Custom macros at the University of Michigan.

E. Alvarado, 11/13/08, 01/28/10

****
gradshim - Do gradient shimming
****

Syntax: `gradshim <('H1')>`

Performs deuterium gradient shimming. If an argument is supplied, it performs proton gradient shimming (for non deuterated solvents). Magnetic field, z0, should be at the correct position for the solvent being used.

****
pagelw - Send a plot with variable linewidth to the printer
****

Syntax: `pagelw <(linewidth <, filename>)>`

Like the page command, but allows to change the line width. Linewidth is in printer's points. The default for a printer at 600 dpi is 3 points. To prepare plots for presentations or posters a linewidth of 5 or more will look better. If a filename is given, the plot will be saved as a postscript file in the current user's directory (usually /home/user/vnmrsys/data).

To be used ONLY with postscript plotters. Notice that the actual printer does not need to understand postscript. It is only necessary that it be defined in vnmrj as a postscript plotter and in CUPS as whatever it actually is (for example PCL).

****
setlock - Read shims and lock parameters for the current solvent
****

Syntax: `setlock`

Reads default shims and lock parameter (z0, lock power, gain, phase) appropriate for the current solvent and loads them in the spectrometer's hardware.

****
UMbc - Baseline correction in an array
****

Syntax: `UMbc<(n|'unbc'<,nsubregion<,minpoints<,minregion>>>)>`

Applies a baseline correction (bc macro) to all spectra in an arrayed experiment. Unlike the bc macro, the correction is applied to all the spectra. If order (n) is not indicated, a spline function is calculated, otherwise a polynomial of order n will be calculated and subtracted from the spectra. Integral regions must have been previously defined.

Examples:

```
UMbc(1,200,8,2)
UMbc(5)
```

See `bc` manual page for more details.

****
UMcalcz0 - Calculates correct z0 based on the expected chemical shift of a peak
****

Syntax: `UMcalcz0 (`*`measured_ppm, expected_ppm`*`)`

It is important to be close to resonance when doing gradient shimming, otherwise it may fail. In deuterated solvents you can easily monitor the lock signal and adjust z0 to bring it to resonance, but in non-deuterated solvents this is not possible. This macro can help you set the correct z0 even in the absence of a lock signal.

Take a proton spectrum and measure the chemical shift of a known peak (solvent, TMS, etc.) . Enter the measured and expected chemical shifts for this peak. For example, if TMS appeared at 1.56 ppm:

```
UMcalcz0(1.56, 0)
```

This macro then calculates the correct z0 for this solvent or sample so that TMS appears at exactly the expected shift. Manually change z0 to the calculated value, type su and perform gradient shimming or take a new spectrum.

****
UMdli - Display listed integral values in a array
****

Syntax: `UMdli`

Calculates a list of integrals for each spectrum in an arrayed experiment. Only one parameter can be arrayed (kinetics, diffusion experiments, etc). The macro asks if the list is to be printed, emailed or both. The integrals are saved in files UMdli1.out and UMdli2.out in the current workspace. The list of integrals is also saved in the text file fp.out in a format appropriate to be used with the kinetics analyses macros kind, kini, kinds and kinis and t1 and t2. (The file fp.out is normally generated with the fp macro and contains peak intensities instead of integrals.)

Integral regions must have been previously defined and scaled. For improved accuracy, a baseline correction should have been applied to all spectra.

If the parameter pad is arrayed, the start time of each spectrum is calculated and printed in the report. If the parameter end_time exists, the actual end time stored for each spectrum is also included in the report.

The email option requires sendmail or postfix and mutt.

****
UMdsarray Display an array of spectra in slanted stack whitewashed mode
********** **********

Syntax: `UMdsarray<(tilt <, height <, step>>)>`

UMdsarray will display an array of spectra in slanted stacked whitewashed mode starting with the first spectrum and ending on the last. The step argument can be used to skip spectra when the array is very large.

Arguments:
tilt is the angle (-90 to 90) that the left edge of the stack of spectra will describe against the vertical axis or the left edge of the page.
height is the fraction (0 to 1) of the display's vertical extent that the stack of spectra will occupy.
step is the increment for the spectral index used by dss or dsww when the spectra are displayed. For example, to display every third spectrum use `UMdsarray(30, 0.5, 3)`.

All argument are optional; with no arguments UMdsarray uses default values (30, 0.5, 1).

****
UMexptime - Calculates experimental time
****

Syntax: `UMexptime:$total, $spc, $fid`

Calculates the correct experimental times for 1D experiments in seconds. This macro is applicable only to the s2pul, Presat and wet1d pulse sequences. Up to one parameter can be arrayed. With other pulse sequences the standard vnmrj “time” macro is invoked.

The first return value is the calculated total experimental time of a 1D experiment (including all pad[] or d2). The second return value is the time for each individual spectrum (if ss is negative, nt+ss is used as the number of transients; if ss is positive, only nt is used as the number of transients in the calculation). The third return value is the time for each individual transient (recycle time).

