



Documentation

Aim and resume

Farseer-NMR was developed to improve and facilitate the analysis of BioMolecular NMR data derived from proteins. Taking a series of 2D-NMR peaklists, automatic analysis, calculation and representation of the results as publication-quality plots is possible thanks to Farseer-NMR. You won't need to spend several days of meticulous work in spreadsheets or obviate part of your results due to complexity. Farseer-NMR presents a straightforward manner to extract the most out of your data.

This documentation PDF is intended to explain how Farseer-NMR works conceptually, the technical details on how to set a calculation running and how to read and analyse the generated results (**which may be numerous!**).

Table of Contents

I. Understanding Farseer-NMR.....	5
a) What is Farseer-NMR?.....	6
The daily problem.....	6
The solution.....	6
How?.....	6
What else?.....	6
b) The Farseer-NMR data structure.....	7
The Farseer-NMR Cube.....	8
c) Multidimensional Analysis Workflow.....	9
A Series of NMR experiments.....	10
Navigating the Farseer-NMR Cube's Axes.....	11
Experimental Series Analysis.....	15
d) The Farseer-NMR Analysis Routines.....	16
Stepwise Descriptions of the Different Routines.....	17
Performing Calculations.....	21
Plotting the results.....	22
e) Comparative Analysis.....	23
Comparative parsing.....	24
Comparing for each dimension/experimental variable.....	28
II. Technical Documentation.....	29
a) Information on available routines and parameters.....	30
Combined Chemical Shift Perturbations (CSP).....	30
Chemical shift normalisation.....	31
Searching lost residues along Y and Z.....	32
Data Fitting.....	33
The FASTA file.....	34
Theoretical PRE profiles.....	35
b) Axes restrictions.....	36
Data fitting.....	37
Different sample constructs.....	38
Paramagnetic NMR Analysis.....	39
c) How to set up a calculation Run.....	40
The initial input structure.....	40
Setting up the experimental tree.....	41
Settings Tab.....	42
d) Practical Considerations.....	43

Using different FASTAs.....	43
III. Results and output.....	44
a) The log file.....	45
b) The results folder hierarchy.....	46
The Main Run Folder.....	46
The Backbone and Sidechains Folders.....	47
The Comparisons Folder.....	48
c) Reading the results.....	49
d) Plotting Templates.....	50
Bar Plots.....	51
Residue Evolution Plots.....	55
Chemical Shift Scatter Flower Plot.....	57
The Δ PRE map.....	58
The Δ PRE heatmap.....	59
IV. Miscellaneous.....	60
a) Warnings, Errors and Troubleshooting (WET) list.....	61
b) Where to download.....	61
c) Social Network.....	61

I. Understanding Farseer-NMR

a) What is Farseer-NMR?

The daily problem



Biomolecular NMR-related projects require thorough investigation of a system under study, which usually translates into testing it against **multiple experimental variables** (e.g. ligand concentration, ligand nature, temperature, pH, paramagnetic agents). Such experimental setups ultimately generate large and complex datasets that easily overload human capacity of analysis by standard means. Treating datasets of this nature in a fast and straightforward manner is a **growing requirement for researchers**. NMR researchers benefit from many computational tools, available at the different stages of the NMR projects pipeline, which boost their capacity to extract the most out of NMR experiments in a fast and reliable way. In spite of this, one of the connections in the NMR analysis pipeline is broken due to a lack of software availability. This connection is the transformation of curated peaklists into biophysically-relevant restraints and data-rich tables and plots.

The solution



Farseer-NMR is a software package that **automatically treats, remediates, calculates and plots** NMR data and restraints derived from experiments measuring the sequential response of a system to a single or **multiple correlated variables**. The process of handling large amounts of diverse NMR data can be tedious, repetitive, error prone and time-consuming; taking days and, in some cases, even weeks. Farseer-NMR removes the tedium, minimises the effect of human error, reduces the time burden to seconds/minutes and simplifies data visualisation. Figure 1 shows the position of Farseer-NMR into the NMR analysis pipeline.

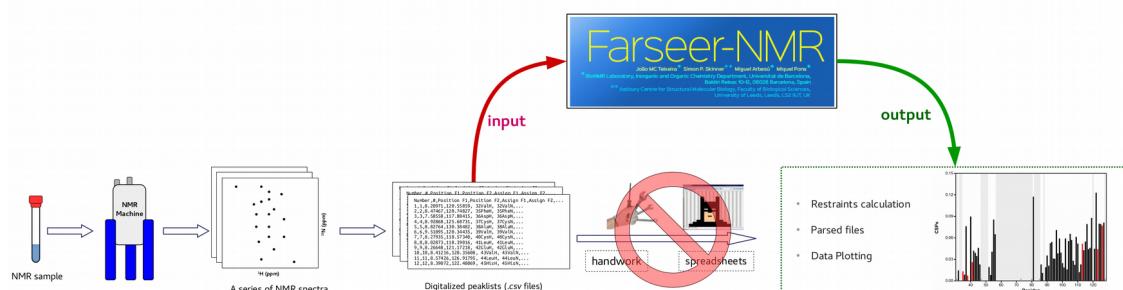


Figure 1: The BioNMR project pipeline. Farseer-NMR acts upon an old and persistent gap: the transformation of peaklists into human-readable and biological relevant data in the form of plots or parsed tables.

How?



Farseer-NMR uses high-dimensional Python 3 Numpy/Pandas arrays to **deconvolute multivariate dependent NMR data** into simpler parts, which are straightforwardly analysed and presented in a human-readable manner and without information loss.

What else?



We have implemented the most common (and some not so common) calculation routines (e.g. PRE, CSP) and several publication-quality plotting templates to improve data representation. Farseer-NMR is written completely in Python and can read the most common NMR peaklist formats: Ansig, NmrDraw, NmrView, Sparky and CcpNmr Analysis 2.4 via simple drag-and-drop import. The graphical interface is written using the most up-to-date version of PyQt, PyQt v5.8, and its modular code base enables facile extension.

b) The Farseer-NMR data structure

Farseer-NMR performs complex analysis on series of NMR experiments that inspect a system's dependency on (up to) three different variables (eg. ligand concentration, ligand nature, temperature, pH, paramagnetic agents, etc...).

Example 1:

Given a protein system **P**, the binding profile of the ligand **L1** was measured at five concentrations (**C**). The same protein **P** was screened against four related ligands (**L1, L2, L3, L4**) and each experiment was repeated at three different temperatures (**T1, T2, T3**).

The above experimental setup embodies a set of 12 experimental series, which result from the combination of two **continuous** variables (temperature and concentration) and one **discontinuous** variable (ligand nature). In total, 60 2D-NMR experiments were acquired ($5 \times 4 \times 3$). NMR data are sensitive to the contribution of each experimental variable. In order to fully understand the contribution of each variable to the system under study, it is essential to have a flexible, yet simple way to access the data, which preserves all information content and allows the deconvolution of those complex contributions into simpler parts.

The Farseer-NMR Cube

To freely navigate and explore experimental datasets spanning multiple conditions, Farseer-NMR loads the whole input data to a single digital object, a Python [Numpy/Pandas five-dimensional array](#); which, for the sake of simplicity, can be visualized as a cube made of 2D data points, where the three-dimensional axes of the cube (*x*, *y*, *z*) are the experimental variables (in Example 1, *ligand concentration*, *ligand nature* and *temperature range*), and each data point is a 2D-NMR peaklist (loaded as [pandas.DataFrame](#)) with the respective rows referring to the residues and columns to the experimental observables previously extracted from the user preferred NMR analysis suite. We have named this object the **Farseer-NMR Cube** (figure 2).

Cube dimensions can have any number of data points and, they can be accessed and combined freely to generate a panoply of series, which encode the answers to different experimental questions.

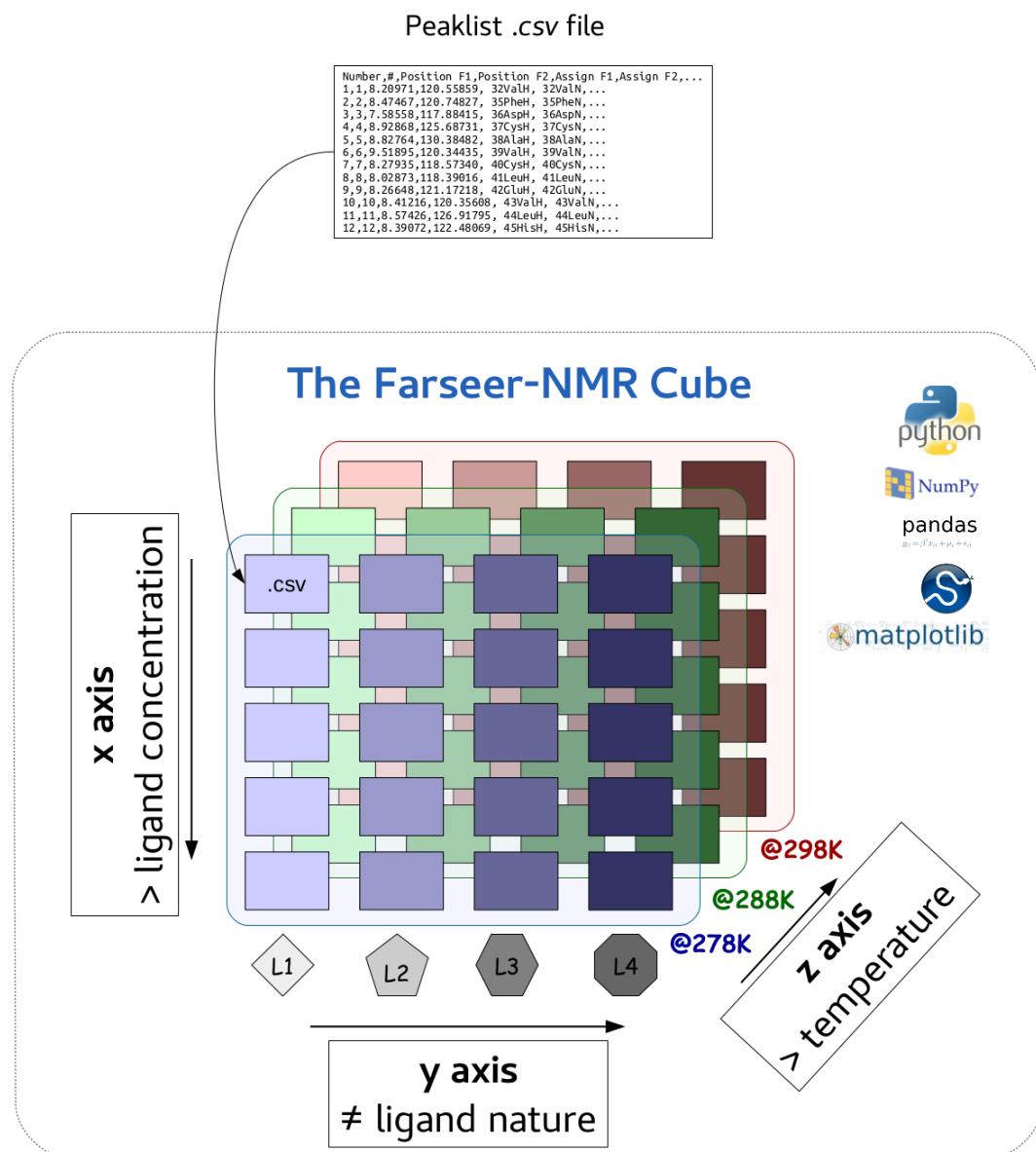


Figure 2: The schematic representation of the Farseer-NMR Cube.

