Introduction to Data Processing and Analysis using NMRPipe in NMRBox (Part 1)

Modern NMR spectrometer systems include host computers with software provided by the manufacturers (mostly Varian/Agilent and Bruker) for acquiring, processing, displaying and analyzing NMR data. Nevertheless, there are many other software packages distributed by academic and commercial sources for processing, displaying and analyzing NMR data that may be used independently of the spectrometer ("NMRPipe", "NMRDraw", "Sparky" and "NMRFAM-Sparky", "NMRView", "Mnova NMR", "iNMR", "Felix NMR", "Nuts", to name just a few). These each have their strengths and weaknesses, and some were designed and are used for only very specific tasks ("Sparky", for instance, was designed for assigning resonances of proteins and other macromolecules).

Here we will introduce the software package NMRPipe (NMRPipe/NMRDraw) which is one of the most popular software packages for processing and analysis of biomolecular NMR data. This software was developed originally by a group of researchers at the National Institutes of Health (NIH), with Frank Delaglio as the principle developer. The program continues to be developed by Dr. Delaglio, in conjunction with the NIH and National Institute of Standards and Technology (NIST). The current website for the software package is: https://www.ibbr.umd.edu/nmrpipe/index.html

The tutorial that follows includes elements specific to the use of NMRPipe in NMRBox. It assumes you have a NMRBox account, that you have installed the VNC Viewer (required for accessing the NMRBox), that you know how to transfer files to/from your NMRBox server, and that you know how to use a 'Terminal' window for using the UNIX/LINUX command line.

So, you need to do the following before beginning this tutorial:

1) The first page of the '**NMRBox - creating an account, logging in, transferring files**' tutorial tells you how to go about registering for an NMRBox account. You need an account before you can do the 'Mnova' exercises. Because it may take up to three business days for your account to be approved after you register, you need to register first. In the meantime, while you are waiting for your account to be approved, you can install the VNC viewer (see step 2).

2) Install VNC Viewer (see the '**Installing the VNC Viewer**' tutorial on the course website)

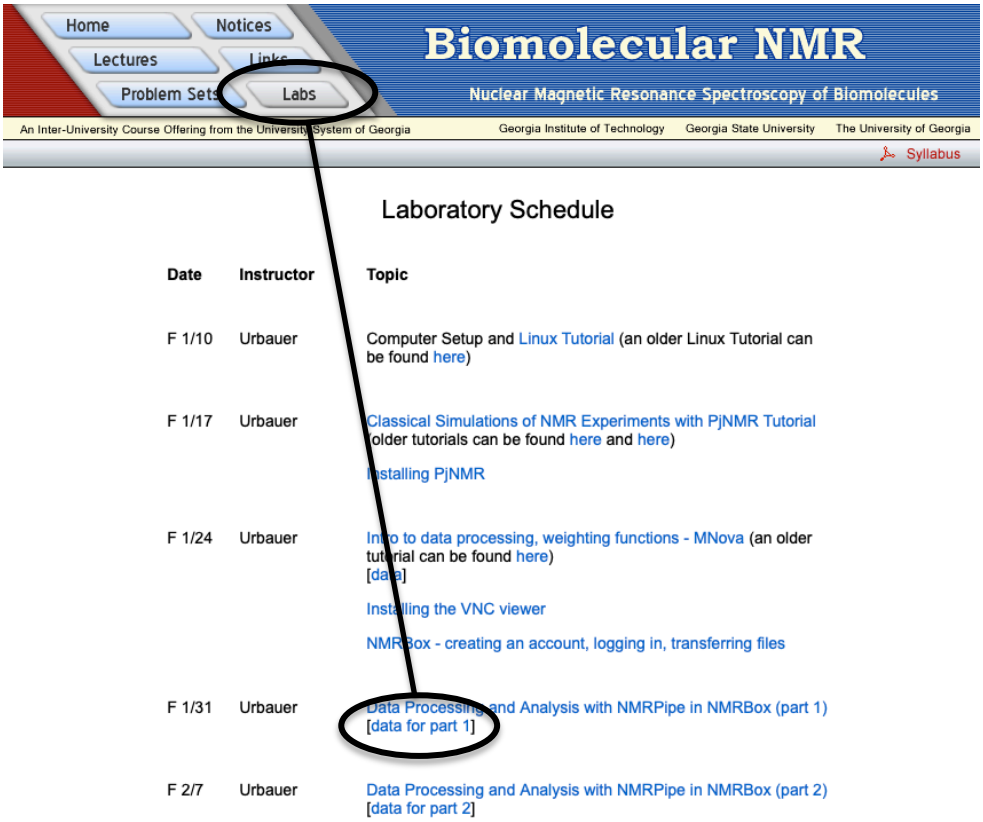
3) Once you have an NMRBox account, you can continue with the '**NMRBox - creating an account, loggin in, transferring files**' tutorial to learn how to log in to your NMRBox account using the VNC Viewer, and how you can transfer files and folders between your computer and your NMRBox account. You should complete that tutorial before proceeding to step 4.

4) Once you've completed step 3, you can proceed (below) with the NMRPipe tutorial '**Introduction to data processing and Analysis using NMRPipe (Part 1)"**.

Finally, the "Demonstration Data" page at the NMRPipe website includes a very good introduction to NMRPipe about two pages down from the top ("NMRPipe: Introduction"), and the NIH still hosts two good tutorials. Here are the links: https://www.ibbr.umd.edu/nmrpipe/demo.html, https://spin.niddk.nih.gov/NMRPipe/doc1/, https://spin.niddk.nih.gov/NMRPipe/doc2/.

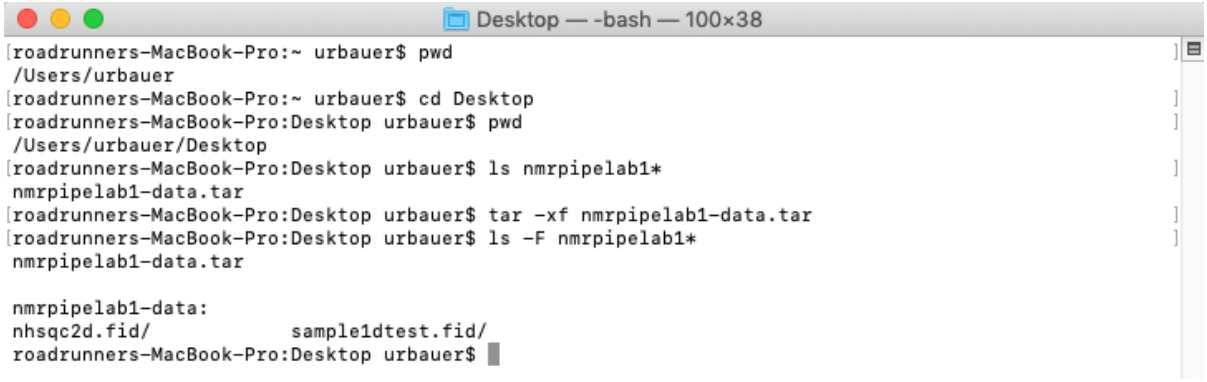
Transferring the NMRPipe lab #1 data to your NMRBox account:

First, download the data from the course website to your computer.



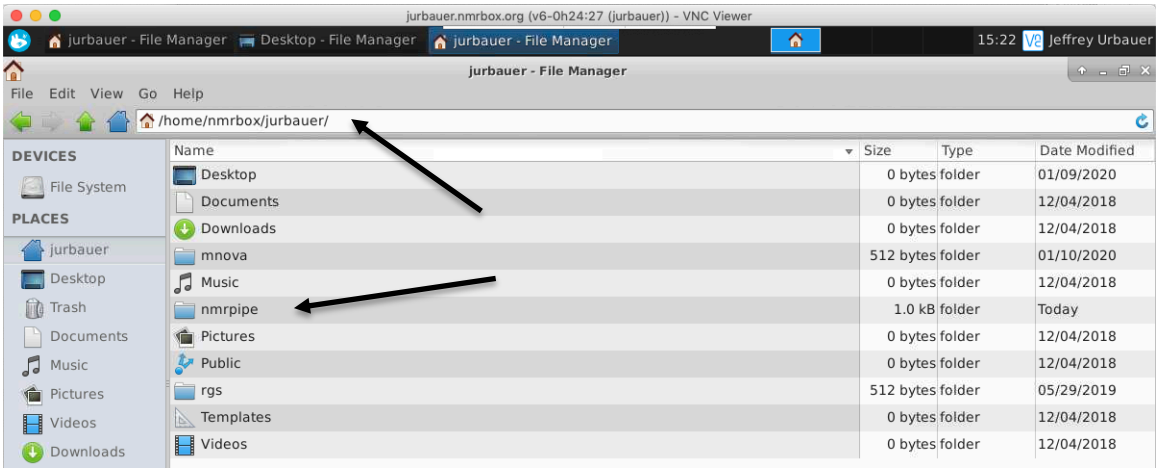
You can put the file anywhere, but, for this tutorial, we'll assume it is on the **desktop**.

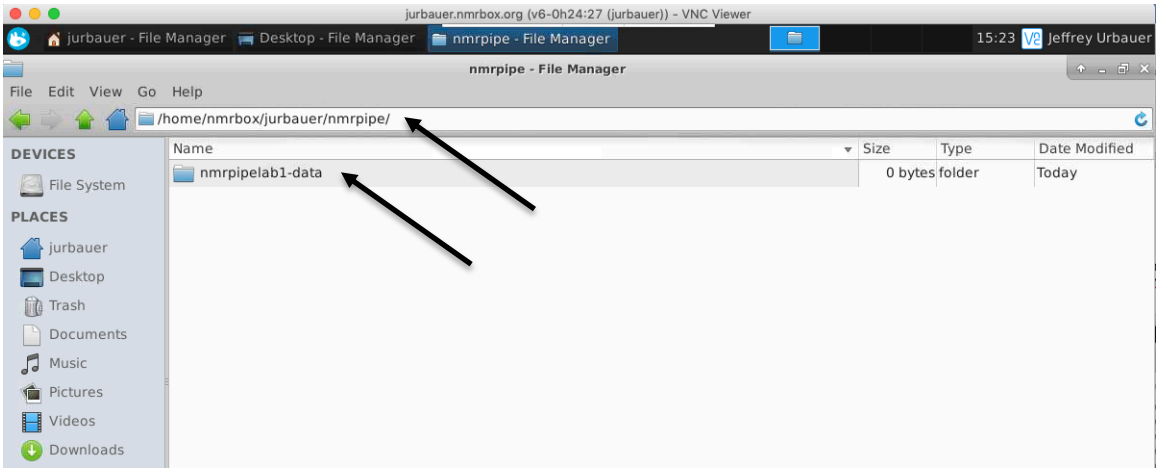
The downloaded file is named 'nmrpipelab1-data.tar'. Your browser may have already opened/extracted the '.tar' file, in which case you will have a folder named 'nmrpipelab1-data'. If not, you need to extract the files. You will accomplish this with the 'tar -xf' command, as shown below (for user 'urbauer' using MacOS).

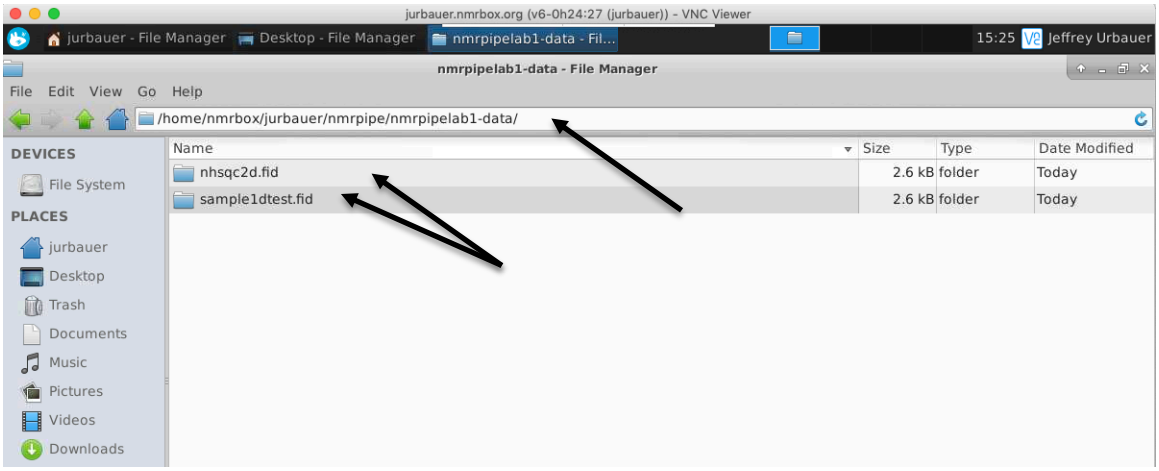


Now there is a folder/directory on your desktop called 'nmrpipelab1-data' that has two subfolders that are Varian spectrometer data directories (Varian 'fid' folders/directories).

Following the instructions in the ‘NMRBox – creating an account, logging in, transferring files’ tutorial on our course website, log in to your NMRBox account and create a new folder named ‘nmrpipe’, and then transfer the ‘nmrpipelab1-data’ folder into that directory (below, for user ‘jurbauer’).







Now, in your ‘nmrpipe’ directory you have a subdirectory named ‘nmrpipelab1-data’, and in that directory you have two subdirectories called ‘nhsqc2d.fid’ and ‘sample1dtest.fid’ which are Varian/Agilent data (“fid”) directories.

