The ssNake reference manual

Wouter Franssen & Bas van Meerten

Version 0.5b (UNDER CONSTRUCTION)

10th July 2016

Contents

1   Running ssNake

1.1   Python versions

ssNake had been programmed to run on both the python 2.x and 3.x branch. We developed ssNake using python 2.7.9 and 3.4.3, but somewhat older version might also be sufficient.

1.2   Numpy versions

We have run into issue while using older versions of numpy. It seems that version 1.7.0 is the oldest that we support (some new commands have been introduced in this version, that ssNake makes use of). Development took place on version 1.8.2.

2   File

Using the ssNake load tools is quite easy. Go to File → Open. Navigating and double clicking on the desired file then loads the data. Many formats supported by ssNake (like Bruker and Varian) have their data in a folder, in which several files with a fixed name are present. For these formats, loading any of the files in this directory is accepted (ssNake searches for the expected files in the selected directory). Loading large data files might take some time, depending on your computer hardware.

When loading the data, the user is prompted to give in a name for the data by which it is identified in the workspace selector (see the Worspaces section for more on this). The default name is the name of the file (or folder for data in which the file name is fixed). If the name already exists as a workspace, the name spectrum$n$ is used (with n the lowest integer that is not in use).

2.1   Formats

ssNake is able to extract NMR data from a number of formats from several vendors. The following table summarizes the support.

Name	Specification
Varian/Agilent	VnmrJ 2 and newer. fid and data (spectrum)
Bruker	Topspin and XWinNMR (fid & ser,1r/i & 2rr/ii)
Chemagnetics	???
Magritek	Both 1D and 2D data
Simpson	1D and 2D, -ascii and -binary
JSON	ssNake JSON file (ascii)
Matlab	ssNake .mat file (binary)

Generally, ssNake loads the data raw data and searches for some variables (e.g. spectral width and spectrometer frequency). The above data formats do not support data with more then two dimensions in a nice way. Getting the higher dimensions is therefore not straightforward. ssNake treats this data as 2D, after which you can split the data yourselves using Matrix → Split.

Varian/Agilent

Loading a Varian/Agilent fid file requires a *procpar* file and a *fid* file. From the *procpar*, the spectral frequency and the spectral width in one or two dimensions is extracted. The *fid* file is then checked for the version (VnmrJ 2 or 3) and the data is extracted from this file.

For processed data (*data*, in the datdir folder for example) it searches for a *procpar* file in that directory or in the parent folder, from which additional parameters are extracted.

Bruker

The Bruker load first checks for the *acqus* and *acqu2s* files and extracts the number of points in both dimensions, the spectral widths and the spectral frequency. Also, the bitorder is checked, which describes the way the data is saved in the *fid* or *ser* file. Using this, the data is extracted.

Bruker processed spectra can be loaded from *1r* and *1i* files for 1D data, *2rr* and *2ii* for 2D data, and *2rr* and *2ir* for hypercomplex data. It additionally takes

parameters from *procs* or *proc2s* files in that directory. Moreover, the spectral frequency is extracted from the *acqus* and/or *acqu2s* files located *two directories up*.

Chemagnetics

Loading Chemagnetics data requires a *acq* (and *acq_2* for 2D) and *data* file. From the former, the number of points in both dimensions is extracted, the spectral widths and the spectral frequency. All the data points are then extracted form the *data* file, and reshaped to a 2D array when necessary.

Magritek

Magritek requires two or three files: *acqu.par* for the spactral variables, and a file that ends with *.1d*. For 2D data, a file that ends with *.2d* is also needed. As with the other formats, the number of data points in the two dimensions is extracted from the parameter file (along with the spectral data) and the binary data in the *.1d* or *.2d* is unpacked.

Simpson

Simpson data files are plain text files, *.fid* or *.spe* as file extension. From the file, the number of points in both dimensions is extracted, and the spectral widths. The spectral frequency is *not* included in the Simpson format, and is put at 0. Simpson files cannot contain a mixture of time and frequency data (both dimensions must be the same type). Simpson binary data is also supported, but not raw binary.

JSON

Loads a JSON file that is saved with ssNake. File must have the *.json* or *.JSON* extension.

Matlab

Loads a Matlab file that is saved with ssNake. File must have the *.mat* or *.MAT* extension. Also supports .mat files that have been resaved to matlab v7.3 format.

Unsupported file formats

Currently, we support all the data formats that we have access to. While we have some code lying around for loading JEOL data, for example, we cannot include this if we cannot test it. If you want your favourite data format to be support by ssNake, please send us a request along with some sample data (1D and 2D).

2.2 Open

When using ssNake's load function, ssNake analyses the selected file name and the names of other files in the directory to figure out the format you want to load. ssNake only needs to check for files that are actually needed for loading the data, so removing useless files from the format does not pose a problem. Below, the checks used by ssNake in this function can be found for each format.