c) Multidimensional Analysis Workflow

The great advantage of having the whole experimental dataset in a single digital object, [the Farseer-NMR Cube](#), is that it can be arbitrarily sliced to investigate specific questions, which are not limited to the acquisition schedule of the multivariable data. [Following the previous example](#), we can ask different questions of the dataset that directly relate to the conditions assayed:

1. *Ligand concentration range:*
 - a. Where does the *ligand* bind to the *protein system P*?
 - b. Are there multiple binding sites or allosteric effects?
 - c. What are the *ligand concentration* dependencies of these effects?
 - d. What is the binding constant?
2. *Ligand Nature:*
 - a. What is the binding profile of the various *ligands*?
 - b. Do the ligands interact with the same binding site?
 - c. Do they evoke the same changes in **P**?
3. *Temperature variations:*
 - a. How does the *temperature* affects the binding profiles of the ligand library?

As explained above, the Farseer-NMR Cube's three dimensional axes correspond to the progression along the three experimental assayed conditions and, therefore, we can explore the above cited questions by slicing the cube along the different axes, where the above points **1**, **2** and **3** correspond to Farseer-NMR Cube's axes **X**, **Y** and **Z**, respectively.

A Series of NMR experiments

A series of NMR experiments is any set of experiments (peaklists) that follow the progressive change of a variable, which can be *continuous* or *discontinuous*, with the first experiment being the reference to which all the others are consecutively compared.

Following the above rationale, we can fix two points along two given axes (e.g. X=C2 and Z=T2) and slice along the third axis (Y) to generate a 1D-vector of 2D-NMR peaklists, which would correspond to the experimental series [Z=T2][X=C2][Y=[L1-L4]]. In general terms, we can generate a set of 1D-vectors corresponding to different series along an axis for each combination of the other two axes.

[Following example 1](#), out of the 12 experimentally-acquired series relating to the progression of ligand concentration (4×3 , in our example), Farseer-NMR can extract up to 47 *in silico* generated series of experiments that result from the different combinations of the other experimental variables:

- 4×3 (ligand concentration)
- 3×5 (ligand nature)
- 5×4 (temperature dependence)

Navigating the Farseer-NMR Cube's Axes

The implemented workflow sequentially generates *series of experiments* out of the Farseer-NMR Cube, which result from all possible combinations of X, Y and Z, by fixing data points along two axes and extracting the series along the third one.

Evolution along the X axis

Farseer-NMR analyses how the overall dataset evolves along the X axis by fixing two data points in Y and Z and creating a series of peaklists along the X axis. A full set of series is generated by walking through the Y and Z data points (Figure 3). In this example a total of 12 series are generated.

Evolution along the x axis

Fixed y and z

Total of **4x3** series of experiments analysed.

X. Ligand concentration range:

- Where does the *ligand* bind to the *protein system P*?
- Are there multiple binding sites or allosteric effects?
- What are the *ligand concentration* dependencies of these effects?
- What is the binding constant?

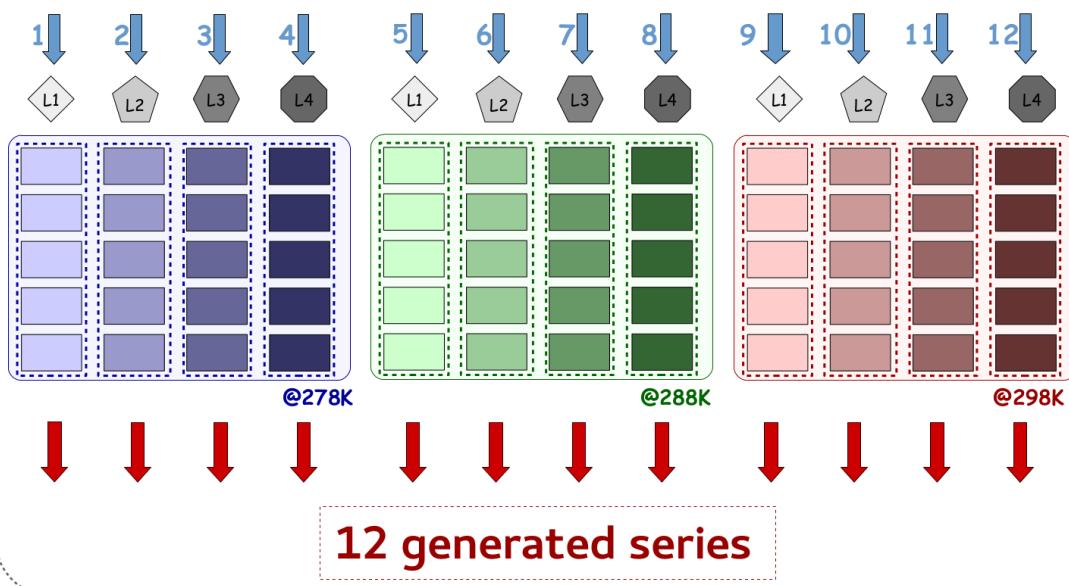


Figure 3: Representation of how the X axis series are generated along the Farseer-NMR Cube.

Evolution along the Y axis

In this case, Z and X are fixed and peaklists are stacked along the Y axis (Figure 4).

Evolution along the y axis

Fixed z and x

Total of **3x5** series of experiments analysed.

Y. Ligand Nature:

- a. What is the binding profile of the various *ligands*?
- b. Do the ligands interact with the same binding site?
- c. Do they provoke the same changes in P?

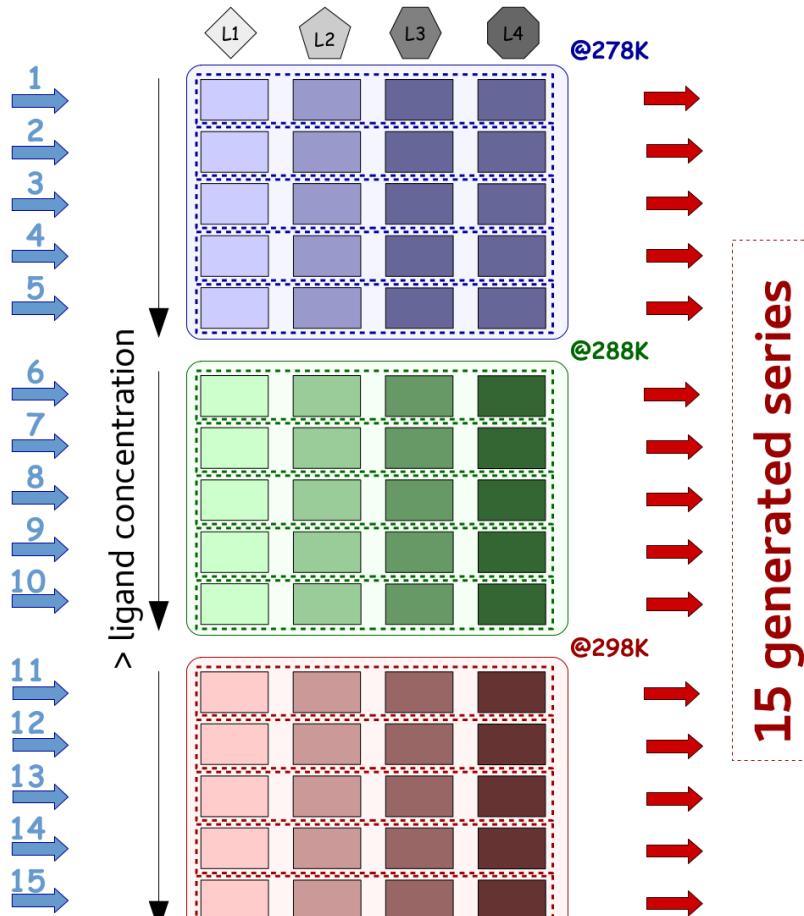


Figure 4: Representation of how the Y axis series are generated along the Farseer-NMR Cube.

Evolution along the Z axis

In this case Z and X are fixed and peaklists are stacked along the Z axis (Figure 5).

Evolution along the z axis

Fixed x and y

Total of **5x4** series of experiments analysed

- z. Temperature variations:
 - a. How does temperature affects the binding profiles of the ligand library?

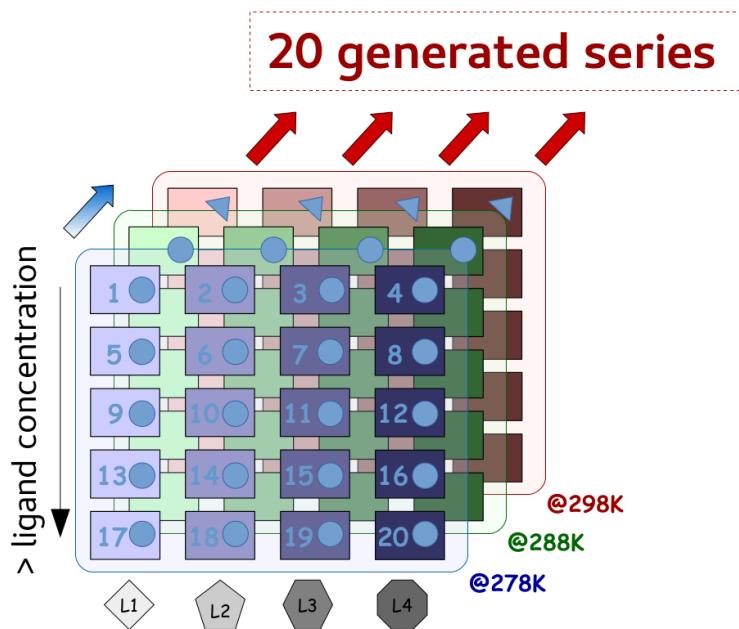


Figure 5: Representation of how the Z axis series are generated along the Farseer-NMR Cube.