Justification: as of vnmrj 2.1B, the value calculated by the "time" macro is wrong when very short recycle times, like those used in some heteronuclear spectra (Hg199, Pt195, etc) are used. For example, if a spectrum with d1=0.001, at=0.01, nt=10000 is acquired and timed, the time calculated by vnmrj ("time" command) gives an incorrect result. Also, "time" assumes all pad[] have equal values.

****
UMmailpage - Sends a plotted spectrum via email
****

Syntax: `UMmailpage <(filename, email, linewidth)>`

This macro is a complement to the page macro. After a plot has been built with the plot or pl and related commands the UMmailpage macro will send the plot via email to an email address. Two files will be sent as attachments. The first is the plot in postscript (.eps extension) which can be later imported into word processing or presentation documents or converted into pdf format. The second file is an image of the plot in png (portable network graphics) format, which can also be imported into documents or edited with graphics programs. The png file has a resolution of 200 pixels/inch to allow printing and displaying in high resolution.

If invoked without arguments, UMmailpage will ask for a name for the files, the email address where they will be sent to and the linewidth of the plot. Default values will be provided in each case. It is advisable to use linewidth values of 2 to 10 for presentation purposes, as values of less than 2 may be too thin to be comfortably observed from a distance or when reproduced in a publication or thesis page.

Requires pagelw, sendmail or postfix and mutt.

****
UMplcosy - Plot a homonuclear 2D spectrum with high resolution projections
****

Syntax: `UMplcosy <(levels<,spacing<,exp1D>>)>`

Plots homonuclear correlated 2D spectra (cosy, noesy, etc). Same as vnmrj's plcosy (see manual) but corrects several bugs/problems:

1- The intensity of the high resolution 1D projections is proportional to that used in the workspace specified in exp1D.
2- The high resolution 1D projection does not have to have been acquired in the same spectrometer.
3- The high resolution 1D spectrum can have spectral limits lower than those of the 2D spectrum.

****
UMplhxcor - Plot a heteronuclear 2D spectrum with high resolution projections
****

Syntax: `UMplhxcor <(levels<,spacing<,exp1D_H<,exp1D_X>>>)>`

Plots heteronuclear correlated 2D spectra (hetcor, hmqc, hmbc, etc). Same as vnmrj's plhxcor (see manual) but corrects several bugs/problems:

1- The intensity of the high resolution 1D projections is proportional to that used in the workspaces specified in exp1D_H and exp1D_X.

2- The high resolution 1D projections do not have to have been acquired in the same spectrometer. Although this is not recommended, you can plot a 100 MHz C13 spectrum as a projection for a 500 MHz hsqc.
3- The high resolution 1D spectrum can have spectral limits lower than those of the 2D spectrum

****
UMplot - Plot a 1D spectrum with default layout
****

Syntax: `UMplot`

A simple macro to plot a 1D spectrum with scale and short parameter list. If integrals are visible, they will be plotted. If the spectrum is not a proton, peak picking will be printed (select the threshold first).

****
UMpoint - Display the number of the data point closest to the cursor's position
****

Syntax: `UMpoint <(1)>`

Displays the number of the data point closest to the cursor's position. This makes sense only if the fid is currently displayed and can be useful to count data points in the fid, for example in preparation for linear prediction operations, to eliminate spikes in the fid, etc.

Using any argument, e.g. `UMpoint(1)`, is a flag to show enlarged individual data points instead of lines connecting the points when a small portion of the fid is displayed.

****
UMrange - Sets the displayed 1D spectrum's left and right limits
****

Syntax: `UMrange (`*`ppm_1, ppm_2`*`)`

Displays a portion of a 1D spectrum within two chemical shift values. Useful to plot identical expansion of several spectra.

****
UMrange2d - Sets the displayed 2D spectrum's left, right, top and bottom limits
****

Syntax: `UMrange2d (F1_ppm_1, F1_ppm_2, F2_ppm_1, F2_ppm_2)`

Displays a portion of a 2D spectrum within the given chemical shift values. The first two values are for the F1 dimension, second two values are for F2 (the proton dimension in an hsqc). Useful to plot identical expansion of several spectra.

****
UMrtir - Retrieve integral regions
****

Syntax: `UMrtir <(filename)>`

Reads integral regions from a file saved with the UMsvir macro. Normalization scale (ins), integral reference (insref), phase and some other parameters (rp, lp, lvl, tlt, is, io, vs, vp) are also loaded from the file.