Name	Folder contains files	File extension
Varian/Agilent fid	procpar and fid	
Varian/Agilent spectrum	data and procpar or ../procpar	
Bruker fid	acqus (and acqu2s for 2D) and fid or ser	
Bruker spectrum	(1r/1i) or (2rr and 2ii/2ir) with procs/proc2s and ../../acqus and ../../acqu2s	
Chemagnetics	acq (and acq_2 for 2D) and data	
Magritek	acqu.par and *.1d or *.2d	
Simpson		.fid or .spe
JSON		.json or .JSON
Matlab		.mat or .MAT

2.3 Save data

Naturally, ssNake can also be used to save data. When saving data, the current ND data is saved, along with the spectral widths, frequencies and if the axis is in time or frequency units. Also, ppm references are saved, along with the processing history. Undo information is *not* saved.

JSON

JSON (JavaScript Object Notation) is a ascii format (i.e. regular text) used to save data structures. Within ssNake this can be used to save the current data in

a clear, human readable format. As JSON data is in ascii, it is not efficient in file size and in speed. If these are necessary then consider saving the data as a matlab binary.

MATLAB

The MATLAB binary format is a open source format in which many different types of data can be saved. Also, it is the native format of MATLAB, which can be used for more special data manipulation, if necessary. As the format is binary, it is efficient in both speed and file size. The content of the file is the same as the JSON file.

2.4   Export data

Using ssNAke you can also export your data. Exporting never saves all the info our own formats do, so there will be loss of information. It is suggested you only use these formats for exchanging data to another program, but not for archiving.

Simpson

Simpson is a general NMR simulation program with its own data format. The format only supports 1D and 2D data, and only if the data in both dimensions is of the same type (i.e. both spectrum, or both time). Apart from that, Simpson is a ascii format, and therefore has the same speed restrictions as the JSON format. The parameters that are saved within the Simpson data are restricted (no offset frequency), so when loading it back into ssNake some information is lost. We therefore advise to use the Simpson format only for transferring data to another program.

ASCII (1D/2D)

This option saves the current data to a bare ascii format (text file). The first column is the axis in units of Hz or seconds. The next columns are alternating real and imaginary for all traces in the 2D data. For 1D data, only a single set of real and imaginary data is printed. The axis of dimension 1 of the 2D data is not saved.

Figure

The Figure option can be used to export any Figure plot within ssNake. The title, axis labels and size of the Figure can be changed. Be aware that choosing a to small height can cause the x-axis label to disappear. After all the titles have been set, pushing 'Save' will save the Figure to a directory of your choice. The Figure can be saved in vector format (.eps, .pdf, .svg), in lossless pixel format (.png), and in lossy pixel format (.jpg). We advice to use vector formats to make high quality pictures.

2.5   Quit

Closes the ssNake application.

3   Workspaces

Within ssNake you can have multiple data files opened. Each loaded file gets its own environment in which you can process the data. These are called workspaces. All ssNake workspaces are independent, meaning that the data from workspace 1 cannot be changed while viewing any other workspace. Going back to a workspace from another goes back to the same view as before: nothing is discarded when switching between workspaces. Also: every workspace has its own undo/redo information. Some tools can transfer data from other workspaces to the current workspace. You can find more about these tools later on, in their respective sections.

When loading data, ssNake prompts for a name. This will become the name of the workspace in which the data is loaded. The title of the graph will be set to this name, and the name will be set in the list of opened workspaces in Workspaces → Active. You can also switch workspace using the tabs that are created for every workspace. Note that you cannot have multiple workspaces with the same name.

3.1   Duplicate

Makes a copy of the current workspace to another workspace, of which the name is asked. All the data and the current view is copied to this new workspace. No undo and redo data is transferred. Also, the current workspace is now the new, duplicated one.

3.2   Delete

Deletes the current workspace and all references to it. The view is changed to the next workspace in the list. Be aware that there is no way to undo this!

3.3   Rename

Renames the current workspace. Note that two workspaces cannot have the same name.

3.4   Active

Shows a list of all workspaces. The current workspaces can be changed by clicking on another workspace name.

3.5   Keyboard shortcuts

ssNake features several keyboard shortcuts to work with the workspaces. Below is a list of all the combinations.

Combination	Effect
Ctrl + w	Close current workspace
Ctrl + d	Duplicate workspace
Ctrl + Page Up	Change current workspace to the one above in the list
Ctrl + Page Down	Change current workspace to the one below in the list

When the current workspace is the last in the list, and Ctrl + Page Down is pressed, the current workspace is set to the first in the list.

4   Macros

Macros can be used to save a particular series of action performed on the data, for later fast reuse. Say that you want to load several data files, and want to set the size of them to 4096 points and perform a Fourier transform. Doing this many times can be annoying. When you create a macro to do this, you only need to execute the macro for every data file. Thus saving time and making sure all are processed in an identical way. Also, macros are shared between workspaces: there is only one pool of macros. How to use macros is explained below.