Experimental Series Analysis

Experimental series are analysed by comparing each peaklist in the series to the reference peaklist. The nature of the output of such comparisons depends on what is being analysed, normally, calculations along series generate NMR parameters (chemical shift perturbations, intensity ratios, etc.), which are plotted at the end. The results obtained are stored directly in the peaklist DataFrame itself. Farseer-NMR contains a set of [analysis routines](#) that can be applied to extract the most out of each series.

For each series, the output of the analysis is exported to [dedicated folders](#).

d) The Farseer-NMR Analysis Routines

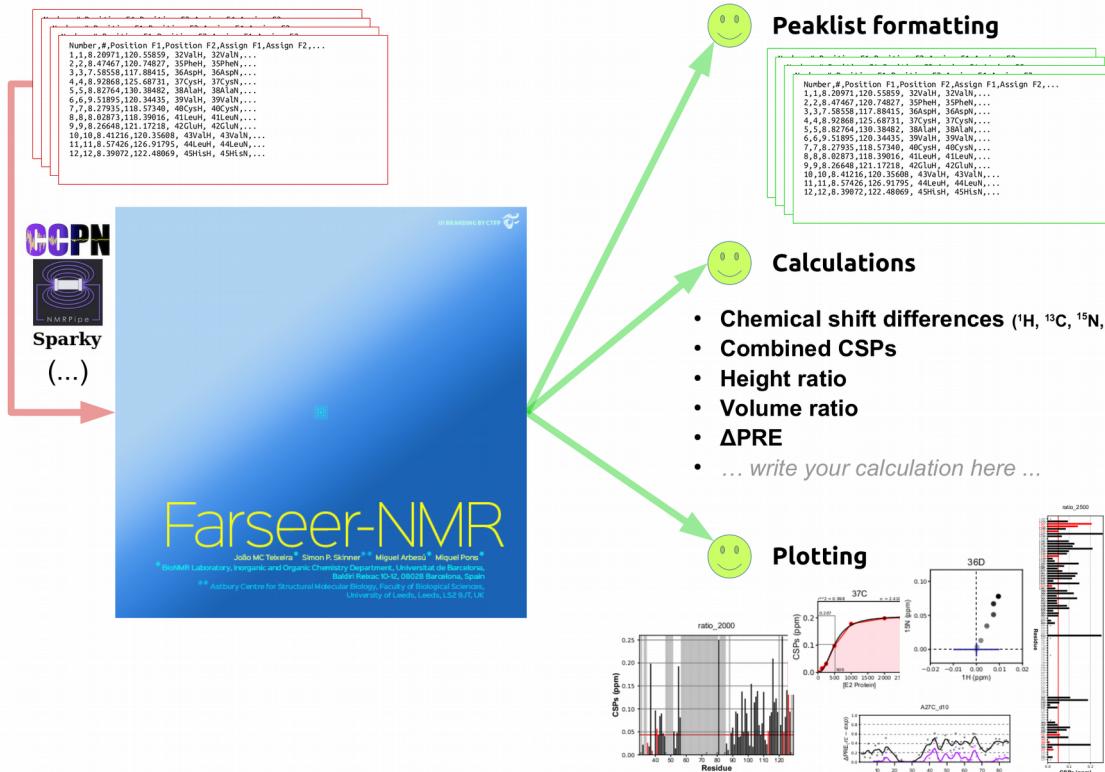


Figure 6: The schematic representation of Farseer-NMR workflow and analysis routines.

The Farseer-NMR routines workflow can be summarised by three main steps:

1. Series of two-dimensional NMR peaklists are extracted from the [Farseer-NMR Cube](#) according to the [Multidimensional Analysis Workflow](#).
2. For each series, NMR restraints are calculated from the NMR observables, which evolve along the series.
3. Results are conveniently plotted and exported in user-friendly parsed data tables organised in dedicated folders.

Stepwise Descriptions of the Different Routines

Treating and Formatting peaklists

For the sake of clarity, this section is explained at a very general level, and there are technical aspects that should be considered in order to fully understand the core of the Farseer-NMR workflow.

The steps described in the following subsections [Reading Assignment Information](#), [Identifying the lost residues](#) and [Identifying the unassigned residues](#), are actually performed before the creation of the [Farseer-NMR Cube](#) and while the peaklists are being organised hierarchically in the [PeakList Tree](#).

Consequently, the following routines serve two purposes:

1. identification of scientific relevance of *lost* and *unassigned* residues
2. adjustment of all peaklists loaded to the same size (rows), due to technical issues in comparing peaklists of different sizes, without re-indexing them.

For these reasons, these steps are performed prior to the creation of the Farseer-NMR Cube. In theoretical terms, this means that these routines are applied to all series. In practical terms, this means that:

1. By default, identification of *lost* residues by comparing peaklists to the reference only occurs along the X axis.
 - a. To overcome this you must activate the [search for lost residues Y/Z](#) flag. This option is especially useful when analysing paramagnetic data, since peaks can disappear after introduction of the paramagnetic tag.
2. Identification of *unassigned* residues based on FASTA files only occurs along the X axis, which means that FASTA files are unique for each Y data point; [as described](#), different FASTA files can only be input along the Y axis.
 - a. This can be overcome by activating the corresponding flag in [search for lost residues Y/Z](#) menu.

Reading assignment information

NMR peaklists are simply tables where rows represent residues and columns contain all the information regarding residue identification, NMR observables and notes.

In the preferred peaklist format of Farseer-NMR (CCPNMR v2 format), assignment information is concatenated into a single column (*Assign F1* in the format `1MetH`) and it is therefore necessary to split this information. Three additional columns are created (*Res#, 1-letter, 3-letter*) to store the assignment information and this information is used to index all the data and resulting calculations (Figure 7).

Assign F1	Res#	1-letter	3-letter
0GlyH	0	G	Gly
1MetH	1	M	Met
2AspH	2	D	Asp

Figure 7: Expansion of *Assign F1* column to three new assignment informative columns.

Identifying lost peaks

Along experimental series, it is common that peaks disappear either due to linewidth broadening or because tracking becomes impossible due to overlap, as a consequence, peaklists along series can differ in size, i.e. different numbers of rows. We describe disappearing peaks as *lost* peaks, and it is important to identify these as they are information rich.

Technical note: The difference in peaklists size greatly hinders direct and straightforward analysis of the peaklists files in traditional plotting tools, because row identities won't match when comparing row by row. In the past, this issue had to be handled manually, whereas Farseer-NMR does it automatically.

The second task of Farseer-NMR is to identify *lost* residues and, thus format all the input peaklists to the same size by comparing the peaklist data points to the reference. New rows are added for each *lost* residue found, and newly generated *Peak Status* column tags residues as *measured* or *lost* accordingly to their nature. Other columns are filled with user defined default values or numpy.nan values (Figure 8).

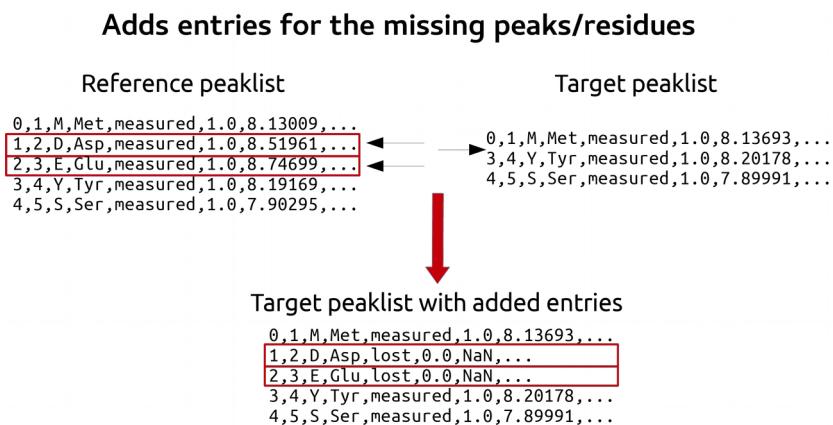


Figure 8: New rows are added to identify the peaks that were lost with respect to the reference experiment.

Identifying unassigned residues

This feature is optional and is performed in a similar manner to the identification of the *lost* residues, with the exception that all the peaklists in the series (reference included) are compared to a previously loaded [FASTA file](#) containing the protein's full primary structure. New rows identify the *unassigned* residues.

Adds entries of unassigned residues based on a FASTA sequence file

```
MDEYSPKRHDVAKLKFCLCESLYDEGIATLGDSHHGWWNDPT  
SAVNQLNLDIEHIASFVMSFKIKYPDDGDLSELVEYLDPTY  
TLFSSYGINDPELQRWQTKERLFRLFSGEYISTLMKT  
  
5,S,Ser,measured,1.0,7.90295,...  
9,H,His,measured,1.0,7.47479,...  
  
5,S,Ser,measured,1.0,7.90295,118.6998,...  
6,P,Pro,not_assigned,0.0,NaN,NaN,...  
7,K,Lys,not_assigned,0.0,NaN,NaN,...  
8,R,Arg,not_assigned,0.0,NaN,NaN,...  
9,H,His,measured,1.0,7.47479,118.26708,...
```

Figure 9: New rows are added to identify those residues that are not assigned in the reference, and in consequence, in the whole series.