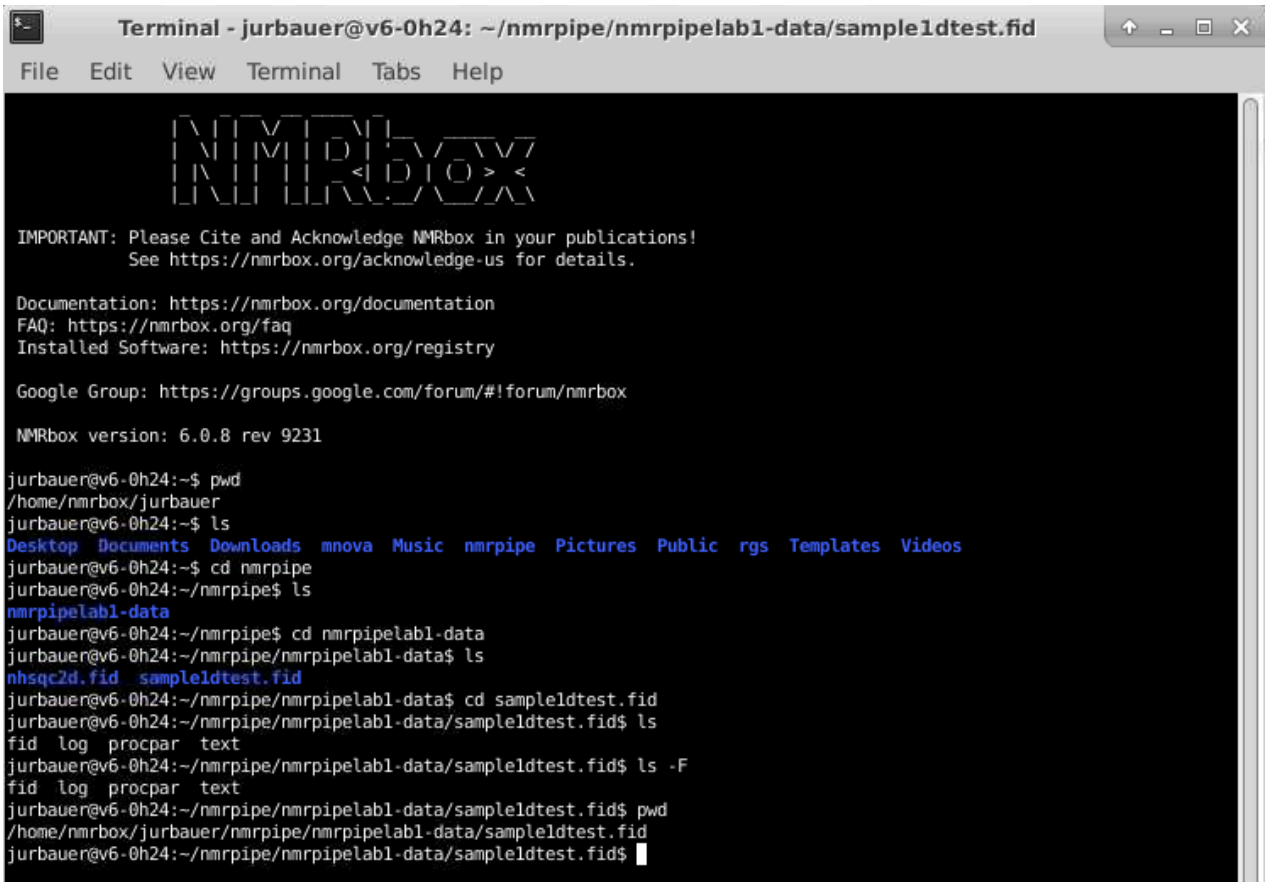
Converting spectrometer data into NMRPipe format:

NMRPipe uses its own data format, so raw data from NMR spectrometers must be converted into NMRPipe format in order for NMRpipe to be able to process it.

The data to be used in this tutorial is from a Varian/Agilent spectrometer, so we will use a NMRPipe script aptly named 'varian' to convert the data (a similar command, 'bruker', is used for data from Bruker NMR spectrometers).

NMRPipe commands are entered from the UNIX/LINUX command line. So, in NMRBox, click the terminal icon at the bottom of the NMRBox window (or right-click in any emty space in the NMRBox window and select 'Open Terminal Here') and a terminal window will appear. You can move this window around and resize as necessary.

It is very important, before invoking the NMRPipe 'varian' conversion script, to use 'cd' to go into the Varian data directory that holds the data you are interested in processing. In our case, use '**cd**' to go into the 'sample1dtest.fid' directory. With '**ls -F**' you'll see there are four files in there (no subdirectories). The 'fid' file holds the data. The 'varian' command will extract the data from the 'fid' file, and, in addition, it will extract other information from the 'procpar' file. So, it is important to '**cd**' into this directory, and it is. Important that these files are present.

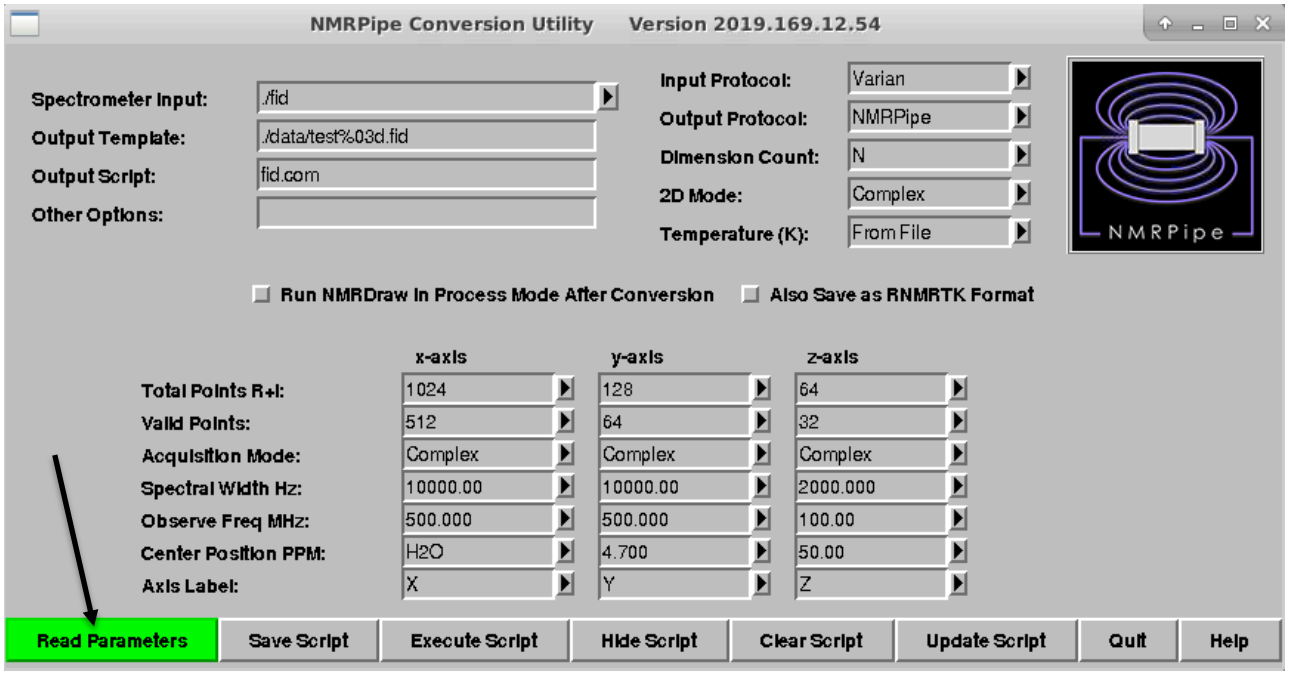


Once you have used '**cd**' to go into the 'sample1dtest.fid' directory, simply enter the command '**varian**' on the command line (omit the quotes).

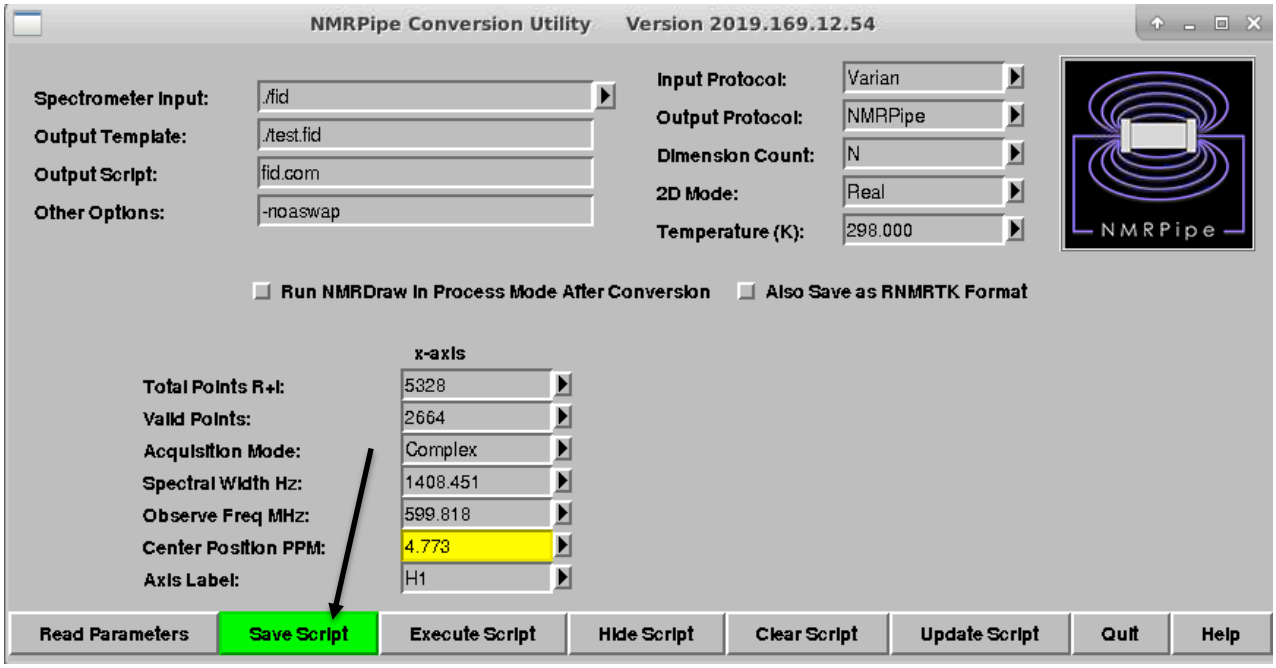
When you enter the 'varian' command, two new windows will appear, the 'NMRPipe Conversion Utility' window and the 'Conversion Script Text' window. The 'NMRPipe Conversion Utility' window is used to read the data and create an NMRPipe script to convert the data properaly, and the script will appear in the 'Conversion Script Text' window (which is empty initially).

Take a look at the items in the 'NMRPipe Conversion Utility' window. In the upper left the 'Spectrometer Input' indicates the data will be extracted from the './fid' file. The processing script that will be generated will be named 'fid.com'. The items at the upper left indicate that the input data are from a 'Varian' instrument, so the protocols ('Input Protocol') to be used to convert the data to NMRPipe format are the 'Varian' protocols. Other parameters, such as 'Temperature' will be read 'From File' (i.e. from the 'procpar' file). In the center of the window are two options you should leave unchecked for now. At the bottom are items that describe the data set and will be extracted from the 'fid' and 'procpar' files ('Total Points R+I' corresponds to the number of points in the fid, 'Valid Points' is typically half of the 'Total Points', 'Acquisiton Mode' indicates how the data were collected (almost always this will be 'Complex', as both real and imaginary points are collected), 'Spectral Width', 'Observe Freq(uency)', 'Center Position' (center of the spectrum in ppm, i.e. the reference frequency or 'carrier' frequency), and axis label. The initial values are just default (place holder) values, and will change when the actual parameters from our experiment are read in. The 'x-axis', 'y-axis', and 'z-axis' correspond to dimensions of a three-dimensional experiment.

The first step is to **click on the 'Read Parameters' button** (which is highlighted in green). This will cause NMRPipe to read the data (from the 'fid' file) and parameters (from the 'procpar' file).

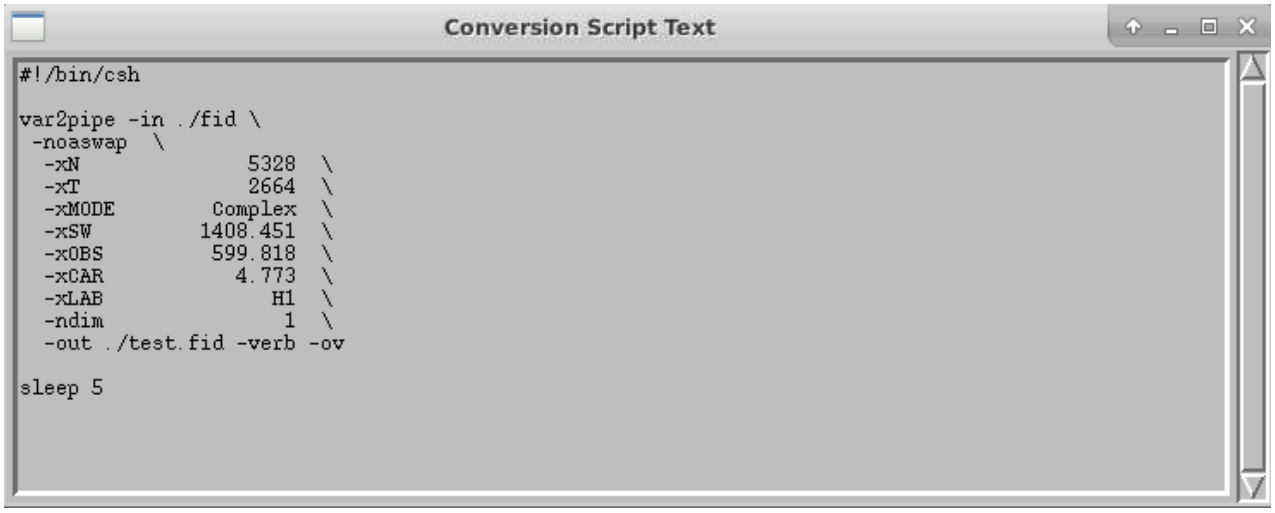


Once the ‘Read Parameters’ button is clicked, the data and parameters from the current experiment are read in, and the values for the items in the ‘NMRPipe Conversion Utility’ window change accordingly. Note that, because these data are from a one-dimensional experiment, the ‘y-axis’ and ‘z-axis’ columns are gone. The ‘x-axis’ parameters indicate this is 1H experiment, using a 600 MHz instrument (‘Observe Freq MHz’ = 599.8 MHz), the spectral width is 1408.451 Hz, and 2664 complex points were collected.