4.1   Start recording

To create a macros, you must tell ssNake that you want to record a series of action, to be saved in the macro. Executing Macros → Start Recording prompts for a name of the future macro, and enables the recording. From now on, all the actions you perform on the data are recorded and saved in this macro.

4.2   Stop recording

After some actions have been performed (Fourier, zero fill, phasing...) the recording of the macro can be stopped using Macros → Stop Recording.

4.3   Run

Can be used to run the selected macro on the current data.

4.4   Delete

Delete a macro.

4.5   Save

Can be used to save a macro to the disk as a .json file. This is a plain text file, in which you can see all the steps that are executed when running this macro.

4.6   Load

Loads a saved macro in ssNake. Execution of the macro can then be performed using Macros → Run.

4.7   Notes on usage of macros

Note that ssNake only records the actions that are not undone. If you perform an action when recording is on, and you undo it while recording, the macro has no entries. However, performing two actions that cancel each other *are* recorded. For example, issuing Fourier twice has no net effect, but a macro of this includes both commands.

SOME PART ABOUT RUNNING MACROS ON MORE OR LESS DIMENSIONS THEN THEY WERE CREATED FOR.

5 Edit

5.1 Undo

In ssNake, every action that causes a change in the data can be undone. When such an action is performed, the undo list is increased with one entry, in which the information is stored as how to undo the performed action. Sometimes this involves the inverse operation (e.g. for fft/inverse fft), but for some actions the old data must be preserved and put back when an undo is requested. Duplicating data to another workspace does not copy the undo list.

Undo cannot be used the undo the current view to the previous view as this is not an operation that changes the data.

5.2 Redo

When Undo is used, the action that is undone is put in the redo list. Using Undo and then Redo therefore performs no net action. Every time a new action (i.e. something other then redo) is performed the redo list is cleared.

5.3 Reload

This action reloads the data from the original source. All undo and redo information is retained., so you can undo back to before you reloaded the data. Note that the undo information can take quite some memory on your computer, especially for bigger data sets.

6 Tools

This section explains all the options included in the Tools menu.

6.1 Real

Real puts the imaginary values of all data points at zero.

Background

In some NMR experiments, only the real value of the magnetization during a specific period is measured. In this case it is convenient to zero the imaginary part to force further processing to ignore this data.

6.2   Imag

Imag puts the real values of all data points at zero.

6.3   Abs

Abs replaces all complex datapoints by the length of the vector they span in the 2D complex plane. That is:

$$F_{\text{new}} = \sqrt{\text{real}(F_{\text{old}})^2 + \text{imag}(F_{\text{old}})^2}$$

for all points in the complex data. These values are put as the real part. The imaginary part is zeroed.

6.4   Phasing

Opens a tool that can be use to change the phasing in the current spectrum or fid. Supports zero order, first order and autophasing.

Background

NMR data is usually recorded as complex data. Here the real part can be said to be the x-magnetization, and the imaginary part the y-magnetization. This gives the NMR signal as a complex vector, that is rotating in the complex plane during the signal acquisition. An ideal NMR signal consists of a cosine in the real part, and a sine in the imaginary part. This gives, after Fourier transform, a spectrum were a pure absorption lineshape (i.e. a peak) is seen. Depending on the way the time signal is recorded, there can be a distortion to this line, as a constant phase is added to both the sinusoids. With zero order phase correction, an additional phase is added to both the complex and the real part, to adjust for this effect. When expressed in complex exponentials, the following holds:

$$f_{\text{new}} = f_{\text{old}} \cdot e^{i\pi \cdot \theta / 180} \tag{1}$$

were f is a complex value in the time or frequency domain and θ the desired phase correction in degrees. When using this zero order phase correction, all points in the time or frequency domain get the same phase correction.

In the spectral domain, a *first* order phase correction is also defined. This leads to a phase correction that linearly depends on the frequency offset with respect to a set central frequency.

ssNake implementation

Selecting phasing in the tools menu opens the phasing window. In this windows, the zero and first order phasing can be set, as well as the reference frequency (centre point) for the first order phasing. The values can be either filled in the boxes, changed by pushing the left and right buttons, or set by dragging the slider to the left or the right. For zero order phase correction, the limits are -180 and 180 degrees. For first order correction, these are -540 and 540 degrees. Higher values for the first order correction sometimes make sense, and can be filled in the box directly. Note that the values for the zero order correction are circular, so that -180 and 180 degrees lead to the same effect.