Performing Calculations

Farseer-NMR compares each series data point (peaklist) to a reference experiment and calculates the user-specified restraints. The results are added to newly-generated columns in the peaklists' [pandas.DataFrame](#)s, which are all exported together at the end of a run. Farseer-NMR can calculate:

1. Chemical shift differences for each nuclei
2. [Combined Chemical Shift Perturbations \(CSP\)](#)
3. Intensity ratios
4. [Data Fitting](#)
5. [ΔPRE](#)

Plotting the results

Farseer-NMR contains several [publication-quality plotting templates](#) to represent calculated data. For each series analysed, the calculated restraints can be plotted in any and all available templates. Each template is highly customisable and enables a user to adapt the representation of data to different publications' requirements.

e) Comparative Analysis

Comparative Analysis does not generate new data. Rather, it consists in parsing algorithms that reorganise the Farseer-NMR results differently.

Further on, we will see that comparative analysis in fact generates additional pseudo dimensions in the [Farseer-NMR Cube](#), which are essential to analyse paramagnetic derived restraints.

Example 2:

The protein **P** was investigated against five progressive concentrations of two similar, yet not identical, ligands (**L1** and **L2**), at 298K. The Farseer-NMR Cube of this data set has the following form (Figure 10):

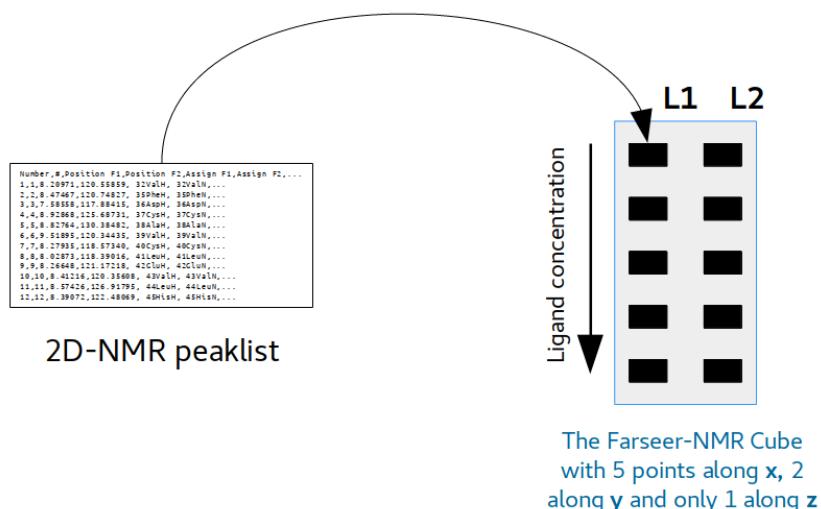


Figure 10: An example of two experimental series stacked along Y axis.

The **z** dimension consists of a single datapoint (298K), and therefore can be disregarded (although, technically, it exists). Consider the case where the user wants to analyse the two experimental series for **L1** and **L2** only along the **x** axis of the Cube, that is, in terms of increasing ligand concentration. In this case, two folders containing the corresponding results for **L1** and **L2** are generated ([further reading on folder organization](#)) containing all the tables and plots requested by the user ([analysis routines](#)).

The generated plots, within each folder, will represent the evolution of the experimental series in response to increasing ligand concentration. However, it is sometimes fruitful to compare the results obtained for the two ligands at a given concentration.

This requirement could be fulfilled by the user simply opening the two plotting figures (or hard copies) and placing them side by side for comparison. This manual procedure would quickly become awkward when comparing large experimental datasets in which several points were acquired for a particular variable: e.g. screening a library of 20 ligands.

Comparative parsing

Following on from the above example, the generated data can be compared along the **y** axis, but it cannot be compared along the **z** axis (temperature), because there is only one point on this axes: *there is nothing to compare to.*

Calculating along X and comparing along Y

Farseer-NMR comparative analysis generates **stacked** plots along the **y** axis for **each** of the points in the **x** axis with the results **previously** obtained in the **x** dimension (ligand concentration), as shown in Figure 11.
N.B.: Comparing along the **y** axis is different from analysing an experimental series along the **y** axis!

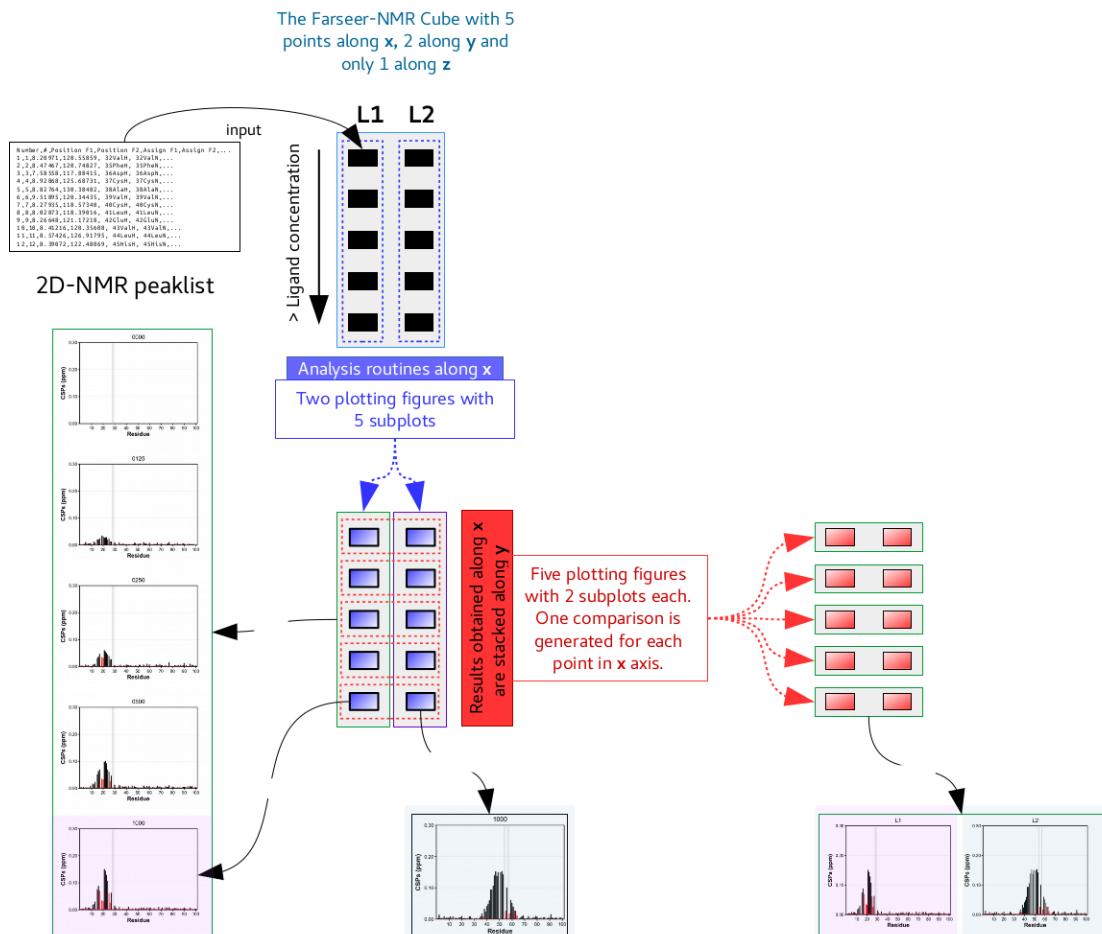


Figure 11: Schematic representation of how results generated along the X axis are compared along the Y axis.

Calculating along Y and comparing along X

Similar to the above, subplots resulting from calculations along the Y axis are stacked along X.



Figure 12: Schematic representation of how results generated along the Y axis are compared along the X axis.

Calculating along Z and comparing along X

Similar to the above, subplots resulting from calculations along the Z axis are stacked along X.

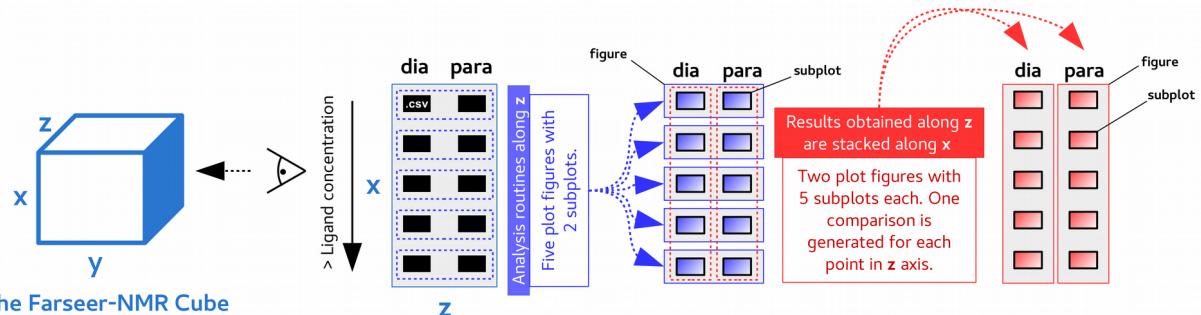


Figure 13: Schematic representation of how results generated along the Z axis are compared along the X axis.

Comparing for each dimension/experimental variable.

The same rationale can be applied to any dimension. Therefore, for each condition axis, comparisons can be made along the other two axes, as long as the axis, along which we want to perform the comparison, has more than one datapoint. Consider the [previous example](#) where 12 experimental series summing 60 HSQC experiments investigating the system's dependence on 3 different variables – Farseer-NMR Cube has dimensions of $5 \times 4 \times 3$.

We have seen that in total 4×3 series can be analysed **along the x axis**. Applying comparative analysis along the y axis will generate 5×3 new sets of parsed data (not new data) and, comparisons performed along the z axis results in a total of 5×4 sets of parsed data.

II. Technical Documentation

a) Information on available routines and parameters

Combined Chemical Shift Perturbations (CSP)

Combined Chemical shift perturbations (CSP) are calculated according to equation 8 of [Williamson 2013](#) (see Corrigendum):

$$CSP_{(ppm)} = \sqrt{\frac{1}{2} [\delta_H^2 + (\alpha \cdot \delta_N)^2]}$$

where δ_H is the chemical shift (cs) difference for the proton, δ_N is the chemical shift (cs) difference for the nitrogen dimension, and α is the normalization constant between both dimensions.

Farseer-NMR allows the user to set up individual α values for each residue type.

Chemical shift normalisation

In some occasions chemical shifts need to be normalized to a given peak. The Chemical Shift Normalisation routine allows the user to normalise chemical shifts along the x axis series to a given peak (residue) of the reference experiment of the corresponding series.

