb
```

line 1. Invokes the unix C shell,
line 2. Reads in the test.fid (complete directory paths are necessary; here the macro is assumed to be in the same directory as test.fid)
line 3. Executes a line-broadening exponential function of 1 Hz
line 4. Executes fourier transform
line 5. Pipes the output to a file called test.ft2, which will reside in the current directory.

Note that all lines after 3 start with a unix pipe. Note also the next line symbol '\', to make this a single line command. It is important to have no space after the '\'.

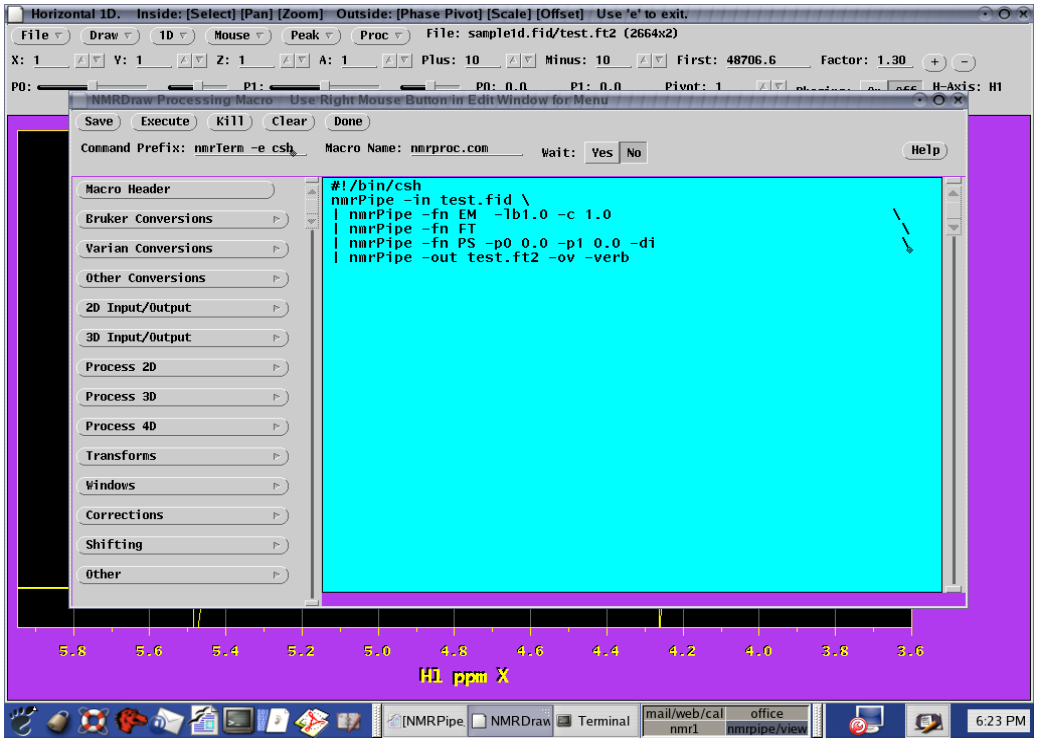
Creating Macros.
The easiest way to create these scripts is to copy an existing one and edit it with a text editor. A couple of macros are available for you to copy and edit.

The nmrDraw program also has an editing tool to aid in the construction of macros. This can be started by selecting 'Macro Edit' in the 'File' menu. It opens an editor and menus that create lines of text for various functions. It is designed for 2D and 3D processing, but you can re-create the above script the following way:

1) select 'macro edit' from the 'File' menu (right mouse)
2) click on Macro Header, then delete everything except the first line; make sure the cursor ends up at the far left of the window.
3) select 'read FID' from the '2D Input/Output' menu
4) select 'Exponential' from the 'Windows' menu
5) select 'Fourier Transform' from the 'Transforms' menu
6) select 'Phase correction' from the 'Transforms' menu
   -edit the 0.0 following -lb to 1.0; ignore the -c argument
7) select 'Write FT2' from the '2D Input/Output' menu

-now edit the EM line (3rd line) to change the -lb number to 1.0

7) Click on the Save button – this will save the macro as 'nmrproc.com' in the directory where nmrDraw was started (probably your home directory). 'Done' will remove this window.



8) Try using this script by copying it into your sample1d.fid directory.
   - cp nmrproc.com sample1d.fid
   -then cd sample1d.fid ; and then type nmrproc.com (it should be executable)
   -this will create the file test.ft2
9) Now 'select file' from the 'File' menu (right button) in nmrDraw, and double click on test.ft2.
   -the processed fid will now be displayed. If the spectrum is phased correctly (from your previous manipulations) make a note of the P0 and P1 values. If not, then try to phase it and make a note of those values.
10) Use a text editor to change the values of P0 and P1 in the 'nmrproc.com' processing macro. If you re-run the macro it will now correctly phase the spectrum. Since it now has 'built-in' phases, the displayed spectrum will appear phased only if the menu Phasing is either off, or on with P0 and P1=0.

Processing 2D data.

NmrPipe is really designed for multidimensional processing. An example of a two-dimensional dataset is available with data conversion and data processing scripts located inside the data directory.

Dataset directory: nhsqc2d.fid
-this is a NH HSQC dataset of an N15-labeled protein in 90%H2O/10% D2O, collected on a 600Mhz spectrometer. Proton dimension is direct t2 dimension, N-15 is in indirect t1 dimension.

Data conversion script: fid.com

Data processing macro: nmrproc.com

```
#!/bin/csh

nmrPipe  -in test.fid              \              # input nmrpipe format fid
| nmrPipe -fn SOL                  \              # remove H2O peak at center of spectrum
| nmrPipe -fn SP -off 0.5 -pow 2  \              # apply 90deg shifted sinebell ( cosine)
| nmrPipe -fn ZF -auto             \             # zero fill to next power of 2
| nmrPipe -fn FT -auto             \             # fourier transform complex data
| nmrPipe -fn PS -p0 -14 -p1 0.0 -di      \     # apply phase correction in f2
| nmrPipe -fn EXT -left  2 -sw  -verb \          #  keep only the left half of the spectrum
| nmrPipe -fn TP                   \            # transpose the data matrix
#| nmrPipe -fn LP -fb   \                       # linear predict in t1 ( commented out)
| nmrPipe -fn SP -off 0.5 -pow 2  \              # apply cosinebell function
| nmrPipe -fn ZF -zf 2           \              # zero fill twice the size
| nmrPipe -fn FT -auto            \             # fourier transform
| nmrPipe -fn PS -p0 -90 -p1 180  -di -verb   \    # apply phase correction.
| nmrPipe -fn TP     \                          # transpose back so proton is on x-axis
 -ov -out test.ft2
```

You can run this script and then look at the resulting spectrum, 'test.ft2 '