While in this menu, the effect of a certain phasing can be immediately viewed in the ssNake window. When a good setting is reach, 'Ok' can be pushed to apply the phasing. Note that for multidimensional data, the preview only shows the current trace. Pushing 'Ok' the applies the same phasing correction on all the traces (might take some time for large data). If the phasing should only be applied on the current spectrum or fid, the box 'Single slice' can be ticked. Note that you still have to push 'Ok' to apply the phasing.

Apart from manual phasing, ssNake also supports two types of autophasing: zero order only, and zero and first order. When pushing these buttons, ssNake will search for the best phasing in the *current* trace, with only first order phasing correction, or with both zero and first order. Do note that the setting that ssNake finds might not be the best: phasing an NMR spectrum is tricky and has no unique mathematical solution. However, for a fast analysis it can be useful.

Also, note that first order phase correction has no meaning in NMR when applied to the time domain. ssNake therefore always applies first order phasing in the *frequency* domain. When first order phasing is applied when viewing the time domain signal, ssNake transforms to the spectral domain, does the phase correction, and transforms back. As Fourier transforms are quite fast, this provides no issues when viewing the effect of the phase correction.

6.5   Swap echo

A tool to swap the time domain data symmetrically around a selected point.

Background

In some NMR experiments, the rise and fall of a spin echo is recorded. One way of processing this data is to remove all the data points before the echo maximum, and subsequently treat the data as a regular FID. In some cases it is more useful to use the full data, rise and fall, of the echo signal. Directly Fourier transforming such data leads to massive first order phase distortion. This can be circumvented by swapping the echo around the centre. In this case, the data is split in two parts: the data from the start of the measurement until the top of the echo, and the data from the top of the echo until the last point. The latter data is put leftmost, the former right. The new signal then starts with a decay and rises again at the end. Due to the symmetry of the echo, the Fourier transform of this signal will have zero imaginary signal, while the real signal is still the same as if a regular FID is transformed. This means that phasing such a line can be easily done.

ssNake implementation

Echo swapping is done by left clicking on the echo maximum of the time signal. By default, the swapping is set at the centre of the time signal. After left clicking, a preview is shown. Pushing 'Apply' applies the transformation to all traces of this dimension. Also, the tag 'Whole echo' is set in the footer. When this is on, Fourier transforming does not multiply the first point by 0.5 (as is needed for data that decays to zero), and line broadening is applied symmetrically around zero.

6.6 Offset correction

Subtracts the average value of the selected data points from all data points.

Background

It can sometimes happen that NMR data does not decay to zero, but to another value. In this case, a time signal with zero frequency and no decay can be thought to have been added to the NMR signal. Usually, this is due to issues with the electronics, and does not imply a NMR signal with these properties. For proper analysis of the signal, this offset should be removed, as it leads to a signal at 0 Hz, which can interfere with the spectrum. This can be solved by

subtracting the average value of the last points of the time domain data from all the data points.

ssNake implementation

Using the Offset correction tool opens a new window. By default, the last 20% of the time data is selected as containing only noise (of which the average is the offset which should be subtracted). Left clicking two times in the graph sets the limits between which the average value is calculated. Pushing apply subtracts this value from all time data points. If the data has more dimensions then 1, the average value is subtracted from all data points. Take note that the average is only calculated for the current graph. If you want to correct the offset for each graph you can select the 'Single slice' option, and do the correction separately in each desired trace.

6.7   Baseline correction

Fits a polynomial line of a given order through the current spectrum while ignoring the selected parts. This line is then subtracted from the spectrum.

Background

In some NMR spectra, experimental conditions were such that a non-informative baseline is present in the spectrum. This can be due to electronic reasons (e.g. pulse ringing) or due to background signals. This usually means that the first points in the time signal are corrupted. In some cases, removing the first data points is the best option, but this leads to signal loss and a large first order phasing error. In these cases, it can be more convenient to try to subtract the errors in the frequency domain. This is done by fitting a nth degree polynomial through the spectrum, and subtract this line from the spectrum. For this, it is essential that the regions of spectral information (i.e. the NMR peaks) are not included in the fit. Failing to do this will cause the baseline correction to attempt to remove the peaks.

Baseline correction is mostly a visual thing. In the case of ssNake, the degree of the polynomial to be subtracted has to be put in. Usually, putting a very high value leads to strange results. This, however, depends on the spectrum to be corrected.

ssNake implementation

In ssNake, baseline correction is performed by fitting a nth degree polynomial through the spectrum, i.e.:

$$S_{\text{corr}} = \sum_{i=0}^{n} a_i x^i,$$

with a_i the amplitude of the ith degree polynomial. Fitting this to the spectrum is a mathematical operation with a single unique solution. Prior to doing this, specific parts of the spectrum can be selectively ignored in the fitting procedure. This can be done by left clicking in the spectrum at the two limits of the area that is to be ignored. The area is then marked with a red background. This way, multiple areas can be selected. Pressing 'Fit' then performs and shows the fit. Apply subtracts the line from the spectrum. Note that by default this single fit is subtracted from *all* the traces. This is useful if the background is identical in all of them. Alternatively, the box 'Single slice' can be ticked to only apply the correction on the current spectrum.