Searching lost residues along Y and Z

In the case of Paramagnetic analysis it is likely to occur that upon addition of the paramagnetic tag some peaks disappear already in the paramagnetic reference experiment, when comparing to the diamagnetic reference. In the Farseer-NMR setup, the search for lost residues is performed along the X axis. To activate the search along Z and Y, the specific flag should be activated. This procedure will compare the first paramagnetic peaklist to the first diamagnetic peaklist and identify those residues lost because of the presence of the paramagnetic species. Following this, the usual procedure of search lost residues along x will take place (for the diamagnetic and paramagnetic series) and the residues identified initially will be taken into consideration.

Data Fitting

In Farseer-NMR we have designed a code platform to simplify the implementation of the user required routines for data fitting. Currently, and as a demonstration, Farseer-NMR can fit continuous data, along the x axis, to the [Hill Equation](#):

$$Y = \frac{Y_{max}[S]^n}{K_{0.5}+[S]^n}$$

where, Y is the observable or calculated NMR parameter, S is the experimental variable (x axis), K0.5 is the half maximal constant and n the Hill coefficient.

Fitting results are stored in two files: 1) a fitting log file where all the information regarding the fitting procedure is stored (residue wise) and 2) a fitting results table (.csv file) where the summary of the fitting with the fit parameters are stored.

Fitting results are also drawn in the [Parameter Evolution per Residue](#) plot template.

The method `FarseerSeries.perform_fit` defines the fitting workflow inside Farseer-NMR, fitting routines are coded in the `fslibs/fitting.py` library. New fitting functions and routines can be implemented following the example for the Hill fit.

The FASTA file

Please consider reading the [axes restrictions for FASTA files](#). Farseer-NMR accepts FASTA files containing the primary sequence of a protein construct and accepts the following different formats:

- optional header starting with ‘>’ character
- full sequence in a single line
- sequence split over multiple lines

Theoretical PRE profiles

Theoretical PRE profiles (such as those generated by [Flexible-Meccano](#)) should be given as a .tsv file where the first column are the residues number and the second column the PRE value. The first line should be commented by “#” character directly followed (without space) by the residue number where the paramagnetic tag was engineered, for example:

```
#50
1 8.469022e-01
2 8.543896e-01
3 8.474678e-01
```

This file should have .pre extension and be placed under the folder *spectra/para/<cond2>/* along side the .csv peaklist files.

b) Axes restrictions

Farseer-NMR loads all the peaklist data into a 5-dimensional array, which can be thought of as a cube made of 2-dimensional data points – the [Farseer-NMR Cube](#). The **x**, **y** and **z** axes of the cube correspond to the different analysed experimental conditions and a maximum of three distinct experimental conditions can be analysed simultaneously. If only one condition is analysed, the Cube is simply a vector or a set of vectors.

In theory, any axis can have any number of data points, however, at present, the GUI allows a maximum of ten to be specified in one calculation run, though this can be easily adjusted if necessary.

Farseer-NMR has been designed to enable any kind of variable to be analysed, however, some caveats apply concerning what type of analysis can be performed along these different axes.

Data fitting

Farseer-NMR allows the fitting of biophysical parameters to continuous data, for instance the determination of affinity constants over a ligand concentration range. Further information about available data fitting routines can be found in [its corresponding section].

Parameter fitting is only available along the **x** axis. It is possible to represent a concentration range series along **y** and **z** and perform all the [analysis routines](#), however, the parameter fitting algorithms are not available for data represented by these axes.

Different sample constructs

Many biomolecular NMR studies involve the analysis of different protein constructs/mutants. A typical example would be:

A series of five ligand concentrations (**L1** to **L5**) are probed against four different protein constructs/mutants (**P1, P2, P3, P4**), resulting in a single 5×4 face of the Farseer-NMR Cube, with ligand concentration along the **x** axis and the protein constructs along the **y** axis.

Conceptually, the two axes could be swapped and the ligand concentration could be represented along **y** and the protein constructs along **x**. However, in order for Farseer-NMR to [read FASTA files](#) and complete the output lists with information on the *unassigned* residues, this data must be put along **y** axis, since FASTA files can only be loaded for the data points of the **y** axis. Consequently, swapping the two axes would prevent the effective use of FASTA files in the analysis.

FASTA files are read only on the **y** axis, but, nevertheless, **y** can take any kind of variable.

Paramagnetic NMR Analysis

Paramagnetic NMR investigation of a system requires the acquisition of all data under both *diamagnetic* and a *paramagnetic* conditions. If, for example, a protein-ligand titration was performed as part of a paramagnetic NMR study, the dataset would consist of two titration series: *diamagnetic* and *paramagnetic*. The *diamagnetic* and *paramagnetic* series would fit as data points of a second axis where the *concentration range* would be fit on the first axis, e.g. *x* axis (see above).

Farseer-NMR contains several routines specifically for paramagnetic NMR data analysis and these routines are restricted to analysis along the *z* axis. Therefore, when analysing paramagnetic NMR data, e.g. PREs, input data should be organised such that the first point in the *z* axis corresponds to the *diamagnetic* series and the second point to the *paramagnetic* series.

c) How to set up a calculation Run

The initial input structure

This section assumes the previous section on the [Farseer-NMR Cube](#) has been read. As explained in [Multidimensional Analysis Workflow](#) section, Farseer-NMR performs complex analysis on NMR data by generating series of experiments from permutation of the experimental variables. However, to initiate a Farseer-NMR run, and hence create the Farseer-NMR Cube, data should be input in a hierarchical fashion. A logical tree of experiments is created where the top level corresponds to the z axis of the Farseer-NMR Cube, followed by the y axis data points and finally the x axis data points, see [next subsection](#).

Setting up the experimental tree

To configure the Farseer-NMR Cube under the GUI, an hierarchical tree ($Z \rightarrow Y \rightarrow X$) must be defined and populated with the corresponding peaklists. Peaklists can be input via two different ways: 1) Through selection “Spectrum Path” button in Settings tab that selects the folder containing the peaklists (Figure 14) or 2) they can be directly dropped in the Side bar of the Peaklists tab (Figure 15), this method can be perform in addition to 1). Peaklist files in the Side Bar cannot share the same name.

The number of data points in each axis can be defined as well as the name for each data point. The “Setup Experimental Series” button draws the experimental tree which must be populated with the peaklists by drag and drop from the side bar. Peaklists in the three can be placed back to the side back via right-click button.

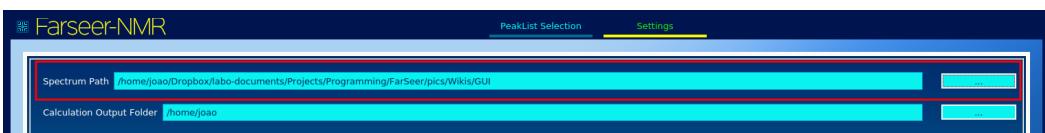


Figure 14: Cropped region of the Settings tab. Selecting the folder containing the peaklist files in the via “Spectrum Path” button.

Figure 15: Adding peaklists to the Side bar, configuring the tree and populating it by drag and drop in PeakLists Selection tab.

Settings Tab

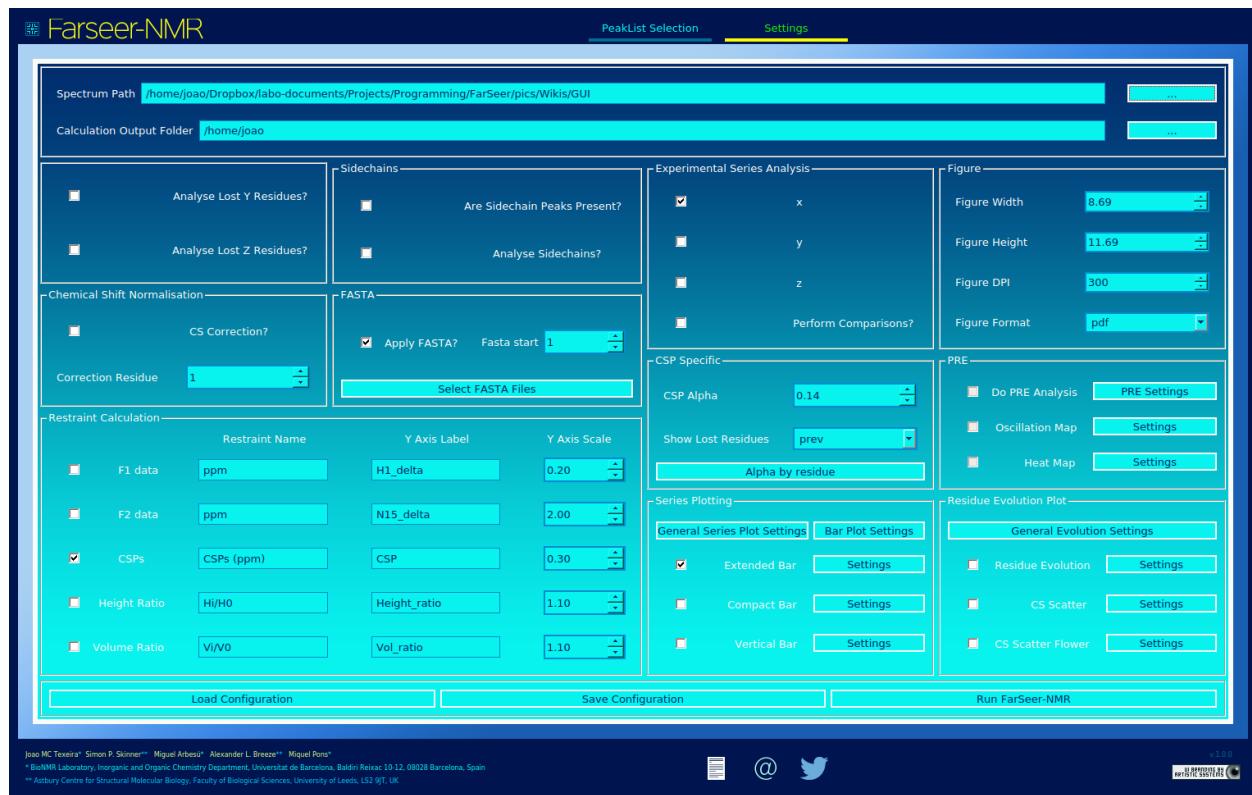
All the settings that have reached a stable release are available in the Farseer-NMR user interface and can be configured directly in the Settings tab. Experimental routines under implementation will be available first only via the JSON configuration file.