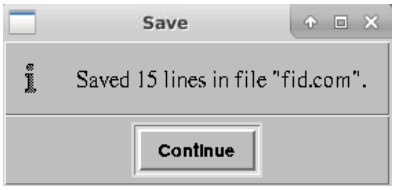


You’ll see that the ‘Conversion Script Text’ window is no longer empty. What is in the window is a script, using NMRPipe “language”, to convert the data into NMRPipe format. Most of the parameters in this script should now look familiar to you, based on what is shown in the ‘NMRPipe Conversion Utility’ window. At the end of the script, the ‘-out’ function indicates that this script will be given the name ‘test.fid’ and will be placed in the current directory (‘./test.fid’). This script can be used then again if you need to convert the data again (no need to regenerate it), and it can be edited if there is a reason to.

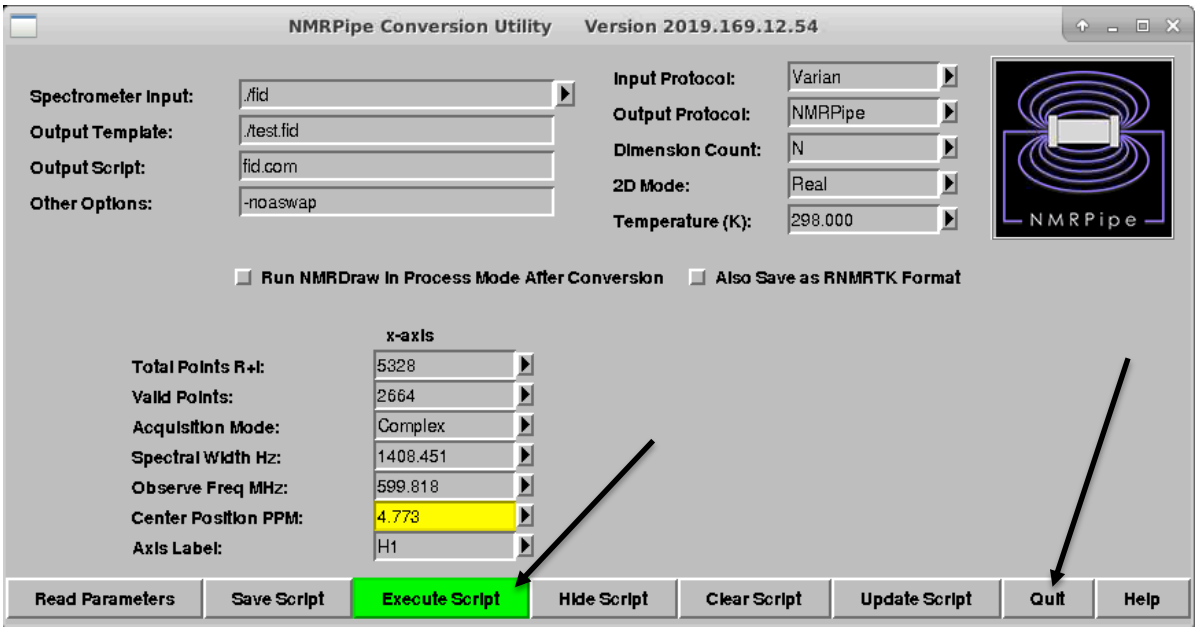
You should now **click the ‘Save Script’ button** to save this script, as ‘fid.com’, in the current directory.



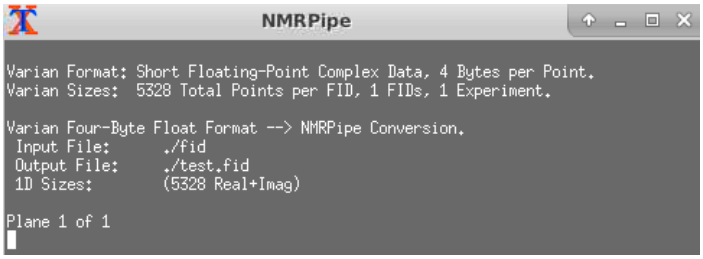
Once you click on the ‘Save Script’ button, a small window will appear (if the process was successful) indicating that the script has been saved, here as ‘fid.com’. **Click the ‘Continue’ button** to continue.



In the ‘NMRPipe Conversion Utility’ window, the ‘Execute Script’ button will now be highlighted. This will then execute the ‘fid.com’ script and generate the ‘test.fid’ file, which is the data in NMRPipe format.



Click the ‘Execute Script’ button. You’ll see a new window appear that gives some details about the conversion, including the input and output file names and an indication (‘Plane 1 of 1’) that this is a one-dimensional data set. This window is transient and will disappear in a few seconds.



Click the ‘Quit’ button. This will end the data conversion session. The ‘NMRPipe Conversion Utility’ and ‘Conversion Script Text’ will disappear. If you use the ‘ls -F’ command, you will see that your ‘sample1dtest.fid’ directory now contains the NMRPipe formatted data (‘test.fid’) and the NMRPipe language script used to create it (‘fid.com’). The ‘fid.com’ script has an asterisk next to it, indicating that it is an executable file/program that you can use later to regenerate the ‘test.fid’ file.

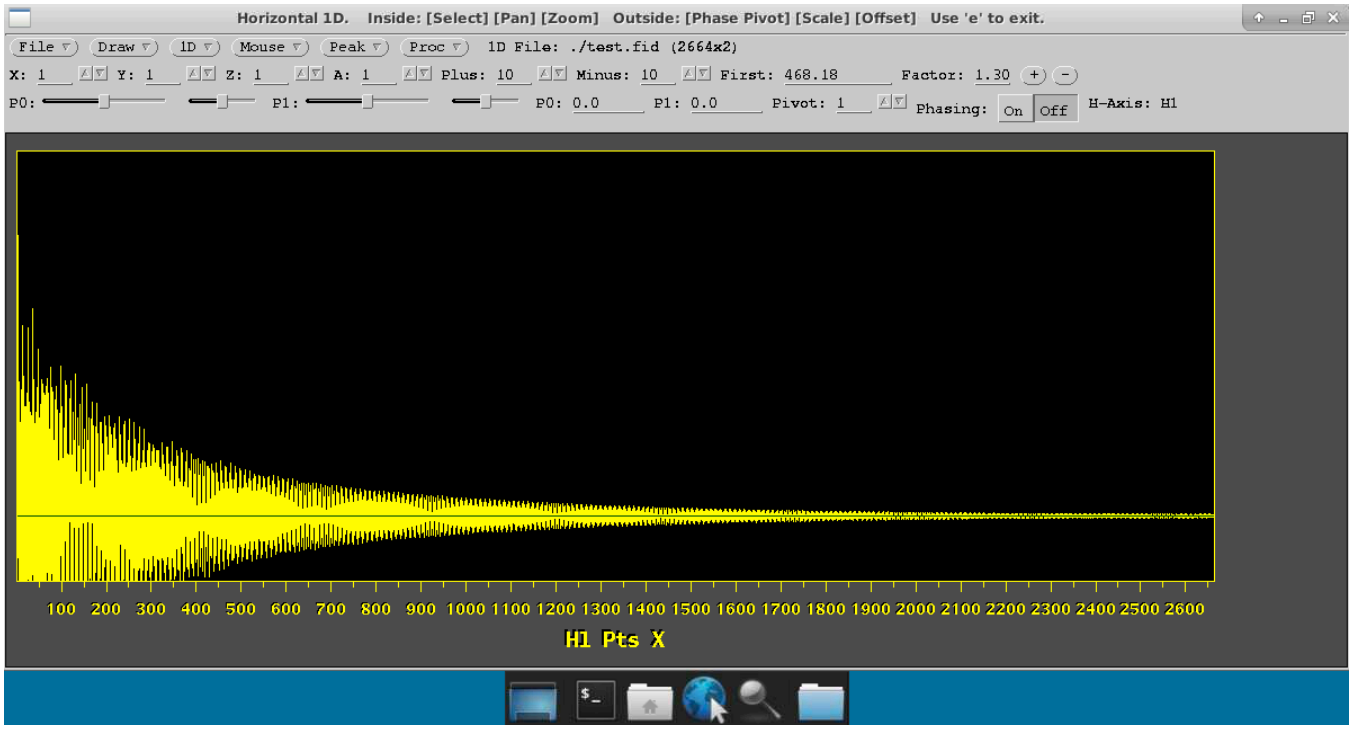
Note that the last four pages of this tutorial can be summarized as: **cd into your data directory**, **type ‘varian’, click ‘read parameters’**, **click ‘save script’**, **click ‘execute script’**, then **click ‘Quit’**.

Simple display of 1D data using the NMRDraw program of NMRPipe:

The NMRDraw program of NMRPipe allows for processing and display of NMR data. For simple 1D data, built in functions in the menu allow for quick and easy processing and display. For more complicated experiments (2D, 3D), processing scripts are typically generated and used for processing.

Please note that the NMRDraw graphical interface uses both **left-clicks** and **right-clicks**. So, in the instructions below, pay close attention to the instructions. Second, if you are using a laptop, it is best to connect a three-button mouse if possible. If you don't have that option, a right-click is usually some combination such as control-click, option-click, command-click, or perhaps some other combination.

Again, to start the NMRDraw program, we will use the UNIX/LINUX command line. First, for the best results, you should **'cd' into the directory with the data**, which now includes not only the raw data ('fid') but the data in NMRPipe format ('test.fid'). Then, to start NMRDraw, from the command line simply **type 'nmrDraw'** (omit the quotes).

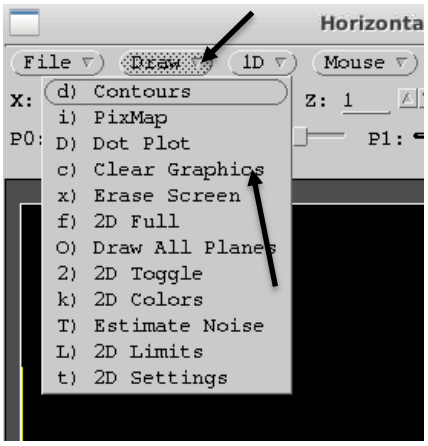


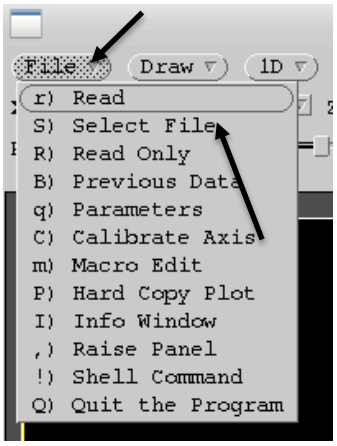
If NMRDraw finds a 'test.fid' file in the current directory, it usually will just open that file and display the contents. Shown in the figure above is the fid from our experiment. Note that our data included 2664 complex points, and you see in the figure what appears to be 2664 points displayed.

Sometimes NMRDraw may fail to find the NMRPipe formatted data file (for example, if it has a non-standard name), in which case you must find the file and tell NMRDraw to open it. Also, even though it is good practice to open NMRDraw in the data directory, this is not absolutely necessary. If so, then NMRDraw will have to be 'told' explicitly where the file is.

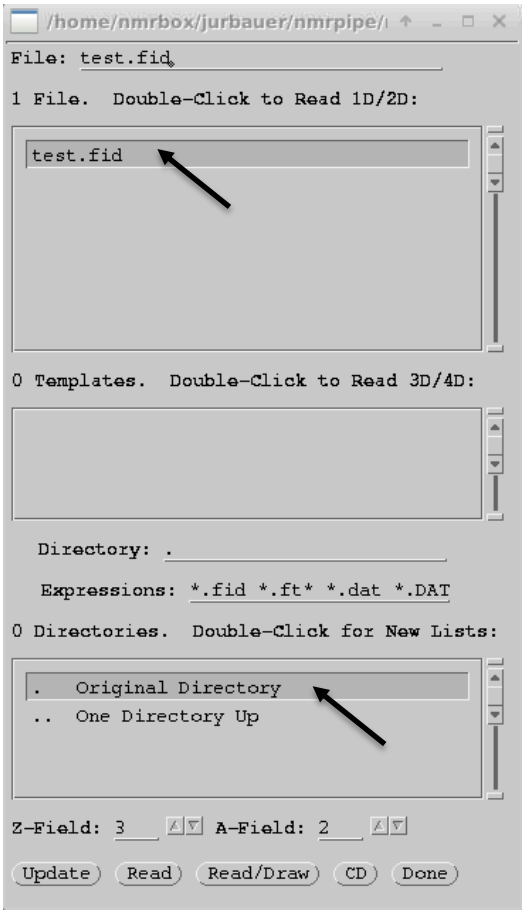
Regarding the drop-down menus shown at the top of the display in the figure above ('File', 'Draw', '1D', 'Mouse', etcetera), typically a right-click opens and closes the menu, while a left-click selects a menu item from an open menu.

We'll close this data file and then reopen it manually. First, **right-click on the 'Draw' menu**. Then **left-click on the 'Clear Graphics'** option (the 'Erase Screen' option will work as well). Once you do, the graphics on the screen (our fid) will disappear.





Then, **right-click on the 'File' menu**, and **left-click to select the 'Select File'** option. The file selector window (below) then opens (you have to resize the window to see the buttons at the bottom).



The directory selector at the bottom of the window allows you to select the directory that your data file is in (in our case, the directory is '.' or 'Original Directory'). Once you have selected the correct directory, the filename of your file should appear in the upper part of the window (in our case, 'test.fid'). Then, **double-left-click on the 'test.fid' option**. You'll see the fid displayed (blocked partially by the file selection window). Click the 'Done' button to remove the menu.

Often, if you started NMRDraw from the data directory, your data file is already selected when the file selector window initially opens, as it is in the window on the left. If so, all you need to do to display the data is click the 'Read/Draw' button at the bottom of the window (then click 'Done' to remove the menu).

Also, if your data was displayed previously, often all you have to do to display it again is left-click the 'File' menu button. Go ahead and clear the graphics again (**right-click on the 'Draw' menu**, then **left-click on the 'Clear Graphics'** option). Then just **left-click the 'File' menu button**, and your data will reappear.