6.8   Subtract averages

Subtracts the average of a region from the respective data. The analysis is performed independently for every slices in the nD data. This effectively does the same as Offset Correction, but then for each trace independently.

ssNake implementation

Opening this tool gives a window were the left and right limit of the selected data can be set. Alternatively, left clicking on the plot sets the start limit on the first click, and the end limit on the second click. Pushing apply performs the subtraction.

6.9   Reference deconvolution

Uses the FIDDLE algorithm to deconvolute a peak shape in the spectrum.

Background

Due to issues with magnetic field inhomogeneity (i.e. shimming), peaks in an NMR spectrum can have a weird shape with extra ridges and small peaks on the side. In some cases it is known from prior knowledge that the visible peak

should be a single Lorentzian line. If so, the line can be deconvoluted to give this Lorentzian line. Naturally, all other lines in the spectrum should have the same distortions. Using the known shape of the easy Lorentzian reference, all other lines can also have their shape corrected.

ssNake implementation

Using the 'Reference deconvolution' opens a window, in which the region of the reference signal has to be set (by left clicking, or by filling in the values). Additionally, the new linewidth of the peak has to be specified. Note that the algorithm cannot increase resolution, only make better lineshapes. Filling in a too small linewidth will result in a heavily distorted spectrum. Pushing apply performs the analysis for all traces.

6.10   Correct Bruker digital filter

Corrects a Bruker fid to start at $t = 0$.

Background

In theory, the recording of an NMR signal starts directly after the excitation point. The signal can then be easily described by a decaying exponential (or another function) and the Fourier transform does not show surprises. NMR data measured on a Bruker device, however, has the peculiar attribute that it starts *before* the physical start of the signal. The origin of this lies in the way Bruker spectrometers use their digital filter, of which the details will not be described here. Direct Fourier transformation of this signal will lead to massive oscillations in the baseline of the resulting spectrum.

To be able to handle Bruker NMR data, this detection error has to be corrected. The preferred way for this seems to apply a strong first order phase correction in the spectrum that (due to Fourier principles) rolls the time domain data in such a way that the first point is now the start of the real signal. Due to this procedure, the points that were at the front of the time signal are now at the end. Apparently this leads to better results than just removing this data.

ssNake implementation

To automate this process, ssNake uses information from a handy text by W. M. Westler and F. Abildgaard. Every Bruker NMR spectrometer has its own

characteristics, and therefore needs a different amount of first order phase correction. For older Bruker hardware, the amount of correction is found by extracting two parameters from the Bruker acqus file, and to use a lookup table supplied by Westler & Abildgaard. For newer hardware, this value can be read directly from the acqus file.

Using this tool opens a windows, in which you have to open (again) the current Bruker file. ssNake then finds the desired phase correction and applies it to all traces.

7   Matrix

This section describes the ssNake matrix manipulation tools. NMR data is nothing more then a data matrix of a specific dimension. Using the matrix tools, you can change the size of a dimension, split the data into more dimension, integrate, multiply, mirror etc. In the following, the different supported matrix manipulations will be introduced.

7.1   Sizing

The sizing tool can be used to edit the number of points in the current dimension.

Background

The number of point taken in an NMR time signal is usually restricted, as increasing the acquisition time leads to more noise (if the signal has decayed) or adds too much strain on the device (if decoupling is used). As the number of points in the spectrum is equal to the number of points in the time signal, this limits the *digital* resolution of the spectrum, making it less easy to extract the desired data. However, if the signal has decayed at the end, the digital resolution in the spectrum can be enhanced by adding zeroes at the end, without effecting the spectrum in terms of signal-to-noise and appearance. This is termed 'zero filling' and is often used in NMR.

If the time signal has not fully decayed, the signal abruptly changes to zero when zero-filling is applied. This leads to distortions in the spectrum (i.e. sinc wiggles) as the time signal is effectively multiplied with a block function. General Fourier theory then states that the spectrum must then be convoluted with the Fourier transform of the block function, which is $\sin(x)/x$, i.e. sinc.

ssNake implementation

Selecting 'Matrix → Sizing' creates a sizing window. In here a single number
has to be filled in the box titled 'Size'. Decimal values will be rounded to the
nearest integer. Apart from numbers, the command 'k' (or 'K') is also supported,
which stands for '1024'. When using this definition there has to be a number
before the 'k'. A valid input would therefore be: '2k'. Which sets the number of
points to $2 \cdot 1024 = 2048$. Pressing 'Apply' executes the sizing command on all
traces in all dimensions.