Settings are categorised in individual boxes according the routines or group of routines they belong to. Additional, and specific, settings may be available by pop-up menus.

The “Calculation Output Folder” is the folder where the spectra will be parsed in and all the output results will be stored according to [section](#).

Plotting settings submenus are hierarchical. For example, settings under “General Series Plot Settings” apply to every plot categorised as Series Plot, which are: the bar plots and the DPRE map.

The Load/Save Configuration buttons allow the user to keep copies of the configuration JSON file containing information of the whole Farseer-NMR session. A previous session can be readily recovered simply by loading a previously configures JSON file.



d) Practical Considerations

This section gathers different guidelines and examples that might help the user to further understand the usage of the Farseer-NMR software and wisely setup its runs.

Using different FASTAs

Farseer-NMR can [load multiple different FASTA files](#) to compare parameter evolution for different mutants.

However, in order to perform analyses along the **y** axis with different FASTA files, Farseer-NMR requires that FASTA sequences are the same length, otherwise row numbers won't match when comparing peaklists. If no analysis is to be performed along **y**, and only the analysis of several constructs is to be performed along **x**, there is no requirement for FASTA sequences to have the same length.

III. Results and output

a) The log file

Farseer-NMR prints the progress of the run to the Terminal showing detailed information on all the operations performed. At the end of a run, the full log is exported to an external file written using [Markdown syntax](#).

b) The results folder hierarchy

The Main Run Folder...

... is the folder that contains the *spectra/* folder (which contains all the input peaklists), is defined in the GUI in the “Calculation Output folder” under the “Settings” tab, and is where all the Farseer-NMR generated output will be stored:

1. a copy of the user defined variables
2. a log file
3. folders where results are stored

The Backbone and Sidechains Folders...

... are created in the main calculation folder and separate the results obtained for backbone atoms and those of side-chains atoms (if present). Farseer-NMR separates these two types of results because, from experience, concatenating results for backbone atoms and side-chains atoms inside the same tables and plots results in awkward representations – both folders have the same internal hierarchy.

The Calculations folder...

... is found under the *Backbone* or *Sidechains* folders and stores the results generated for the different sets of series analysed along each [Farseer-NMR Cube](#) axis (condition).

The conditions subfolders...

... contain the series originated from slicing the [Farseer-NMR Cube](#) along a specific axis, where *cond1*, *cond2* and *cond3*, refer to X, Y and Z axis respectively.

The different series

The series generated for each axis (*cond** subfolder) are hierarchically stored according to the data points fixed for each of the other two axes. Axis parent-child permutation follow the order:

- Z → Y → X
- X → Z → Y
- Y → X → Z

Some examples:

- The analysis generated from a series of experiments (*increasing ligand concentration*) performed at 298K for ligand *L1*, will be stored under the following folder tree: *Backbone/Calculations/cond1/298/L1/*.
- For the 298K for ligand *L2*, the folder would be *Backbone/Calculations/cond1/298/L2/*.
- Along the Y axis, for ratio1 and 278K, *Backbone/Calculations/cond2/ratio1/278/*.

Each of these series subfolders store the results generated from the [Farseer-NMR Analysis routines](#). [Read further](#) on how the results are stored.

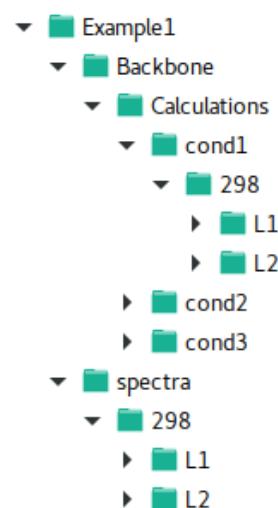


Figure 16: An example of how results for the different dimensions are organised under the Farseer-NMR data structure.

The Comparisons Folder

The *Comparisons* folder is found at the same level as the *Calculations* folder, stored under *Backbone* (or *Sidechain*), and stores the output from the [Comparative Analysis](#) method.

This folder has an additional initial subfolder, namely C1, C2 or C3, corresponding to the X, Y and Z axes, respectively. These folders represent the *axis along which the calculations* were previously performed. Inside, there are *cond1/*, *cond2/* and *cond3/* folders corresponding to the *axes along which the comparisons* were performed.

Example:

Calculations performed along the X axis and compared along the Y axis are stored under: *Backbone/Comparisons/C1/cond2/*.

Recall that each comparison is by itself a series of experiments. Therefore, in this folder hierarchy you will find the that series are organised in the same manner as described above for the *Calculations/* folder. The *Backbone/Comparisons/C1/cond2/* structure is followed by a <preceding axis>/<succeeding axis>/<[results]> folder tree where series along these axes are stored. Below a descriptive picture where, 3D_comparisons is the main Run folder, r1 is the first data point along the X axis and 278K is the first data point along the Z axis. r2 and r3 are other data points along X axis and 298K is the other datapoint long the Z axis. This series runs along the Y axis data points.

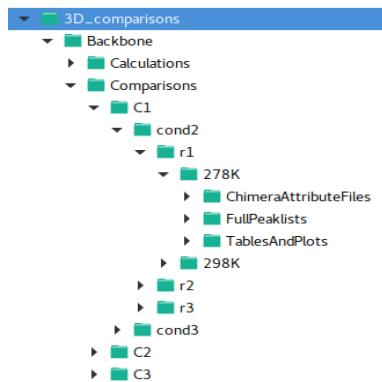


Figure 17: An example of how comparative analyses are organized under the Farseer-NMR data structure.

c) Reading the results

Inside each series folder, you will find different subfolders that organise the output data: tables, parsed files and plots.



Figure 18: The folders where results from the Parseer-NMR Analysis routines are stored.

The **FullPeaklists** folder stores the parsed peaklists in tab separated files that make up the series analysed. These peaklists have the same information as the originally input peaklists plus the results generated by the analysis routines and improved representation features:

1. All peaklists are parsed to the same length (number of rows/residues) so that they can be easily compared externally,
2. Identification of *unassigned* and *lost* peaks (Proline residues included),
3. Three additional columns identifying the *residue number*, the 1-letter amino acid code, the 3-letter amino acid code and a *Peak Status* information.
4. A column for each of the calculated NMR parameter.

The **TablesAndPlots** subfolder stores the plots created. One subfolder is created for each NMR parameter represented (*H1_delta*, *15N_delta*, *CSP*, *Height_ratio*, ...). Inside each subfolder is a figure file for each plotting template requested and a *.tsv* file with the data used for creating those plots. See here for a the list of [all available plotting templates](#).

The **ChimeraAttributeFiles** folder stores [Chimera Attribute](#) parsed files that can be directly used in [UCSF Chimera](#) and contain the calculated data for each restraint and each titration data point.

d) Plotting Templates

Farseer-NMR contains a set of plotting templates that represent the calculated data in a simple, organised and **publication-ready** manner. There are plots that represent commonly used styles, and others that we have specifically designed and implemented for improved data representation.

Each generated figure represents the restraint evolution along the whole titration, either in different subplots or concatenated into a single plot.

The structure of the figures, subplot organisation in columns and rows, colours, font types and several other plotting style options are highly customisable under in the corresponding GUI menu or the JSON configuration file.

Below is a dummy example, where a randomly generated protein of 100 residues is probed against different concentrations ratios of a ligand (1:0, 1:0.125, 1:0.250, 1:0.5, 1:1, 1:2, 1:4) which cause chemical shift perturbations in a specific region.

Bar Plots

Bar plots represent the evolution of a calculated restraint in configurable and commonly used form. There are three bar plot templates available: **compacted**, **extended** and **vertical**.

General features:

- all text is customisable (font type, size and style)
- X and Y ticks and scales are customisable
- customisable colours for identification of *lost*, *unassigned* and *measured* bars
 - *lost* residues can be represented in three different manners ('full', 'prev' or 'zero')
 - 'full', represents a full bar
 - 'prev', represents the value of the previously measured point
 - 'zero', represents no value
- customisable bar width
- identification of Proline residues (boolean flag)
- user-defined labelling of bars
- user-defined colouring of bars
- a grid option
- a significance threshold line

Compacted Bar Plot

Compacted bar plots are designed to fit a half-page width figure in a scientific reviewed publication and are generally drawn in an overall figure of a columns vs rows matrix of subplots (Figures 19 and 20).

Specific features:

- summarised x axis ticks
- shadowed regions to represent *unassigned* residues.

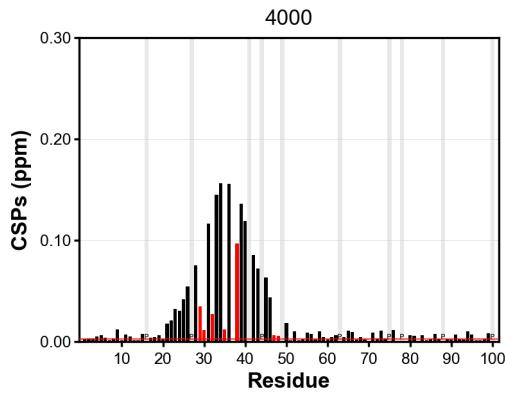


Figure 19: Compacted Bar template subplot

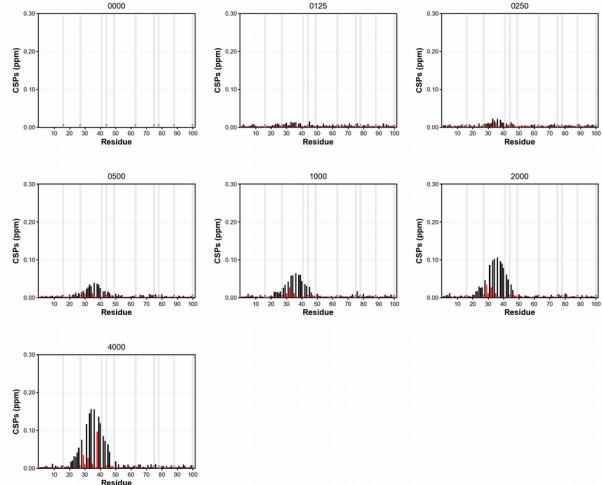


Figure 20: Full picture of a 3×3 subplot table representing the CSPs evolution of the whole series.