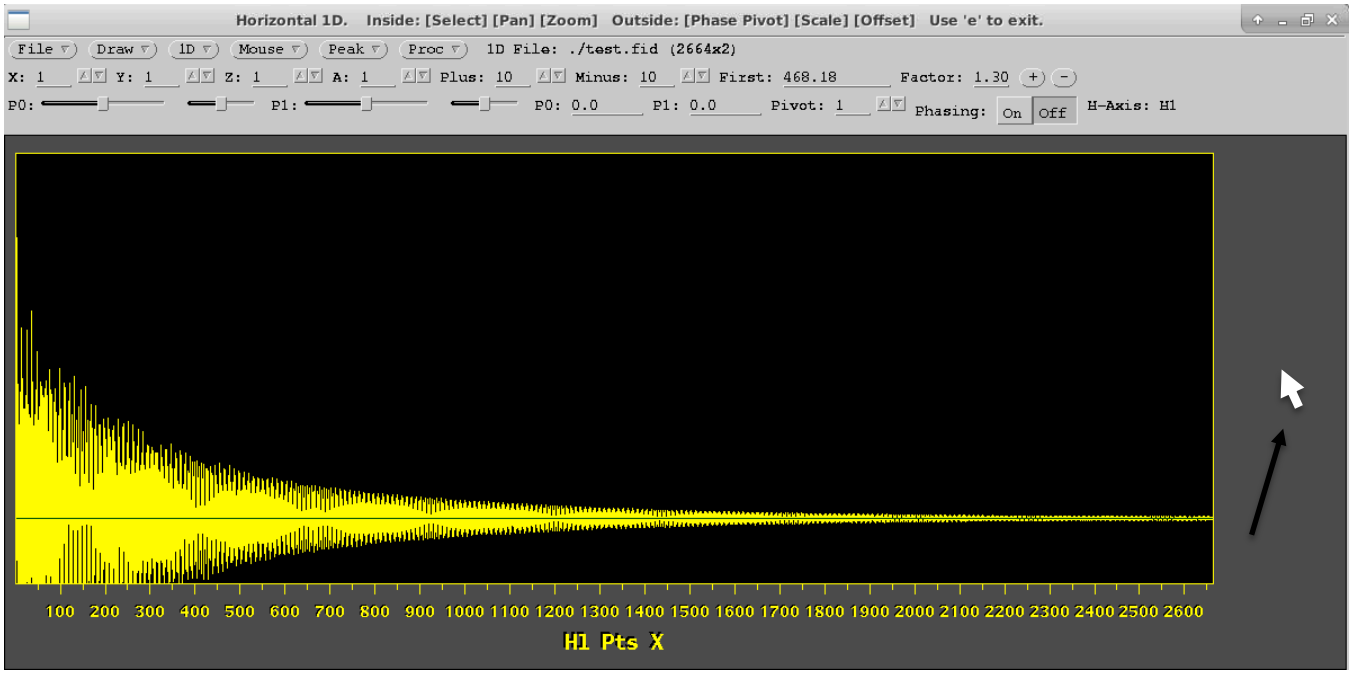
Often you'll find that NMRDraw has multiple ways to accomplish the same task.

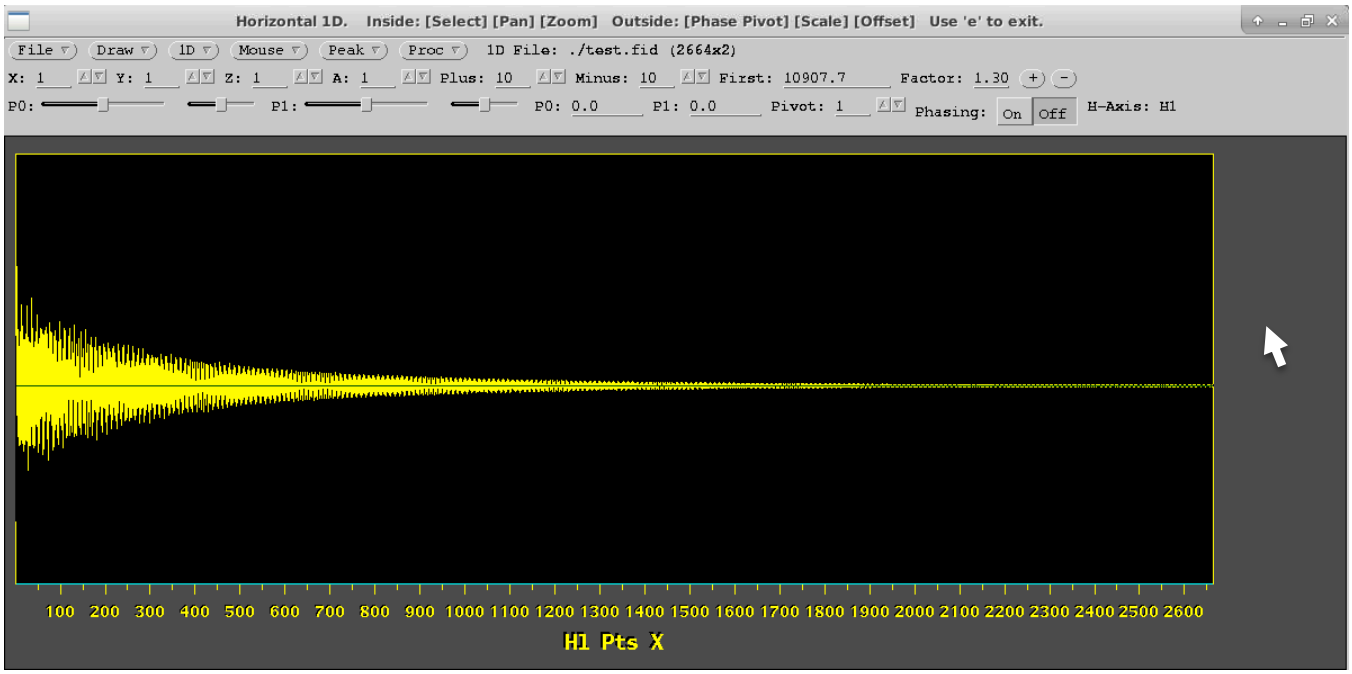
Hint: sometimes the NMRDraw program will 'glitch', and the data (or spectrum, if you have already processed the data) will disappear. If you left-click the 'File' menu, either the data or the spectrum will return. This is the quickest way to recover. If it doesn't work, you may have to quit the program (right-click on the 'File' menu, then left-click on the 'Quit the Program' option) and start over.

Simple display changes:

As shown below, the fid is displayed such that the vertical scale is too large for the window, and the data are not centered vertically in the window.

To change the vertical scale and the vertical position, put the cursor in the gray area to the right of the spectral display window. Then, to **adjust the vertical position**, **hold down the right mouse button and move the cursor up/down**. To **adjust the vertical scale**, **hold down the middle mouse button and move the cursor up/down** (rolling the scroll wheel will have no effect). The data in the lower figure (below) have the vertical position and vertical scale adjusted.



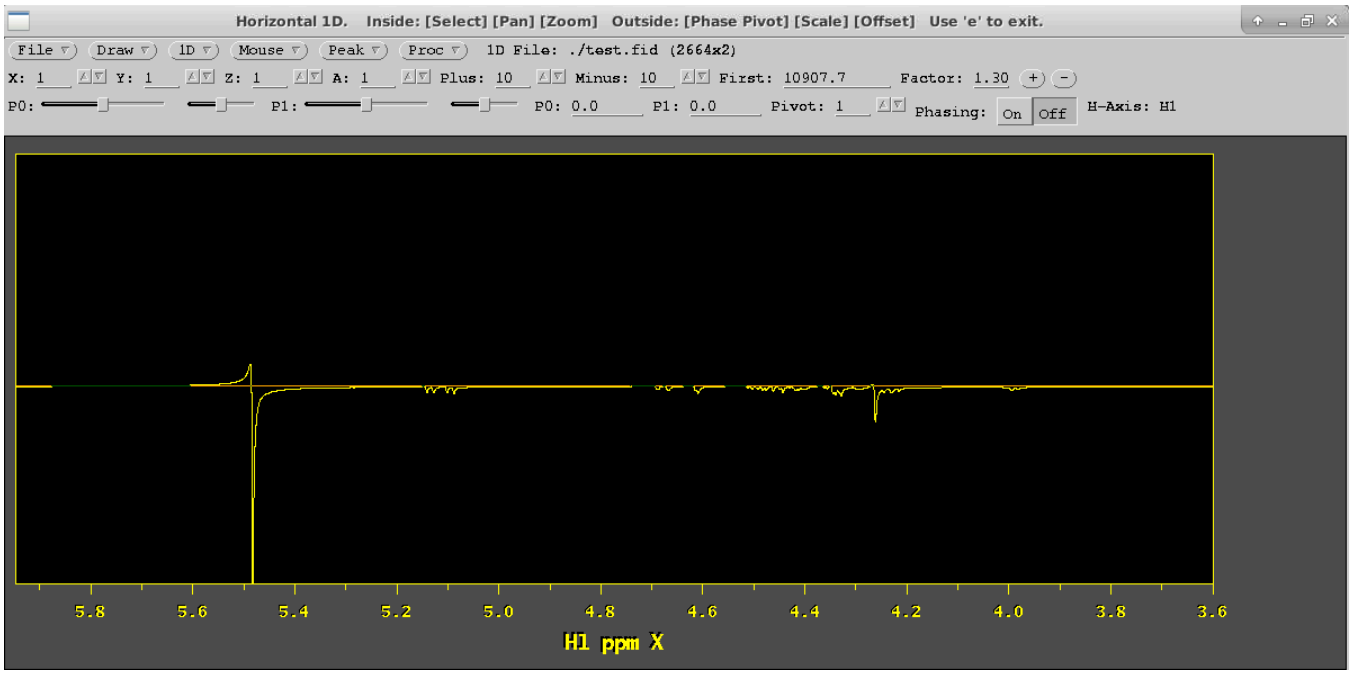


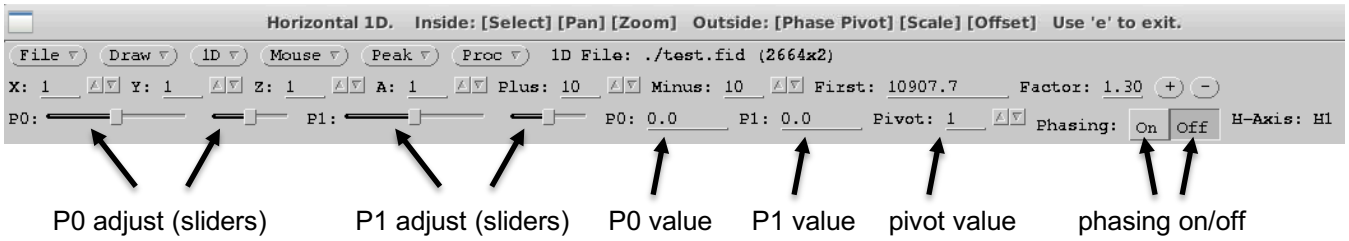
Fourier transformation, phasing, and spectral display:

Here we'll use the simple, built-in functions in the NMRDraw menus to Fourier transform our 1D data and display the spectrum.

First, following the tutorial above, open the data and display the fid (as shown on the previous page).

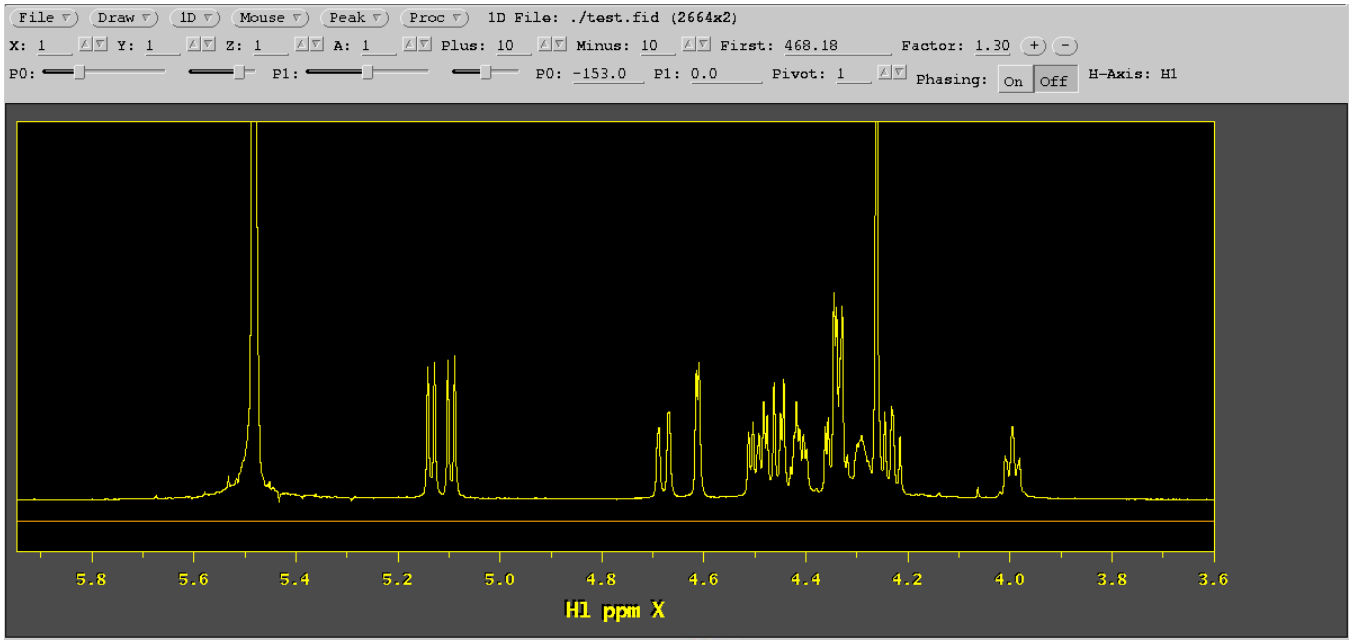
The quickest and easiest way to Fourier Transform the data is using the 'Auto-Process 1D' menu option in NMRDraw. First, **right-click the 'Proc' menu** to see the options, and then **left-click on 'Auto-Process 1D'**. You'll see the spectrum displayed as shown below, with the signals far out of phase.