If the supplied number is larger than the current size, zeroes are added at the
right-hand side of the time signal. If the value is lower, points are removed from
the right-hand side of the time signal. If the whole echo type is on, the zeroes
are added in the centre of the time signal, or points are taken symmetrically
from the centre, if the size is reduced. Alternatively, the position of the zero
filling can be specified by left-clicking in the spectrum, or by manually filling in
the desired offset value in the box 'Offset'.

When applied to the frequency domain, zero filling makes no real sense.
Therefore, ssNake always applies it in the time domain. So when the size is
chaed within a spectrum, it is transformed back to the time domain, the size is
changed, and a forward Fourier transform gets the data back to the frequency
domain.

7.2   Shift data

Shifts the data to the left or to the right, by removing data points on the left or
on the right, and filling with zeroes on the other side. Negative shifts are left
shifts, positive right shifts.

Background

Data shifting can be used if some data points on the left or right of the time
signal are unnecessary or unwanted. When measuring a spin echo, for example,
it is not uncommon to start recording the signal some microseconds before the
expected echo top, to make sure the full signal is measured. The data can then
be left shifted to remove the data points before the echo maximum, and yield a
regular spectrum.

ssNake implementation

Starting the shift data tool creates an input window. Within this window, a integer number must be filled in that equals the number of data points of the desired *right* shift. Left shifts can be made by using negative values. Alternatively, the arrows on the right and left side of the input box can be used to shift the data in the direction of the arrow. The effect of the operation on the current 1D can be seen in the main window. When pushing 'Apply' the shift is performed on all traces of the nD data.

ssNake can also perform the time domain shift from within the frequency domain. When in a spectrum, an inverse Fourier transform is used to go back to the time domain. The data is then shifted with the desired measure, and a forward Fourier transform gives the spectrum of the shifted time signal. The effect of the sift on the current 1D can be seen in the main window. As this calculation is a bit more involved, it can take some time if the data set is large.

7.3   Integrate

Integrates a specific part of the current dimension on all traces.

Background

As NMR is a quantitative technique (when applied correctly), the area under the different peaks in a spectrum is a representative for the relative number of nuclei that resonant at this frequency. Calculating the integral of these peaks can therefore supply the user with an indication on the composition of the material that is analysed. Integration in ssNake using the matrix tool integrates across (specific parts) of the current dimension. Therfore, the resulting data will have one dimension fewer than before. Say, for example, you have a set of 2D data, of which every trace has the same signal, but a different intensity of this signal. When in the second dimension (in which the peak is shown) the peak can be integrated by supplying the left and right limit of the peak. This will result in 1D data, of which every point represents the integral of the selected area of one of the spectra.

ssNake implementation

When using the integration matrix tool, a window pops up, were you have to select the limits of the area you want to integrate in the current dimension. This

can be done by typing in the numbers of the data points, or more conveniently by left clicking in the spectrum. The first click sets the first limit, the second click the second limit. Pushing apply then performs the integration. Be aware that integration values can depend on, amongst others, the amount of zerofilling that is used.

7.4 Sum

Sums a specific part of the current dimension on all traces. Is implemented in the same way as integrating, only no scaling with the step sizes is performed.

7.5 Max

Takes the maximum value of the selected part in the current dimension on all traces. Implementation is the same as with integration.

7.6 Min

Takes the minimum value of the selected part in the current dimension on all traces. Implementation is the same as with integration.

7.7 Max position

Returns the x value of the maximum position (i.e. top of a peak) within the selected area for all traces.

Background

When analysing some experiments, it is useful to make a plot of the position of a peak versus some changing variable (e.g. time, temperature). This features allows the user to extract such data. It finds the maximum value within the selected region, and extracts the corresponding x-value. It does this for all traces, and therefore reduces the number of dimensions by 1. For example, if one has 2D data, of which the most intense peak in D2 shifts linearly in D1, this tool tool can be used to generate 1D data that shows this linear relation.

7.8 Min position

Returns the x value of the minimum position within the selected area for all traces. Works the same as Max position.

7.9 Average

Returns the average of the selected region. Implementation is the same as with integration.

7.10 Diff

Differentiates the data in the current dimension (i.e. returns the difference between each data point and the next data point). This is equal to the slope between the points. Remember that NMR data is the frequency domain is always depicted from right to left, so the direction of differentiating is different in the time domain than in the frequency domain (or, in fact, it is the same but *looks* different).

7.11 Cumsum

Performs a cumulative sum on the data in the current dimension. This means that each point is now equal to its old value, and those of all the previous points added. Remember (as with Diff) that the frequency domain is mirrored, and the direction of Cumsum is also different.

7.12 Extract part

Deletes all parts of the current dimension that are outside the selected region.

ssNake implementation

This function can be used to delete parts from the current spectrum or fid. Left clicking on a prt sets the first limit on the first click, and the second limit on the second click. Apply deletes all parts outside this region on all traces. When doing this in the spectrum, the spectral width and zero point (reference frequency) are also changed to keep all peaks at the same frequency.