Justification. When following a very slow kinetics reaction over a period of many hours or even days, it is common practice to collect the data as a series of independent spectra instead of the "array" of spectra commonly used in kinetics experiments. This macro was created to facilitate the measurement of integrals under identical conditions. After one of the spectra is appropriately processed (phased, referenced, baseline corrected, integral slope and bias are corrected, integral regions are defined and scaled) the UMsvir will save the regions, scaling factor and reference in a file. After processing in a similar fashion other spectra in the series, UMrtir will load the integral regions, scaling factor and reference previously defined so the integrals can be compared. Notice that the new spectrum may require fine adjustment of its phase and integral slope and bias.

****
UMsavetime - Save the current unix time
****

Syntax: `UMsavetime`

Saves the current time in the parameter end_time. The time is saved in posix (unix) format corresponding to the number of seconds elapsed since Jan 1, 1970. For example, Oct-22-2008 13:40:52 is 1224697252. Parameter end_time is created if it doesn't exist.

****
UMsendmail - Sends a spectrum via email
****

Syntax: `UMsendmail <(filename, email)>`

UMsendmail sends a saved spectrum or any other file via email to an email address. In contrast to UMmailpage, this macro sends the actual raw data (the free induction decay, FID) which can be reprocessed with any Varian-compatible NMR processing software. The fid, consisting of several files inside a folder with a ".fid" extension

is compressed into a zip file before being attached to the message. The file can be unzipped with the “unzip” utility in Linux, with “winzip” in MS Windows or with similar utilities in Mac OS. If the file is not an FID but a regular file, it is not compressed before being sent.

If invoked without arguments, UMsendmail will ask for a name for the file and the email address where they will be sent to.

Requires the bash script of the same name, sendmail or postfix and mutt.

****
UMsetupkinetics - Sets up parameters for a kinetics experiment
****

Syntax: `UMsetupkinetics`

UMsetupkinetics helps setup parameters for a kinetics experiment. The macro first asks for the total time available for the experiment in minutes; then it asks for the desired time interval between spectra. Then it calculates the number of spectra and the values of the pad[] parameters. If the numberof steady state scans (ss) is positive, it is converted to a negative number as recommended for a kinetics experiment. Also, autogain (gain='n') is disabled. Current parameters are used to calculate the experimental time for each spectrum.

This macro also creates a parameter, end_time, which will be used to store the *actual* time when each spectrum was finished. The time will be in posix (unix) format corresponding to the number of seconds elapsed since Jan 1, 1970. For example, Oct-22-2008 13:40:52 is 1224697252. Notice that in order to store the end times, the acquisition must be started with au instead of go or ga.

****
UMsvir - Save integral regions
****

Syntax: `UMsvir <(filename)>`

Saves integral regions, normalization scale (ins), integral reference (insref), phase correction and other parameters (rp, lp, lvl, tlt, is, io, vs, vp) in a file. The actual integral values are not saved. Saved regions can be applied to another spectrum with UMrtir .

This is useful to reproduce exactly the same regions in several different spectra, and if the spectra were acquired under identical conditions (nt, d1, pw, gain, etc), and absolute intensity display (ai) is used, integral values can be compared directly between separate spectra.

****
UMtime - Display experimental time or calculate nt
****

Syntax: `UMtime` - display experiment time
`UMtime(`*`seconds`*`)` - recalculate nt for a specific time

UMtime calculates the total experimental time for an experiment taking into account small delays introduced in the acquisition by the hardware. It is designed for experiments with very short acquisition and recycle times, as it frequently occurs with some nuclei, where vnmrj's time macro is inaccurate. In kinetics experiments, where the pad parameter is arrayed, it calculates the total experimental time including the pre-acquisition delays ("time" does not). In inversion recovery experiments, where d2 is arrayed, it calculates the total experimental time including all d2.

`UMtime(`*`seconds`*`)` recalculates nt so that the acquisition time for a single fid (pad not included) is approximately the requested time in seconds.

UMtime was designed only for the 1D experiments mentioned above. If the pulse sequence is not "s2pul" the regular vnmrj time macro is called. Arrayed parameters other than pad or d2 are not considered in the calculation.

****
UMunarray Extracts the fids from an array of 1D fids
****

Syntax: `UMunarray<(filename)>`

Extracts the spectra of an array and writes them as separate spectra. The original file is preserved. A folder named 'filename' will be created and the new files will be written inside. A numeric suffix (_fid001, _fid002, etc) will be appended to the filenames. If no filename is provided, vnmrj prompts for one.

This macro allows separate processing of the component spectra of an arrayed experiment. Not suitable for 2D spectra.

For diffusion, kinetics, variable temperature, T1 or T2 experiments. Only one parameter must be arrayed.

****
UMviewman - Display a manual page
****

Syntax: `UMviewman<(name)>`

Displays the manual page of a macro, command, pulse sequence or help file in a graphics window. The paths selected in the user's applications preferences (for example, userdir/manual, /vnmr/manual, etc) are searched in sequence for a file specified by 'name'. When no argument is used, it displays the manual page of the current pulse sequence (`seqfil`).

Example: `UMviewman('gHSQC')`