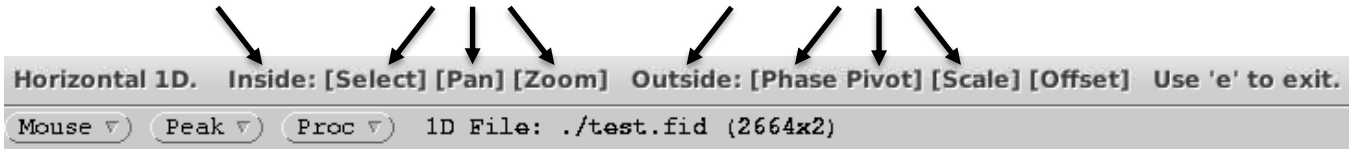
Phasing the spectrum is accomplished with the functions that comprise the bottom row of menus. At the right are the buttons that turn phasing on/off. At the left are sliders for adjusting the zero-order (P0) and first order (P1) phase corrections. The left-most slider of each pair is the 'coarse' adjustment, the right slider is the 'fine' adjustment. The values of P0 and P1 are also displayed. There is the option to set a 'Pivot' (do yourself a favor, and always keep the 'Pivot' at the smallest value, usually 0 or 1).

This spectrum is phased properly with only a zero-order (P0) phase correction of -153.0. So, first, **click the 'On' button** to turn phasing on, then **adjust the value of P0, using the coarse slider, to something close to -153.0**, then use the **fine adjust slider to get exactly -153.0**. Then, **make sure to click the 'Off' button** to turn phasing off. Using the procedures you learned above for adjusting the vertical position and vertical scale, you can adjust the spectrum to create a better display (as below).



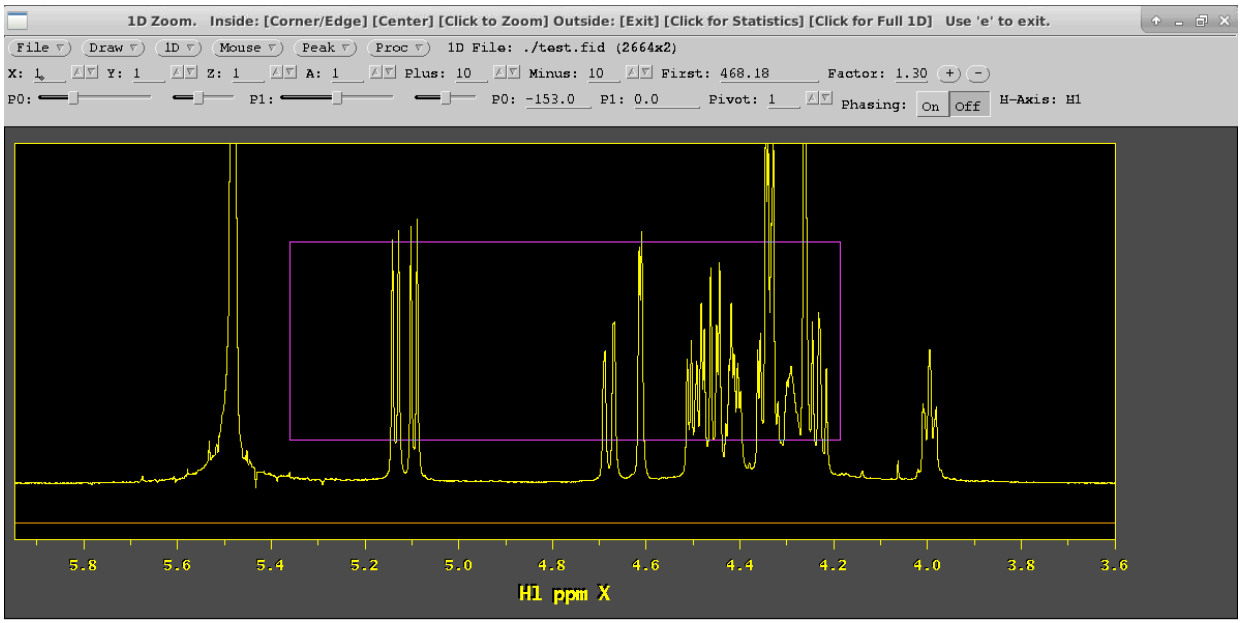
More often than not, the spectrum will require both zero (P0) and first (P1) order phase corrections. In this case, keeping 'Pivot' at the smallest value, iteratively adjust P1 and P0 until all signals are phased.

Notice, at the very top of the NMRDraw main window is a line which defines the functions of the buttons on the mouse. When the cursor is 'Outside' of the data window (i.e. in the gray area to the right), the right button controls vertical position/offset ('Offset') and the middle button controls vertical scale ('Scale'), as we saw previously. The left button controls 'Phase Pivot'. So, when the cursor is 'Outside' the main window (in the gray area to the right), moving the mouse up/down while holding down one of the mouse buttons has the indicated function. When the mouse is 'Inside' the data window, the left mouse button allows selection of a window to zoom in on ('Select'), the middle mouse button allows the window to be moved around ('Pan'), and the right mouse button allows the selected region to be expanded.



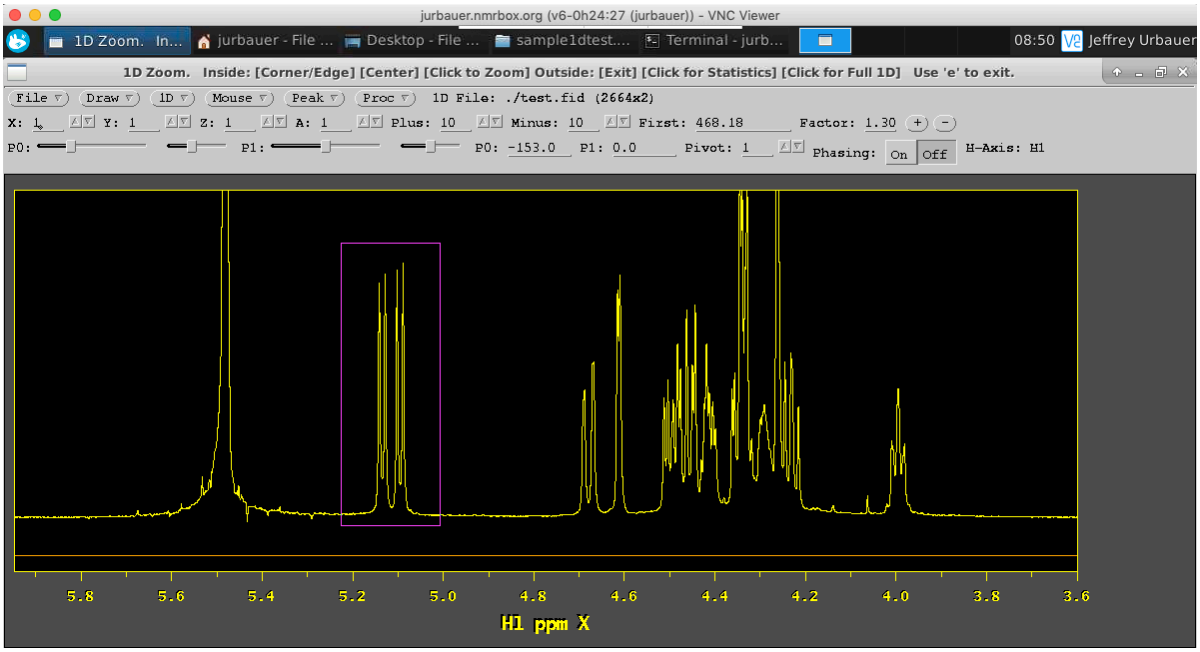
For the 'Inside' functions to work, a selection first has to be made from a menu. This is demonstrated below for the 'zoom' functions.

In order to expand a region of the spectrum (‘zoom in’), **right-click on the ‘Mouse’ menu**, then **left-click on the ‘1D Zoom’** option to select it.

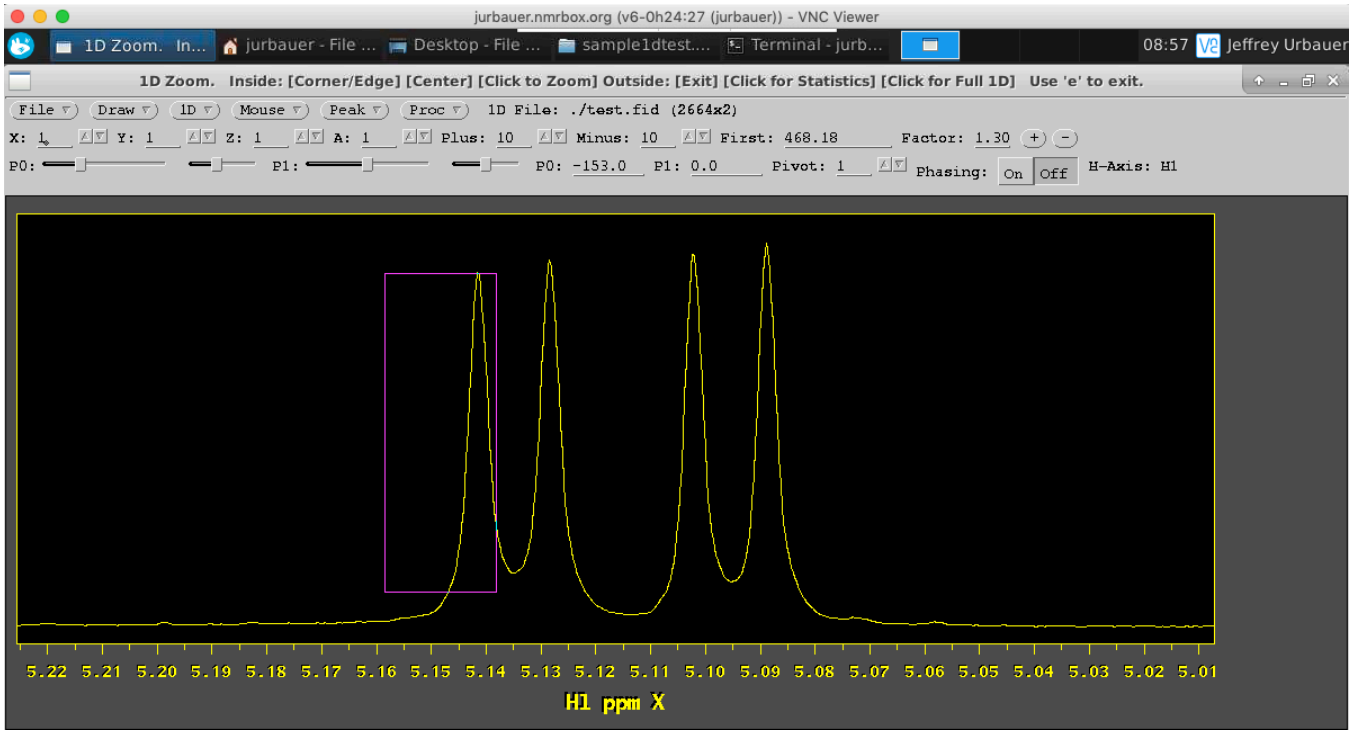


A purple box will appear. This box defines the area of the spectrum that you would like to zoom in on. The size and position of the box can be adjusted using the left and middle mouse buttons. Notice the functions of the mouse buttons have changed somewhat.

Using the left mouse button, click (and hold) on parts of the purple box to resize it. Using the middle mouse button click (and hold) on the box to move it around. Please note that these functions can be a bit ‘glitchy’. Below, we’ve placed the box around the pair of doublets centered at about 5.12 ppm.



Once you've defined the area you want to expand, simply right-click (with the cursor inside the spectrum window). This will expand the region chosen, as shown below. The purple box will still be present. As shown on the top of the window, to exit this mode (remove the purple box) simply type 'e' on the keyboard, and the box will disappear.



NMRDraw has many 'shortcut' keys that allow you to perform tasks without accessing the menus and menu items. For instance, if you right-click on the 'Mouse' menu, you'll see that the last entry, 'Exit Mode' has the shortcut 'e' ('e' for 'Exit'). So, above, when you typed 'e' to get rid of the purple box, you could have also right-clicked the 'mouse' menu, and then left-clicked the option 'Exit Mode'.

In the menus, you'll see many examples of these shortcut keys. For instance, in the spectrum shown above we have zoomed into one region of the spectrum. If we would like to once again display the full spectrum, we could right-click on the '1D' menu, and then left-click on the '1D Full' option. You'll notice next to the '1D Full' option is the shortcut 'F' (upper case F). So, type 'F' and the full spectrum should once again be displayed. You can also experiment to see what happens if you simply type 'f'.

NMRDraw has a full complement of shortcut keys to accomplish various tasks. Once you become familiar with NMRDraw, these are a great way to save time.

Thus far we've not yet demonstrated the use of apodization functions in NMRPipe/NMRDraw, or other processing functions such as zero-filling or linear prediction. We'll do that below.

Apodization:

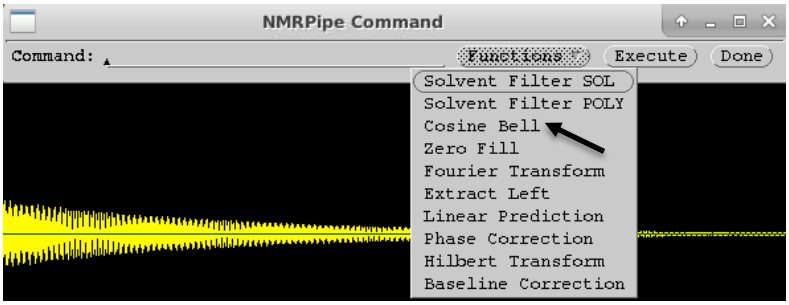
Here we demonstrate how to use the NMRDraw menus to apply apodization functions to our fid, zero-fill, and perform linear prediction.

At this point, you should quit the program (**right-click the 'File' menu**, **left-click the 'Quit the Program'** option), and then restart it in the same data directory as before. If your data (fid) are not displayed, go ahead and select the file as described above to display the fid.