7.13 Flip L/R

Flips (i.e. mirrors) the left and right side of the current dimension. The axis is not flipped.

7.14   Delete

Deletes specific data points from the current dimension. Should be entered in array from. For example: [1,2,40] deletes data points 1, 2 and 40. Remember that ssNake uses the python way of array indexing: 0 is the first point. The above code therefore deletes the second, third and forty–first data point. Do remember that the indexing goes from right to left in the case the current dimension is a spectrum (the frequency axis is flipped).

7.15   Split

Splits the current dimension in n parts with an equal number of data points. The newly created dimension is named D1, and all the other dimension numbers are increased by one (shifted one lower). This tool can only spilt the data in parts of equal size. It is therefore necessary to make sure that the size of the data in the current dimension is divisible by n, or in math.

7.16   Multiply

Multiplies the y-values of the data in the current dimension by a given number. If a single number is supplied, all the points are multiplied by this number. Alternatively, an array with the same length as the data can be given to specify a position dependent multiplication. For example, for a dimension with 3 data points: [1,3.34,2]. This multiplies the first data point by 1, second by 3.34, etc. Do not forget the about the mirroring in the spectral domain (see Diff)!

7.17   Reorder

Reorders the data in the current dimension. Input is an array with the new, desired positions. For example, [0,2,1] keeps the position of the first element ('0'), but interchanges the second and the third. It is also possible to supply the input array by loading a filer (as is used in Non-Uniform Sampling). If the new data length is longer than the old, the missing points are filled with zeros. Also, the length of the desired new dimension can be given. If the length of the final data is shorter than this value, some zeros are appended.

7.18   Concatenate

Concatenates (i.e. sticks) that data of dimension n after dimension $n + 1$. So if the original data is 10x5x3, and dimension 2 is selected, it becomes 10x15. This can be seen as the opasite of the 'Split' function. Note that after applying this function the numbers of the dimensions in the ssNake window changes accordingly.

8   Fitting

In NMR spectroscopy, the shape of the resonance lines can provide information on the strength of interactions that the nucleus experiences. ssNake can, as many other programs, fit some idealized lineshapes with are often encountered in real life spectra. Fitting in ssNake ranges from trivial 'full width at half maximum' (FWHM) fitting, to the complex quadrupolar Czjzek distribution. The following section will explain the way ssNake handles the different types of spectra it can fit, and which equations are used to simulate the different spectra.

8.1   Fitting basics

Roughly speaking, ssNake uses two types of fitting procedures: those with a mathematical solution, and those who have to be approximated iteratively. An example of the former type is the fitting of the width of a peak at halve height (FWHM). This involves nothing more than finding the left and right side of the peak (at halve height) and to calculate the width of this interval. This can be done in one easy mathematical procedure, and involves no iterative loop. The second type of fitting procedure is used for the powder lineshapes (e.g. quadrupolar and chemical shift lineshapes). With these problems, there is no direct mathematical solution. We are searching a parameter space for the solution that best fits the experimental spectrum. This is done by trying different settings, and qualifying hoe well these fit. Based on these attempts, a new setting is tested. This is continued until an optimum has been found. There are several ways to do such an iterative fitting: ssNake uses the Nelder-Mead algorithm [1] (also known as "Simplex").

8.2 S/N

S/N (or SNR) calculates the signal-to-noise ratio between a defined section of noise, and a peak. SNR is a useful way to describe the quality of a spectrum, with too low values being a sign for inaccurate data. In ssNake, opening the S/N tool creates a window, in which the limits of the section of noise has to be set, as well as the region with the peak of interest. Left clicking in the plot can set these values (noise limits first, then signal limits). Note that it is essential that the region that is specified as 'noise' really is noise. Any remaining signal will disrupt the SNR calculation. When the limits are set, ssNake directly calculates the SNR, using:

$$\text{SNR} = a_{\text{max}}/\text{std}(\text{NoiseRange}) \tag{2}$$

with a_{max} the maximum value within the range with the signal, and std(NoiseRange) the standard deviation of the noise.

8.3 FWHM

FWHM, or Full Width at Half Maximum, is a tool to calculated the width of a peak. Selecting the tool in the menu creates a window, in which the boundaries of the peak of interest have to be set, for ssNake to search for the width of the peak. Make sure that there is only one peak in the selected window, otherwise the result might be inaccurate. The easiest way to select the regions is by left clicking in the spectrum. After valid input is given, ssNake directly calculates the FWHM in the units of the current axis. The order of input of the two limits is not important for the eventual calculation.