Extended Bar Plot

The extended bar plot is designed to fit a whole page width figure in a scientific reviewed publication and are generally drawn as overall figures of vertically stacked subplots representing the titration evolution (Figures 26 and 27).

Specific features:

- bars individually identified by residue labels up to 100 labels (larger proteins get progressively summarized ticks)
- customisable x ticks colours

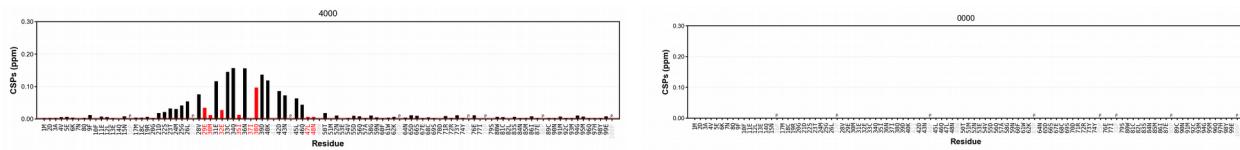


Figure 21: Extended Bar template subplot

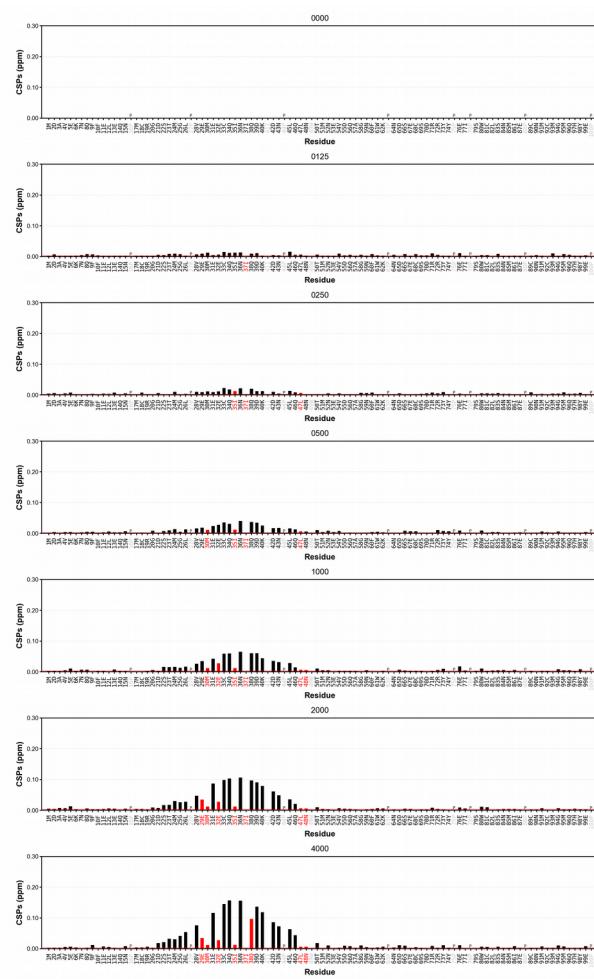


Figure 22: Full picture of a 7×1 subplot table representing the CSPs evolution along the whole series.

Vertical Bar Plot

The vertical bar plot is designed to fit narrow spaces and column organisation styles in scientific publications and are generally drawn as overall figures of horizontally stacked subplots representing the titration evolution (Figures 23 and 24).

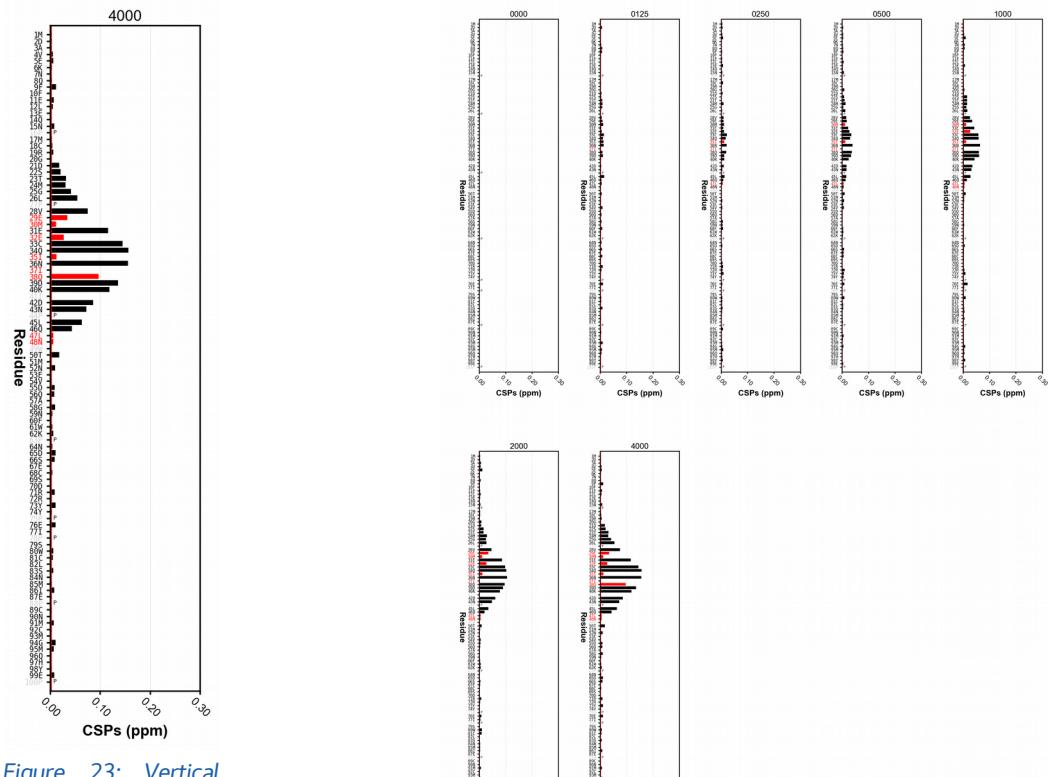


Figure 23: Vertical Bar template subplot

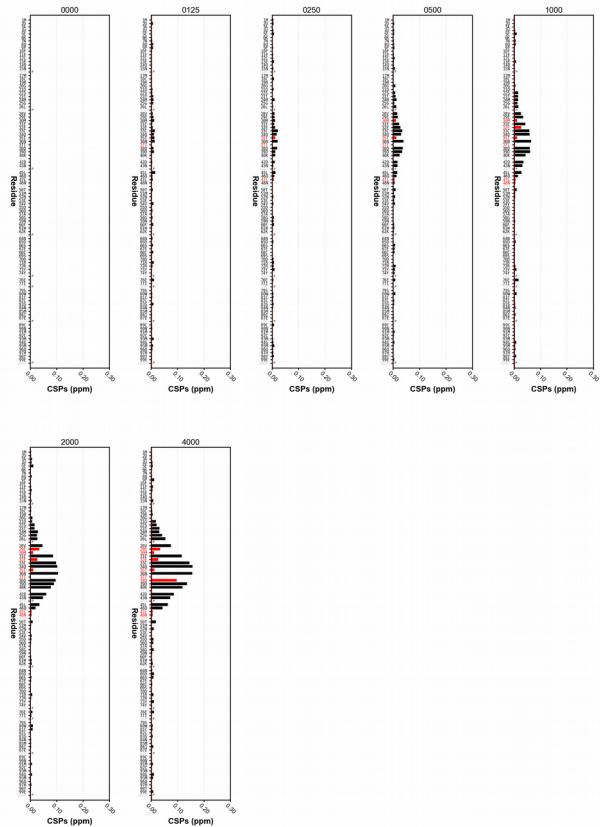


Figure 24: Full picture of a 5x2 subplot table representing the CSPs evolution along the whole series.

Residue Evolution Plots

Restraint Evolution

Residue evolution plots represent the evolution of a given restraint over the whole titration for individual residues. The generated figure is amassed into one subplot for each residue in an $M \times N$ matrix. It is designed to fit a page width figure in the *Supporting Information* of a scientific manuscript. Individual plots can be cropped externally and used in specific figures of the main article body. Data represented in this manner also can be fit to a given equation.

General features:

- allows data fitting!
- all text and labels are customisable (font type, size and style)
- X and Y ticks and scales are customisable
- customisable colours:
 - shades
 - plot colour
 - fit curve colour
- customisable lines width
- identification of unassigned and *lost* residues
- *lost* residues have no data point in plots

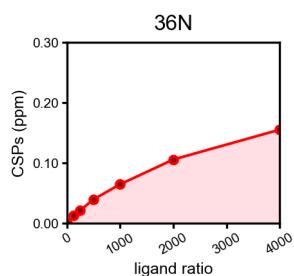


Figure 25: Subplot template of the restraint evolution representation per residue.

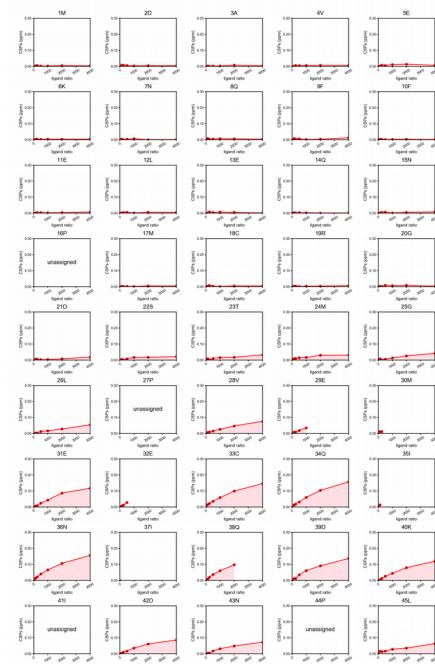


Figure 26: The full figure as generated by Farseer-NMR where a subplot is drawn for each residue.

Chemical Shift Scatter Plot

One of the most innovative plots of Farseer-NMR is the Chemical Shift Scatter plot: it translates to a plot the chemical shift evolution in two observed dimensions (generally ^1H and ^{15}N) for each residue, separately. The generated figure is amassed into one subplot for each residue in an $M \times N$ matrix. It is designed to fit a page width under the *Supporting Information* of a scientific manuscript. Individual plots can be cropped externally and used in specific figures of the main article body (Figures 27 and 28).