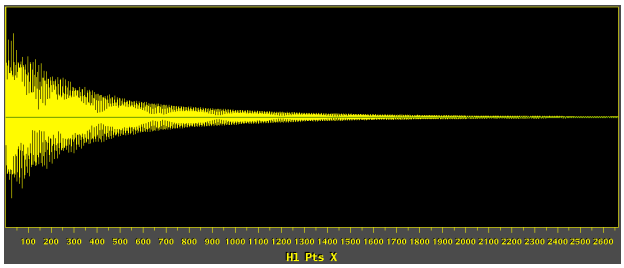
First, **right-click on the 'Proc' menu**. Note the 'Convert Varian' option. Go ahead and **left-click on the 'Convert Varian' option** and you'll see the 'NMRPipe Conversion Utility' window and the 'Conversion Script Text' window. So, if you choose, in the future, you can perform the data conversion within NMRDraw, rather than using the 'varian' command from the UNIX/LINUX command line. **Click the 'Quit' button** to close the 'NMRPipe Conversion Utility' and 'Conversion Script Text' windows.

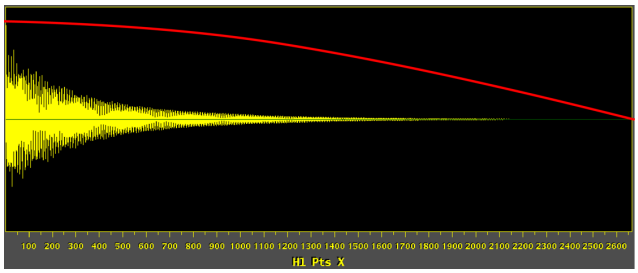
Now, once again **right-click on the 'Proc' menu**. Then, **left-click on the 'NMRPipe Command'** option. This will open the 'NMRPipe Command' window (see below). In this window we can enter commands to modify our fid using apodization/window functions, zero-fill, linear predict, and otherwise manipulate our fid. We can also perform the Fourier transform from this command window.

Right-click on the 'Functions' menu. This opens a menu with functions that we can use to process and Fourier transform our data. Then **left-click on the 'Cosine Bell'** function to select it. In the 'Command' window, you'll see the command "SP -off 0.5 -end 1.0 -pow 1". Then **click the 'Execute' button** to multiply the fid by this function.

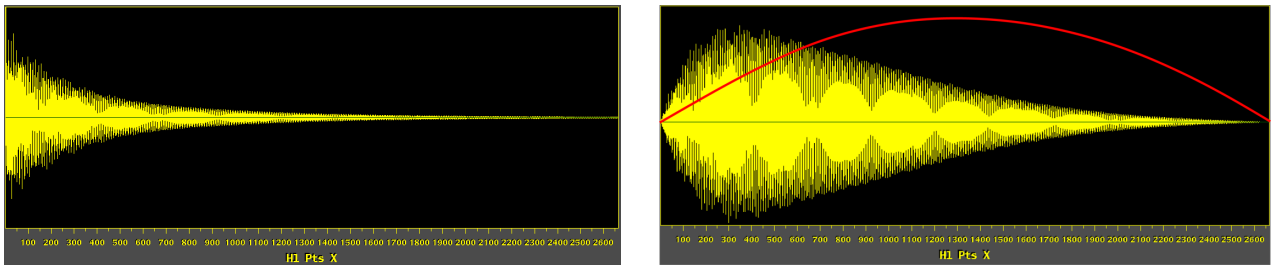


The cosine bell function looks just like it sounds. It is a cosine function that starts at a maximum ("1.0") and decays to zero ("0.0") at the end. When we multiply our data by this function, the first point of our data is unaffected (its amplitude is multipled by 1.0), but the rest are multiplied by ever decreasing values, with the last point of our data multiplied by 0.0. Below are our original data (left) and our data multiplied by a cosine bell (right), with the cosine bell function shown in red.





The cosine bell function was generated with the NMRPipe 'SP' command. This is a general command for generating adjustable sine window functions for apodization. The '-off' option/flag specifies the offset from a sine function in units of pi (π) radians. A value of 0.5 (1/2) indicates the sine function is shifted 90º, i.e. π /2 radians, and therefore is a cosine function. The '-end' option/flag specifies the ending (last) point in the function, in units of pi (π) radians. A value of 1.0 indicates the sine function goes through pi (π) radians or 180º (starts at 0.0 intensity, ends at 0.0 intensity), so a value of 1.0 means the last point in the fid will be multiplied by 0.0. The '-pow' option/flag specifies an exponent. This is normally 1.0, but, some other values (2.0) are sometimes used for specific purposes ("sinebell squared" function). So, this function can be used to generate a variety of window functions for apodization. Below is another example. On the left is the raw data, on the right is the data multiplied by an unshifted sinebell function ("SP -off 0.0 -end 1.0 -pow 1).



For the functions presented above, after you've applied the function, the 'NMRPipe Command' window will remain open. You can then **right-click the 'Functions' button** again, and then **left-click the 'Fourier Transform' option** to get the processed spectrum. You should apply the functions above, Fourier transform the data, and compare the results (i.e. what do these apodization functions do?).

There are many other apodization functions available in NMRPipe, although they are not available from the 'Functions' menu in the 'NMRPipe Command' window:

-**exponential line broadening**: this function improves signal-to-noise at the expense of resolution (for instance, '**EM -lb 1**' then '**Execute**' gives 1 Hz line broadening, '**EM -lb 5**' then '**Execute**' gives 5 Hz line broadening)

-**lorentz to gauss**: this function combines the exponential and gaussian functions and results in a gaussian lineshape that improves resolution at the expense of signal-to-noise by first multiplying the fid by an inverse exponential and then a gaussian (for instance, start with '**EM lb -0.7**' and '**Execute**', followed by '**GM -g1 0.8 -g2 0.8 -g3 0.0'** and '**Execute**')

Like the cosine and sine bell functions, you should try these functions and see the effects on the spectra after Fourier transform.

From the UNIX/LINUX command line, you can get help with these functions with the **'NMRPipe -help -fn'** command. For instance 'NMRPipe -help -fn SP' will give you information on the 'SP' apodization function. There is also (still) an excellent reference resource on the NIH website:

https://spin.niddk.nih.gov/NMRPipe/ref/nmrpipe/

Go to that site and click on the function name at the left (for instance, 'SP') and you'll get a very detailed description of the function and its use(s). Give it a try.

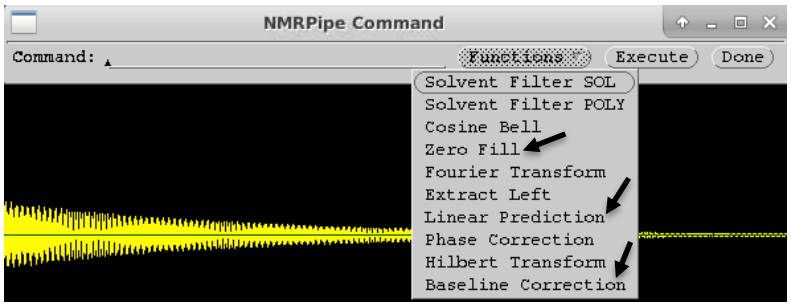
Zero-filling, linear prediction and baseline correction:

Zero-filling adds zeros to the end of a fid. This can help somewhat, sometimes, to improve the digital resolution of the spectrum (more points means fewer Hz per point). Zero-filling in NMRPipe is accomplished with the 'ZF' function

Linear predication is a mathematical procedure whereby the amplitudes of additional points are predicted based on an analysis of existing points. Typically, this procedure is best suited for cases where the fid is truncated, such as in indirect dimensions of multi-dimensional data sets, but it can also be utilized in 1D spectra. Linear prediction in NMRPipe is accomplished with the 'LP' function.

In cases where the baseline is curved or rolling, corrections, typically based on fitting the curvature to an appropriate polynomial equation then subtracting, are very common. Polynomial-based baseline correction in NMRPipe is accomplished with the 'POLY' function.

Experiment with some of these functions. They are all available as options under 'Functions' in the 'NMRPipe Command' window.



If you select 'Baseline Correction', the default command will be 'POLY -auto -ord 1', indicating an automatic baseline correction with a polynomial of degree 1. Because the baseline of our data is very flat, this will not have much of an effect. But should you have data with significant baseline problems, try this function. If you just use 'POLY -auto' the program will attempt to deduce the correct (best) order for the polynomial, or you can try different orders.

If you select 'Zero Fill', the default command will be 'ZF -auto'. This command rounds the number of points up to the nearest power of 2. 'ZF -zf 1' doubles the size of the data set once (the '1' means doubling once). So, if you have 2500 points, the 'ZF -zf 1' will zero-fill the data to 5000 points. The command 'ZF -zf 2' will double the data size twice. So, if you start with 2500 points, doubling once gives 5000 points, and doubling twice gives 10,000 points. You can also select an exact number of points. For instance, 'ZF -zf 16384' zero-fills the data to 16,384 points.

If you select 'Linear Prediction', the default command will be 'LP -fb'. The 'fb' stands for 'forward-backward', which is one method that can be used for linear prediction. This command, by default, uses an 'order' of 8 and doubles the number of points in the fid. You can select the number of points to predict with the '-pred' option. Linear prediction is relative complicated mathematically, and the number of ways to do the prediction is formidable. If you are interested, a good place to start is the NMRPipe NIH resource (link on previous page).

So, experiment with these functions. Compare the processed spectrum before and after zero-filling or linear prediction. See if you can detect changes/improvements in digital resolution.

Processing with macros: creating macro files using NMRDraw:

Using the menus and functions available in them is fine for routine analyses where it is not necessarily important to create and save a record of the processing scheme. However, it is critical for most serious work that a record of every detail of the processing scheme be created and archived.

Recall, when NMRPipe converts the spectrometer data to NMRPipe format, a record is created (the file 'fid.com') and stored (in the data directory). So, if it is necessary to reproduce the processed data, that file (remember, it is an executable file) includes all of the information available to accomplish this.

Just like the data conversion process, it is essential to create a record of all manipulations involved in processing the raw time domain data into the frequency domain data. NMRPipe allows for commands to be issued from a shell script (a 'macro'), allowing for a complete record of the processing procedure to be saved. This is typically the preferred way to process data using NMRPipe for many reasons. When the best route for processing is not known, and many options have to be tried, it is typically much easier and faster to edit a macro file, then run it again, in order to compare processing options. If several data sets need to be processed identically, it is MUCH faster to do this with a macro. For multi-dimensional data sets, the processing can be very complicated, and adjusting/editing a macro to optimize the processing is essential. These are just some of the reasons.

Processing scripts, 'macros', can be written and edited using any text editor (like the data conversion scripts). Below is an example, for instance, for processing a 1D data set, and what each line means:

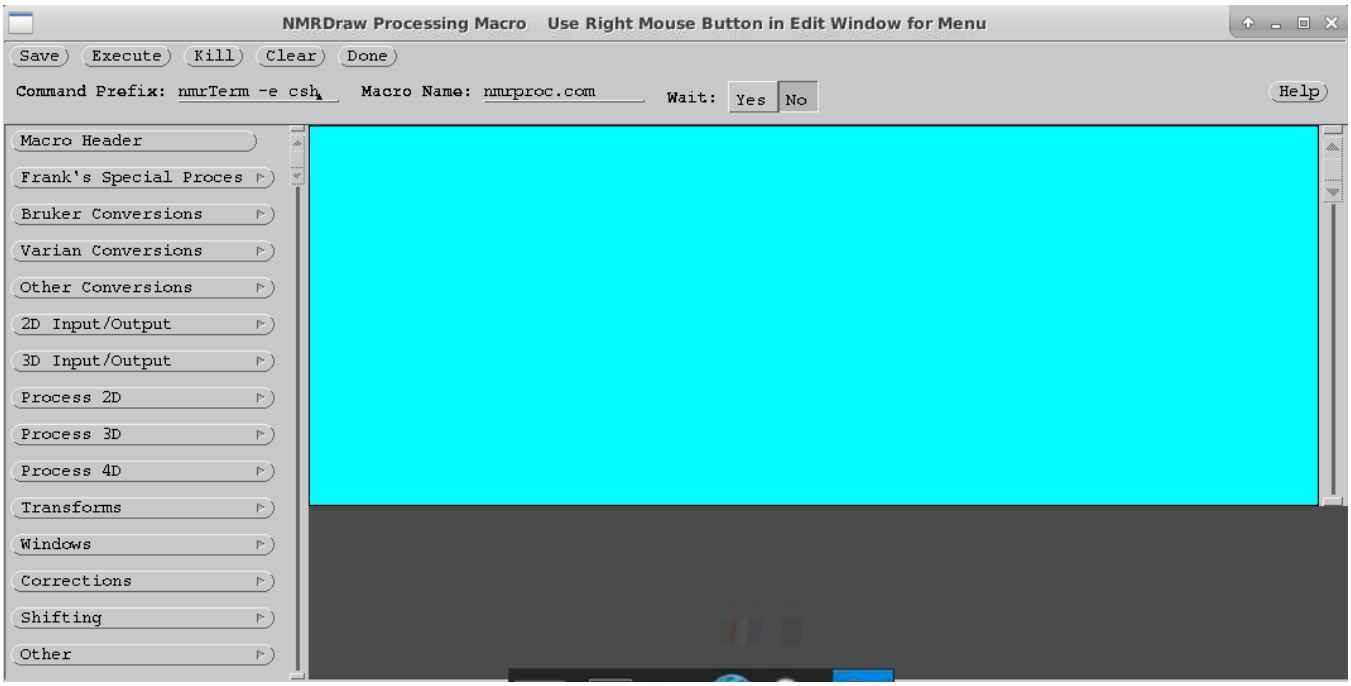
```
#!/bin/csh
nmrPipe -in test.fid              \
| nmrPipe -fn EM -lb 1.0          \
#| nmrPipe -fn LP -fb                \
| nmrPipe -fn FT                   \
| nmrPipe -out test.ft2 -ov -verb
```

Line 1: invokes the C shell (we will be using the C shell)
Line 2: reads in the 'test.fid' data
Line 3: apodization of the fid using exponential broadening ('EM') of 1.0 Hz
Line 4: linear prediction (commented out, so linear prediction will NOT be performed)
Line 5: Fourier transformation
Line 6: The output (the frequency domain spectrum) will be in a file named 'test.ft2'

A few things to note:
- except for the first and last lines, each line ***ends*** with a backslash ('\'), and ***NOT*** a slash ('/')
- if there are ANY characters OR spaces AFTER the backslash, the macro will not function properly !!!
- blank spaces between the end of the command and the backslash are OK
- the output from the first NMRPipe command is 'piped' to the second using the UNIX/LINUX pipe character, the vertical line (' | '), hence the name 'NMRPipe'
- lines can be commented out using a '#' symbol as the first character in the line
- if the macro is started from the data directory, then it will read 'test.fid' from that directory, and will place the file 'test.ft2' in that directory as well, otherwise, full paths to the file locations must be used

Macros can be written with any text editor. Often the easiest way to create a new macro is to copy an existing macro and editing it. However, NMRDraw has a built-in tool to aid in the generation and editing of NMRPipe macros, which is introduced below.