8.4 Centre of Mass

Calculates the centre of mass of a selected part of a spectrum or FID. For a symmetric peak, this is equal to the centre of the peak. However, for an asymmetric peak, identifying the centre is not possible via visual methods and a centre of mass calculation is required. Opening the tool creates an input window, in which the limits have to be set (as with the FWHM tool). After these are set, the centre of mass is calculated in units of the current axis, using:

$$R = \frac{1}{M} \sum_i y_i \cdot x_i \quad , \tag{3}$$

with $M = \sum y_i$ and y_i the y-value corresponding to x-value x_i.

8.5 Second order quadrupole static

This fitting routine can be used to fit a static powder lineshape that is caused by the second order quadrupolar effect. A simplex method is used to find the (local) optimal solution. It is therefore necessary to make sure the initial parameters are somewhat close to the final fit, especially for fits with multiple sites.

Formulae

The second order quadrupole static relies on some relatively standard formulae: those of the second order quadrupolar coupling. This equation can be split up in two parts: an angle independent factor, and the angular factors. This latter part is then split in three parts depending on the influence of the asymmetry parameter η. In short:

$$\nu = -\frac{3C_Q{}^2}{6\nu_0(2I(2I-1))^2}(I(I+1)-3/4)(A+B\cdot\eta+C\cdot\eta^2) \tag{4}$$

With:

$$A = -\frac{27}{8}\cos(\theta)^4 + \frac{15}{4}\cos(\theta)^2 - \frac{3}{8} \tag{5}$$

$$B = \left(-\frac{9}{4}\cos(\theta)^4 + 2\cos(\theta)^2 + \frac{1}{4}\right)\cos(2\phi) \tag{6}$$

$$C = -\frac{1}{2}\cos(\theta)^2 + \frac{1}{3} + \left(-\frac{3}{8}\cos(\theta)^4 + \frac{3}{4}\cos(\theta)^2 - \frac{3}{8}\right)\cos(2\phi)^2 \tag{7}$$

For a given cheng number (i.e. number of powder orientations), the *A*, *B* and *C* parameters are contant, and only have to be calculated once for the fitting procedure.

9 Help

9.1 Chemical shift conversion tool

In NMR literature there are several definitions in use for describing the chemical shift tensor. This tool provides an easy way to inter convert them. The definitions are based on the very useful website of Klaus Eichele (http://anorganik.uni-tuebingen.de/klaus/nmr/index.php?p=conventions/csa/csa). In the most useful conventions, the three tensor values δ_{xx}, δ_{yy} and δ_{zz} are rewritten to an average value, a width and an asymmetry. Below follow the definitions of the different conventions.

The primary convention is called the 'Standard Convention'. This convention just gives the three principal components of the tensor, without any restructuring, called δ_{11}, δ_{22} and δ_{33}. They are ordered in such a way that: $\delta_{11} \geq \delta_{22} \geq \delta_{33}$.

The second convention we call the 'xyz Convention'. This is essentially the same as the Standard Convention, only the ordering is different. We now have δ_{xx}, δ_{yy} and δ_{zz}. With $|\delta_{zz} - \delta_{iso}| \geq |\delta_{xx} - \delta_{iso}| \geq |\delta_{yy} - \delta_{iso}|$. With δ_{iso} the isotropic value, defined as the average of the three principal components. In this definition, δ_{zz} is the value furthest away from the isotropic value, δ_{yy} the closest, and δ_{xx} the remaining value.

The third definition is the Haeberlen definition (or Haeberlen-Mehring-Spiess). Here, the principal components are ordered as in the xyz convention, only now they are redefined into a width, asymmetry and isotropic value. With $\delta_{iso} = (\delta_{xx} + \delta_{yy} + \delta_{zz})/3$, $\delta_{aniso} = \delta_{zz} - \delta_{iso}$ and $\eta = (\delta_{yy} - \delta_{xx})/\delta_{aniso}$. This way, the asymmetry parameter η always lies between 0 and 1. Not that due to the ordering of the principal components, the sign of the anisotropy (i.e. the width) is ill defined for $\eta = 1$.

The final definition is the Herzfeld-Berger definition. This convention uses the Standard Convention as a basis, and then defines the isotropic value as $\delta_{iso} = (\delta_{11} + \delta_{22} + \delta_{33})/3$, the span as $\Omega = \delta_{11} - \delta_{33}$ and the skew as $\kappa = 3(\delta_{22} - \delta_{iso})/\Omega$. This way, the span is always positive, while the skew lies between -1 and 1.

Use of the tool

Using the chemical shift conversion tool is quite easy. You can fill in all the necessary values for one convention, and push the 'Go' button next to it to convert it to all the other definitions. Note that ssNake also checks the definitions. So if you mess up the order of the Standard Convention, for example, it will be put in the correct order. The same is done for values that are outside the defined limits (e.g. η and κ). Pushing 'Reset' will reset all the values to 0.

Bibliography

[1] J. A. Nelder and R. Mead. *The computer journal*, **7**, 308 (1965).