- all text and labels are customizable (font type, size and style)
- customisable colours:
 - colour of gradients
 - colour of shapes
 - colour of missing data points ('lost' residues)
- customisable points styles: list of ordered shapes or colour gradient circle.
- identification of unassigned and 'lost' residues
- externally configurable rules (default to 0.01 ppm) that is centred at the origin

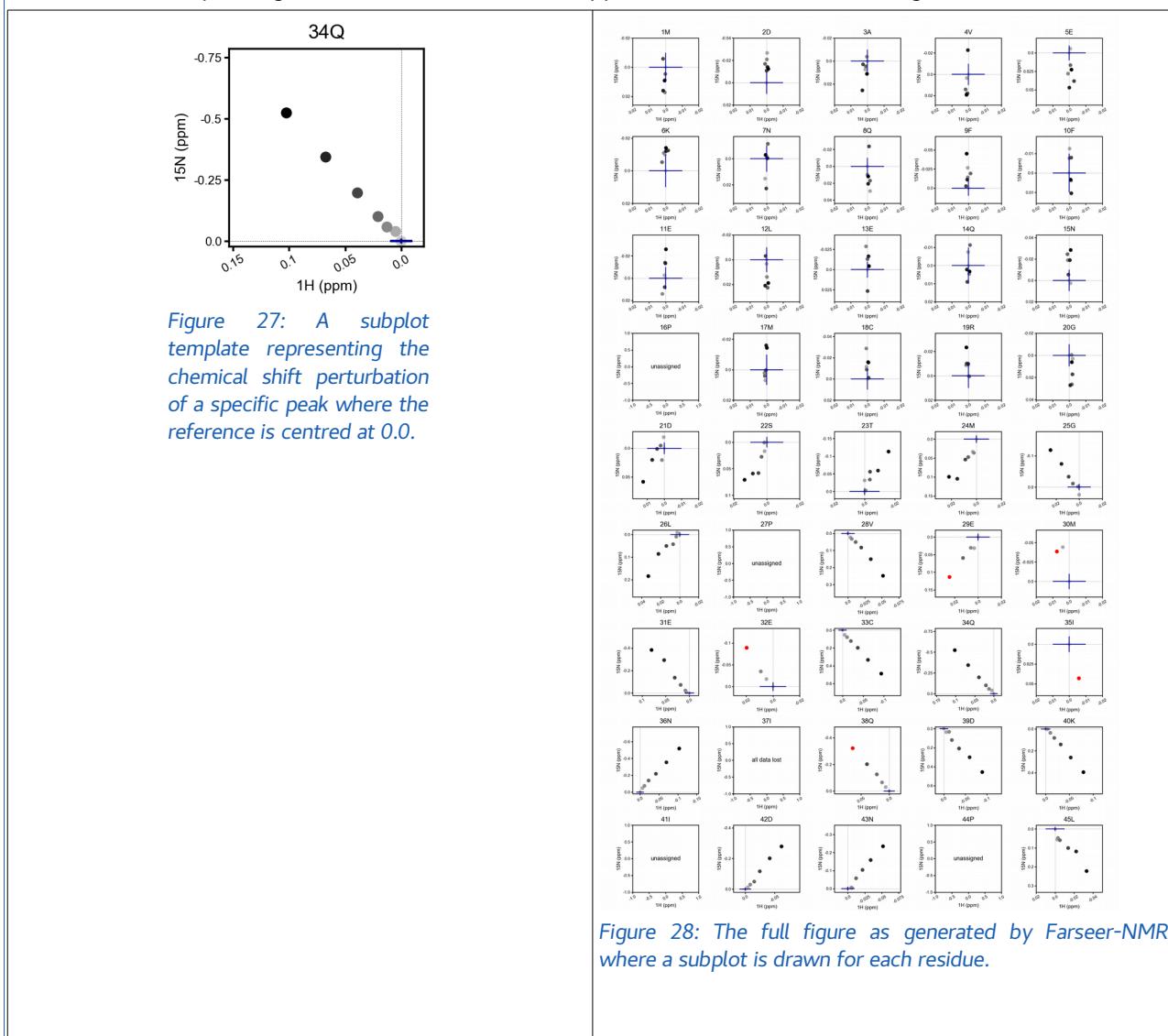


Figure 27: A subplot template representing the chemical shift perturbation of a specific peak where the reference is centred at 0.0.

Figure 28: The full figure as generated by Farseer-NMR where a subplot is drawn for each residue.

Chemical Shift Scatter Flower Plot

Following the idea of the [Chemical Shift Scatter Plot](#), the *Chemical Shift Scatter Flower* plot amasses all that information in a single plot. The spread of the chemical shifts away from the centre resemble a flower's petals, allowing easy discrimination of affected residues and grouping of them according to their changing nature(s) (Figure 29).

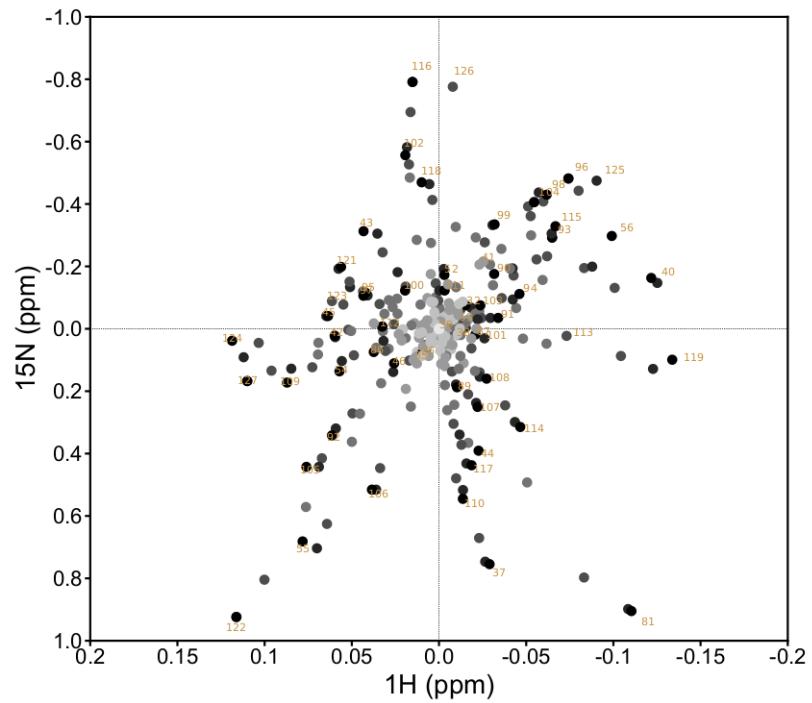


Figure 29: The Chemical Shift Scatter Flower plot.

The ΔPRE map

The ΔPRE map is generated under “comparative analysis” of cond3 (z axis) along another axis, usually x. It represents the ΔPRE scatter values for the reference experiment and a smoothed line resulting from the Gaussian filter as explained in [Arbesú et al. 2017](#), side by side the same analysis for the given data point. A gradient colour can be applied along the series to represent the progression of the data. A tag pin is placed at the paramagnetic tag position for representation. User defined regions or residues can be, respectively, shaded or highlighted.

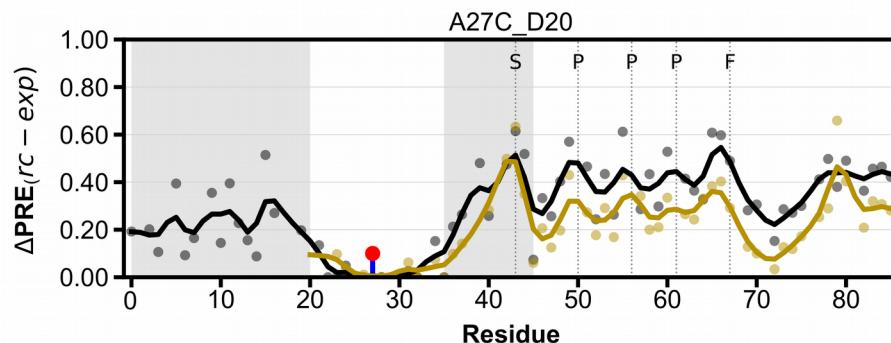


Figure 30: ΔPRE map. In black the values for the reference experiment. In gold the values for A27C_D20 construct. Regions 0-20 (Met1 is considered Met0 in this construct) and 35-45 are shaded to gray and residues 43, 50, 56, 61 and 67 are highlighted as examples. Data from Arbesú et al. 2017.

The Δ PRE heatmap

The Δ PRE heatmap represents the Δ PRE as a bar code giving for each residue a colour within a gradient where 1 is maximum relaxation enhancement and 0 is no enhancement. Results are better represented upon data smoothing with Gaussian filter.

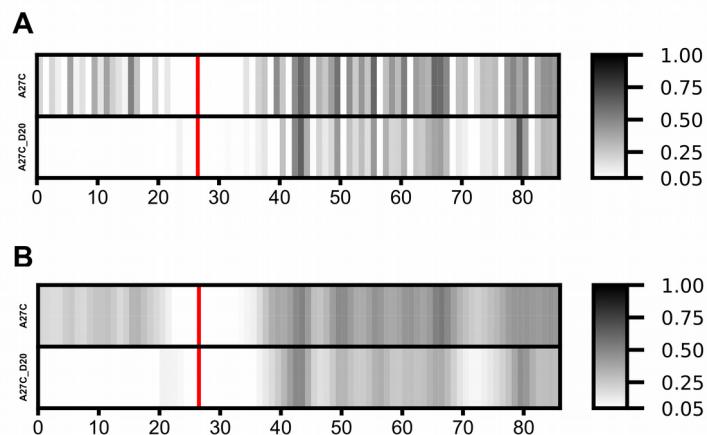


Figure 31: The Δ PRE heatmap templates. A) with raw data. B) with Gaussian smoothed data. Tag position is represented by a red line.

IV. Miscellaneous

a) Warnings, Errors and Troubleshooting (WET) list

We decided to keep maintain this as a web page because items are hyperlinked in the Farseer-NMR output messages. Visit the full page here <https://github.com/joaocteixeira/FarSeer-NMR/wiki/WET-List>.

b) Where to download

Farseer-NMR is available under a GitHub repository: <https://github.com/joaocteixeira/FarSeer-NMR>

c) Social Network

Find us on:

- Twitter [@Farseer-NMR](#)
- [Research Gate](#)
- Mailling list