The NMRDraw ‘Macro Edit’ program is a convenient, menu-driven tool to create and edit NMRPipe processing macros. It is accessed with a **right-click on the ‘File’ menu** then a **left-click on the ‘Macro Edit’ option**. The ‘NMRDraw Processing Macro’ window will appear.



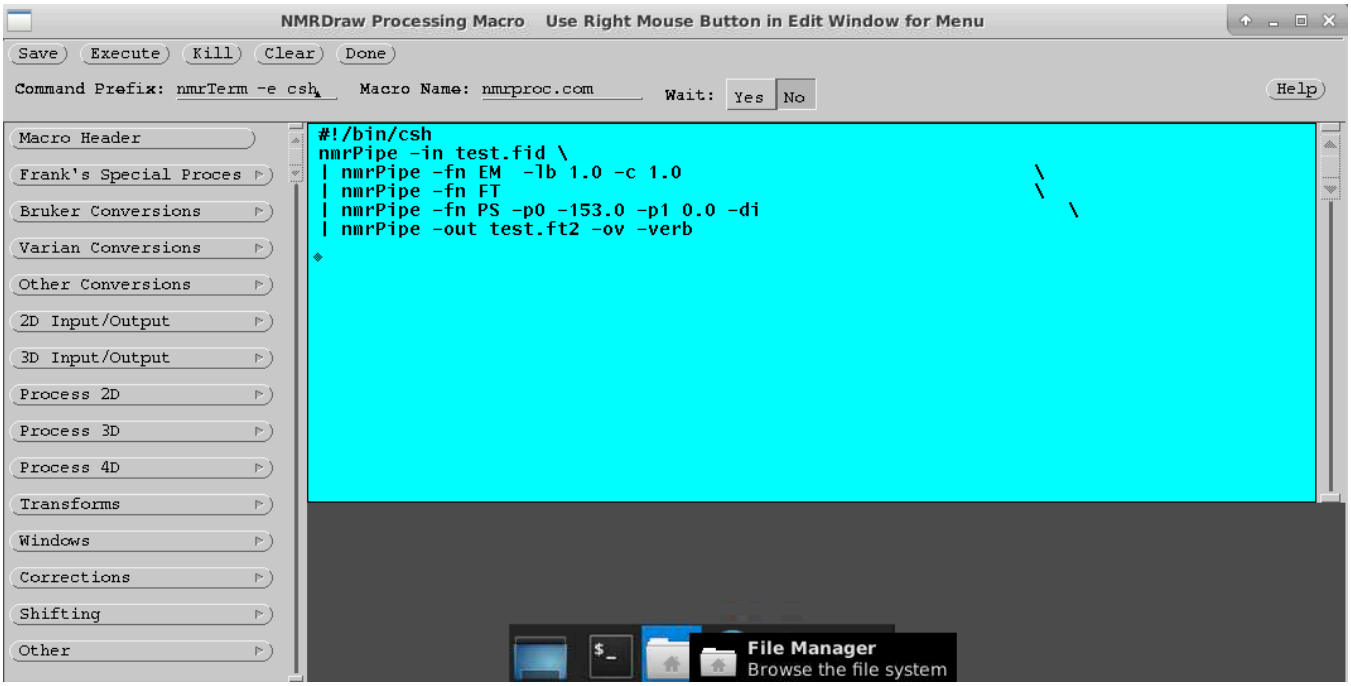
At the top, the ‘Save’ (save the macro to a file, i.e. save your work), Execute (run the macro), ‘Kill (stop a running macro), ‘Clear’ (clear the screen, start over), and ‘Done’ buttons are pretty much self-explanatory. Just below are ‘Command Prefix’ (in the window shown below, indicating that a C shell will be used) and ‘Macro Name’ (the default is nmrproc.com).

On the left-hand side are menus with processing options. It may be necessary to resize this window or use the scroll bar to see the menus at the bottom. To write a macro, options are selected from the menus to build a complete macro.

Here we’ll build a simple macro to process some 1D data. We will assume that the current directory is our data directory, and that our data (‘test.fid’) is in that directory, that the macro we will create will be stored in that file, and that the processed spectrum (‘test.ft2’) will also be stored in that directory. The macro we will be building is shown in the figure on the next page (below).

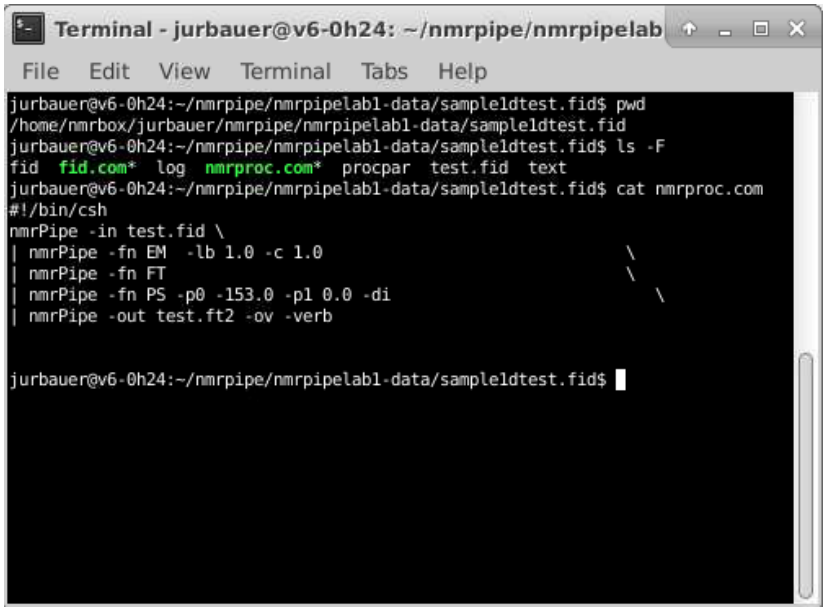
- **Left-click on ‘Macro Header’**. Then **erase everything except the first line** (in the green window, select everything except the first line with the cursor, then delete). This will add the first line indicating we’ll be using the C shell. **Make sure the cursor ends up on the second line at the far left**.
- **Right-click the ‘2D Input/Output’ menu**, then **left-click the ‘Read Fid’ option**. This will add the line that we will be reading the data from a file named ‘test.fid’. Now, **make sure the cursor ends up on the third line at the far left** (you should always make sure the cursor ends up on the next line at the far left).
- **Right-click the ‘Windows’ menu**, the **left-click the ‘Exponential’ option**. This will add an exponential (linebroadening) function of 0.0 Hz (‘-lb 0.0’). You should edit this line so there is 1.0 Hz of line broadening (‘-lb 1.0’). When you are finished, make sure the cursor is on the next line to the far left.
- **Right-click the ‘Transforms’ menu**, the **left-click the ‘Fourier Transform’ option**. This adds the Fourier transform (make sure cursor is on next line, far left).

- **Right-click the 'Transforms' menu**, the **left-click the 'Phase Correction' option**. This adds the phase correction. Change '-P0 0.0' to '-P0 -153.0' to add the proper phase correction for our data. Leave '-P1 0.0' as is (then make sure cursor is on next line, far left).
- **Right-click the '2D Input/Output' menu' menu**, the **left-click the 'Write FT2' option**. This will cause the output to be written to the file named 'test.ft2' that will be stored in the current directory.



- Click the 'Save' button to save the macro. You may get a window that opens saying 'Unable to Store as New file', etcetera. You can disregard this (click the 'Continue' button in this window).
- Click 'Done' to close the 'NMRDraw Processing Macro' window.

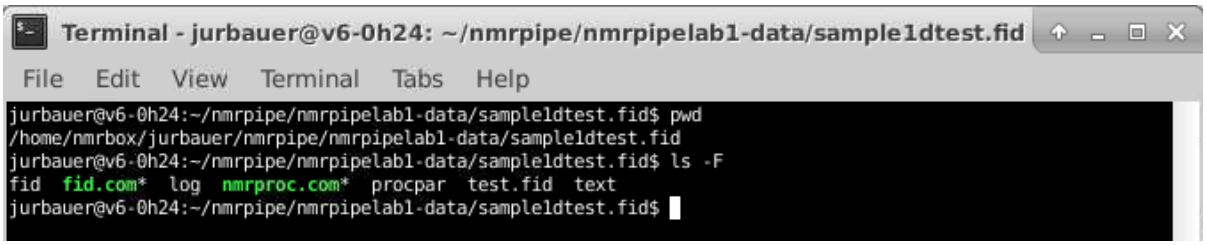
Now, **quit NMRDraw**. If you started NMRDraw from our data directory (/home/nmrbox/username/nmrpipe/nmrpipelab1-data/sample1dtest.fid) then the contents of the directory should include a file 'nmrproc.com', which is executable (has an asterisk when 'ls -F' command is used), and the contents of this file (use 'cat') should be the same as shown in the 'NMRDraw Processing Macro' window above.



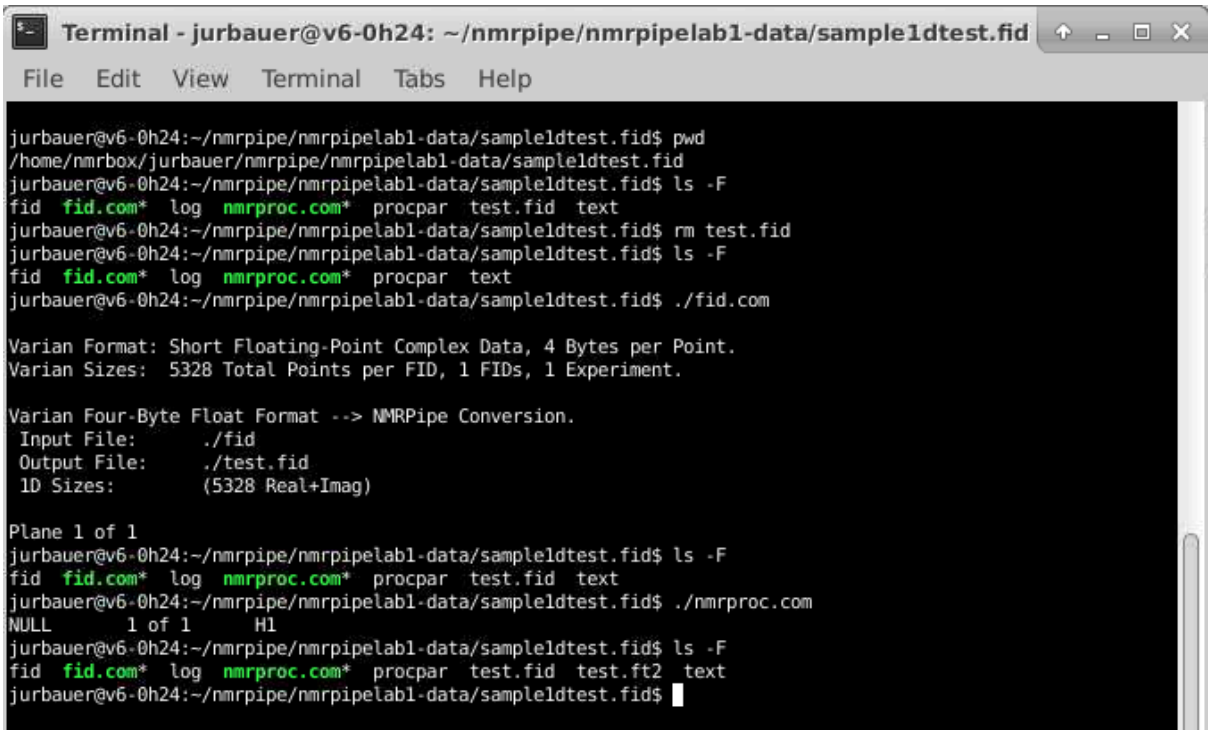
In the next section (below), we'll use the macros 'fid.com' and 'mrproc.com' to convert and process our data from the UNIX/LINUX command line.

Processing with macros: running macros:

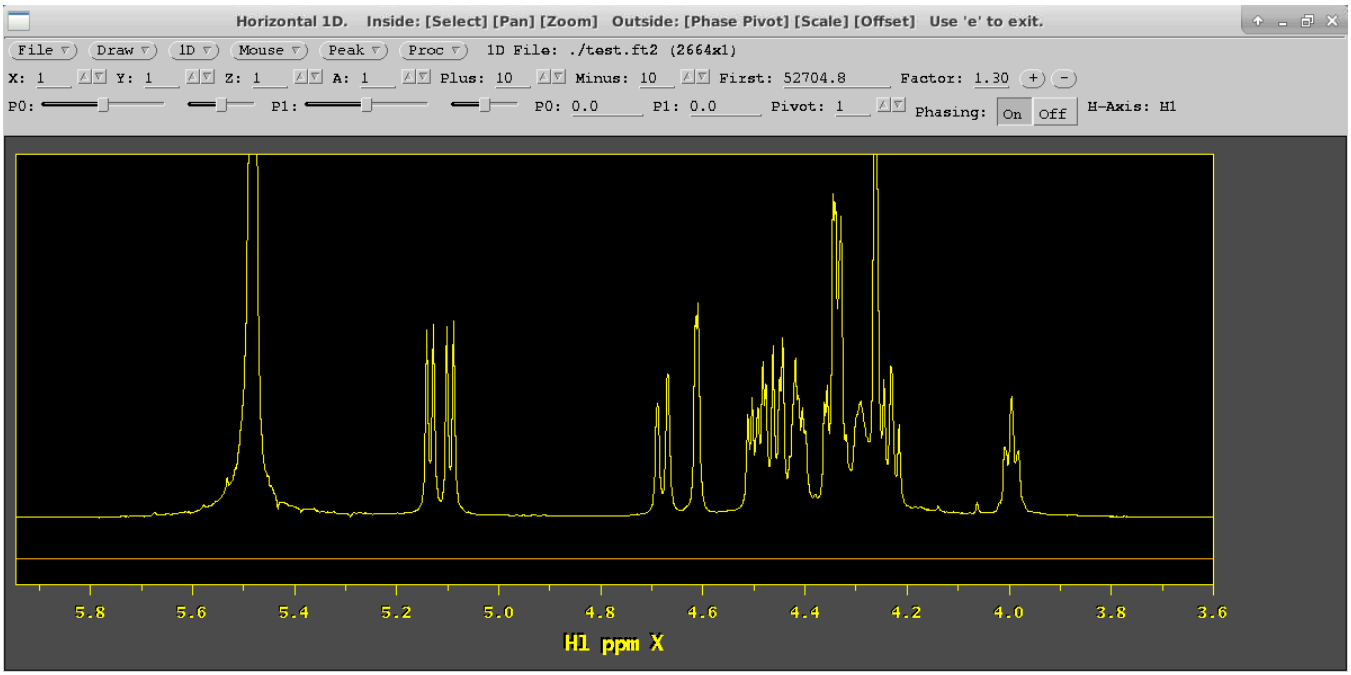
Quit NMRDraw if you've not already. Then, with a 'terminal' window, use 'cd' to go into our data directory (/home/nmrbox/username/nmrpipe/nmrpipelab1-data/sample1dtest.fid). Confirm the contents using 'ls -F'. You should have the four original Varian files ('fid', 'log', 'procpar', 'text'), you should have the converted data file ('test.fid'), you should have the macro that NMRPipe used to convert the Varian data to NMRPipe format ('fid.com'), and you should have the macro we created to process our data ('nmrproc.com'). These latter two files are executable, so they should have an asterisk by them indicating that fact (in this version of UNIX/LINUX on NMRBox, the filenames are also colored green).



Now, delete the 'test.fid' file ('**rm test.fid**'). We will re-create this file using the 'fid.com' executable macro. Then, the re-created 'test.fid' file will be used as input for the executable macro 'nmrproc.com' to create the processed (frequency domain) spectrum. The files are executed simply by the command './filename' (so, '**./fid.com**', then '**./nmrproc.com**'). You'll see the 'test.fid' file has been recreated, and the file 'test.ft2', which is our spectrum, is also now present.



Now that the file with our processed spectrum is created, we can view the spectrum with NMRDraw. **Start NMRDraw** (in our data directory!). NMRDraw will probably exhibit one of the following behaviors. It may directly display the processed spectrum. It may directly display the fid. Or, it may display nothing. Probably it will just display our processed spectrum. If not, right-click on the 'File' menu then left-click on 'select file'. In the top window you should see 'test.ft2' (probably just below 'test.fid'). Just double-click on 'test.ft2' then click 'Done' at the bottom. Or, you can just select 'test.ft2', then click 'Read/Draw' at the bottom (then click 'Done'). In any case, the spectrum will be displayed.



Voila!

**********UNIX TIP**********

Notice that the 'fid.com' and 'nmrproc.com' files have an asterisk next to their names when the '**ls -F**' command is used. This means that the files are executable (macros that can be executed). If not, or if the files don't 'run' when you try to execute them, then the file permissions are probably not set properly. To remedy this, use the unix command 'chmod 755 filename' (so, 'chmod 755 fid.com', and 'chmod 755 nmrproc.com'). They should now be executable, and you should be able to run them as described above.

****

**********Words of Wisdom**********

Life is easy if, when you run NMRDraw, you always start from ("cd into") the directory where your data is. Then, all of your files (for that experiment) will be in that directory, including the macros, final spectrum, and converted data. So, for each experiment, all files for that experiment will be stored in the directory for that specific experiment. This makes it very easy for your work to be reproduced at a later date. As a scientist, you are obliged to ensure reproducibility. Also, this is simple, and there is a lot of elegance in simplicity. Furthermore, after you graduate, and your PI needs to go back and reproduce what you've done, she/he will be very happy that you did it this way. That's very important.
****

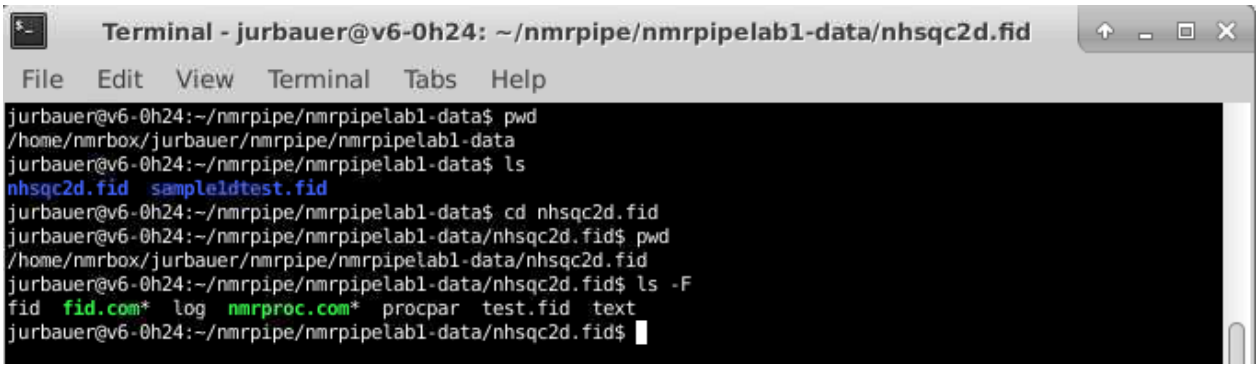
Processing 2D data:

NMRPipe shines when it comes to processing multidimensional NMR data. Here, we will use macros, that have already been prepared, to convert the data from a two-dimensional dataset to NMRPipe format, and then process the data into a two-dimensional spectrum. The dataset is for a ^{1}H, ^{15}N-HSQC spectrum of a small protein molecule, and is named 'nhsqc2d.fid'. The spectrum was acquired on the protein using a 600 MHz Varian NMR instrument. The directly detected dimension (t2) is ^{1}H, and the indirectly detected dimension (t1) is ^{15}N.

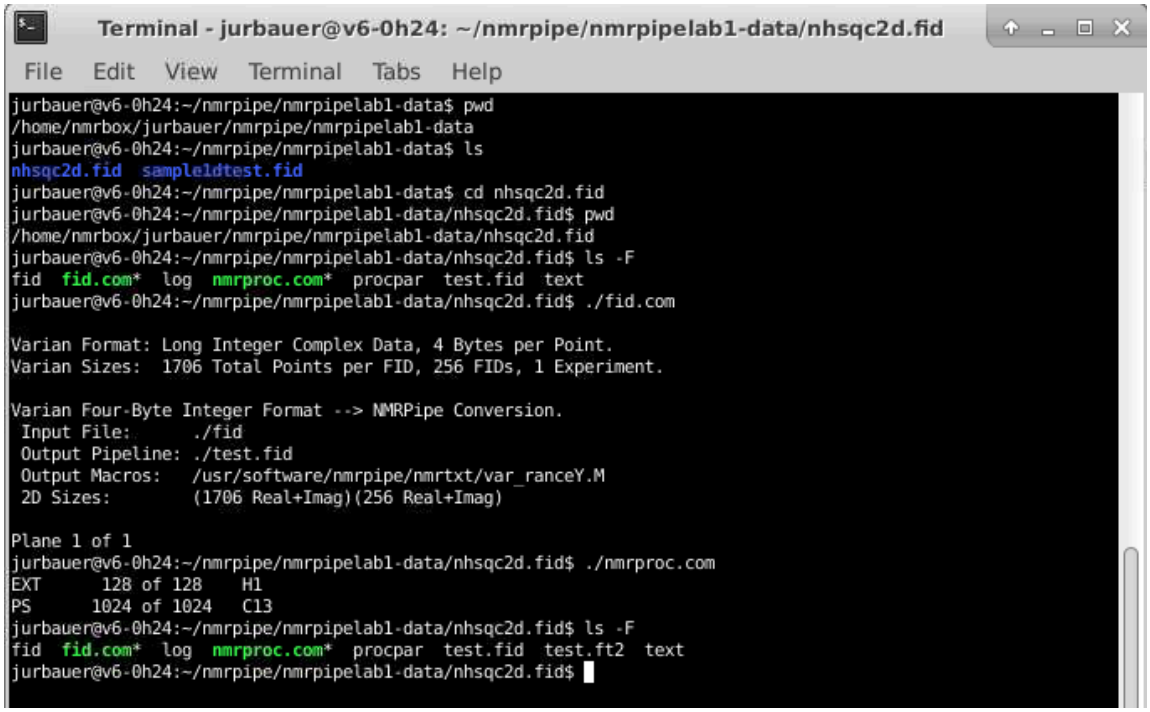
The script for converting the Varian 'fid' to NMRPipe format is named 'fid.com', and the processing macro is named 'nmrproc.com'. The contents of 'nmrproc.com' are shown below, along with some (abbreviated) descriptions of the function of each line in the macro.

```
#!/bin/csh                                          # use C shell
nmrPipe  -in test.fid              \                # input NMRPipe format data/fid
| nmrPipe -fn SOL                    \              # remove large H2O signal at center of spectrum
| nmrPipe -fn SP -off 0.5 -pow 2  \                 # apply 90 degree shifted sinebell (cosinebell)
| 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 (direct) dimension
| nmrPipe -fn EXT -left -verb 2 -sw   \             # 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 to twice the size
| nmrPipe -fn FT -auto               \             # Fourier transform
| nmrPipe -fn PS -p0 -90 -p1 180 -di -verb     \   # apply phase correction in f1 (indirect) dimension
| nmrPipe -fn TP     \                             # transpose back so 1H is on x-axis
 -ov -out test.ft2                                 # write the spectrum to 'test.ft2'
```

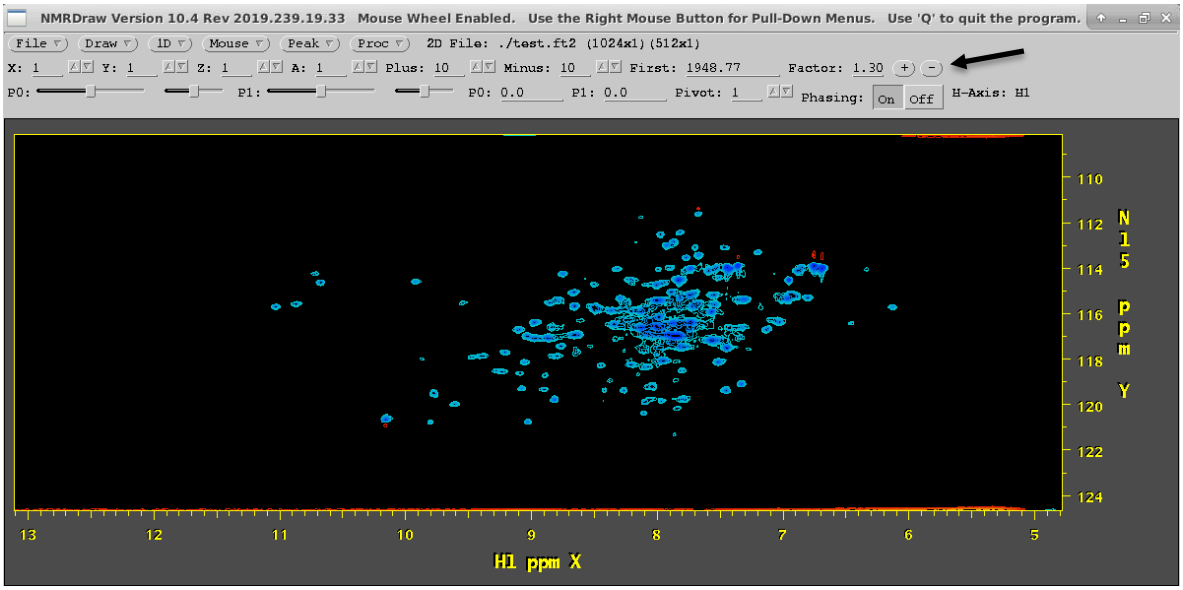
The data for this 2D HSQC data set are in the directory 'nhsqc2d.fid' in the 'nmrpipelab1-data' directory ('/home/nmrbox/username/nmrpipe/nmrpipelab1-data/nhsqc2d.fid'). If you 'cd' into that directory, you'll see the Varian files ('fid', 'log', 'procpar', 'text') along with the data conversion script ('fid.com') and the processing script ('nmrproc.com'). The latter two files each have an asterisk, indicating they are executable.



To convert and process the data from the UNIX/LINUX command line, once you are in the '/home/nmrbox/username/nmrpipe/nmrpipelab1-data/nhsqc2d.fid' directory, simply type './fid.com', and, once that macro finishes, type './nmrproc.com').



You'll now see the file ('test.ft2') that is the processed 2D spectrum. Start NMRDraw from the same ('nhsqc2d.fid') directory. Probably, NMRDraw will automatically plot the 2D data on the screen. If not, **right-click the 'File' menu** and **left-click the 'select file' option**. **Double click 'test.ft2'**, then **click the 'Read/Draw' button** at the bottom (you may have to resize the window to see the buttons at the bottom). **Click 'Done'** to close the window. The data should be displayed.



You can change the intensities of the signals (change the contour level) with the '+' and '-' buttons. Click one of the buttons 4 or 5 times, then left-click the 'File' menu button. Note the intensity changes. We will explore more 2D data processing, display and analysis in the next tutorial.