



## Documentation

## Aim and resume

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

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

# Table of Contents

<b>I. Understanding Farseer-NMR.....</b>	<b>5</b>
What is Farseer-NMR?.....	6
The daily problem.....	6
The solution.....	6
How?.....	6
What else?.....	6
The Farseer-NMR data structure.....	8
The Farseer-NMR Cube.....	9
Multidimensional Analysis Workflow.....	11
A Series of NMR experiments.....	12
Navigating the Farseer-NMR Cube's Axes.....	13
How experimental series are analysed?.....	17
The Farseer-NMR Analysis Routines.....	18
The different routines stepwise.....	19
Performing Calculations.....	23
Plotting the results.....	24
Comparative Analysis.....	25
The comparative parsing.....	26
Comparing for each dimension/experimental variable?.....	29
<b>II. Technical Documentation.....</b>	<b>30</b>
Axes restrictions.....	31
Data fitting.....	32
Different sample constructs.....	33
Paramagnetic NMR Analysis.....	34
How to set a calculation Run.....	35
The initial input structure.....	35
Setting up the experimental tree – Peaklist Selection Tab.....	35
Settings Tab.....	36
<b>III. Results and output.....</b>	<b>38</b>
The log file.....	39
The results folder hierarchy.....	40
The main Run folder.....	40
The Backbone/ and Sidechains/ folders.....	41

Comparisons Folder.....	42
Reading the results.....	43
Plotting Templates.....	44
Bar Plots.....	45
Residue Evolution Plots.....	49
Chemical Shift Scatter Flower Plot.....	51
The $\Delta$ PRE Analysis.....	52
<b>IV. Miscellaneous.....</b>	<b>53</b>
Warnings, Errors and Troubleshooting (WET) list.....	54
Frequently Asked Questions.....	54

# I. Understanding Farseer-NMR

## What is Farseer-NMR?

### The daily problem

**!** Current Biomolecular NMR-related projects require thoroughly investigation of the system under study which usually translates into testing it against **multiple experimental variables** (eg. ligand concentration, ligand nature, temperature, pH, paramagnetic agents, etc...). 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**. The NMR researchers enjoy from many computational tools, available at the different stages of the NMR projects pipeline, that boost their capacity to extract the most out of the NMR experiments in a fast and reliable way. Though, there is one of the NMR-pipeline connections that is broken and a huge lack of software availability exists. That connection is the transformation of digitalized 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 sequential experiments measuring the 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 even weeks in some cases. Farseer-NMR removes the tedium, minimises the effect of human error and reduces the time burden to seconds/minutes and simplifies data visualization. Figure 1 represents the integration of Farseer-NMR into the NMR experimentalist pipeline.

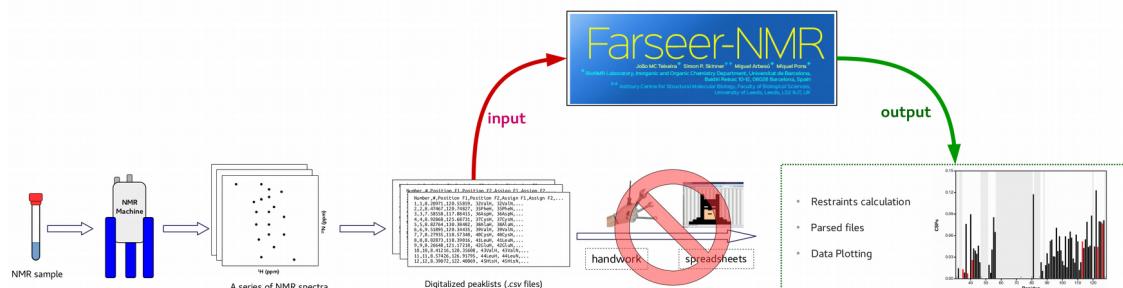


Figure 1: The BioNMR project NMR pipeline. Farseer-NMR acts upon an old and persistent gap: the transformation of digitalized peaklists into human readable and biological relevant data in the form of plots or parsed tables.

### How?

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

### What else?

**!** We have implemented the most common (and some not so common) calculation routines (PRE, PCS, 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, CYANA, XEASY, 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.

## 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** conditions (temperature and concentration) and one **discontinuous** variable (ligand nature). In total, 60 2D-NMR experiments were acquired (5x4x3). 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 is essential to have a flexible, yet simple, way to access the data which preserves the whole information and allows the deconvolution into simpler parts of those complex contributions.

## 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).

Dimensions can have any number of data points and, [as described][5], they can be accessed and combined freely to generate a panoply series that encode the answer to different experimental questions.

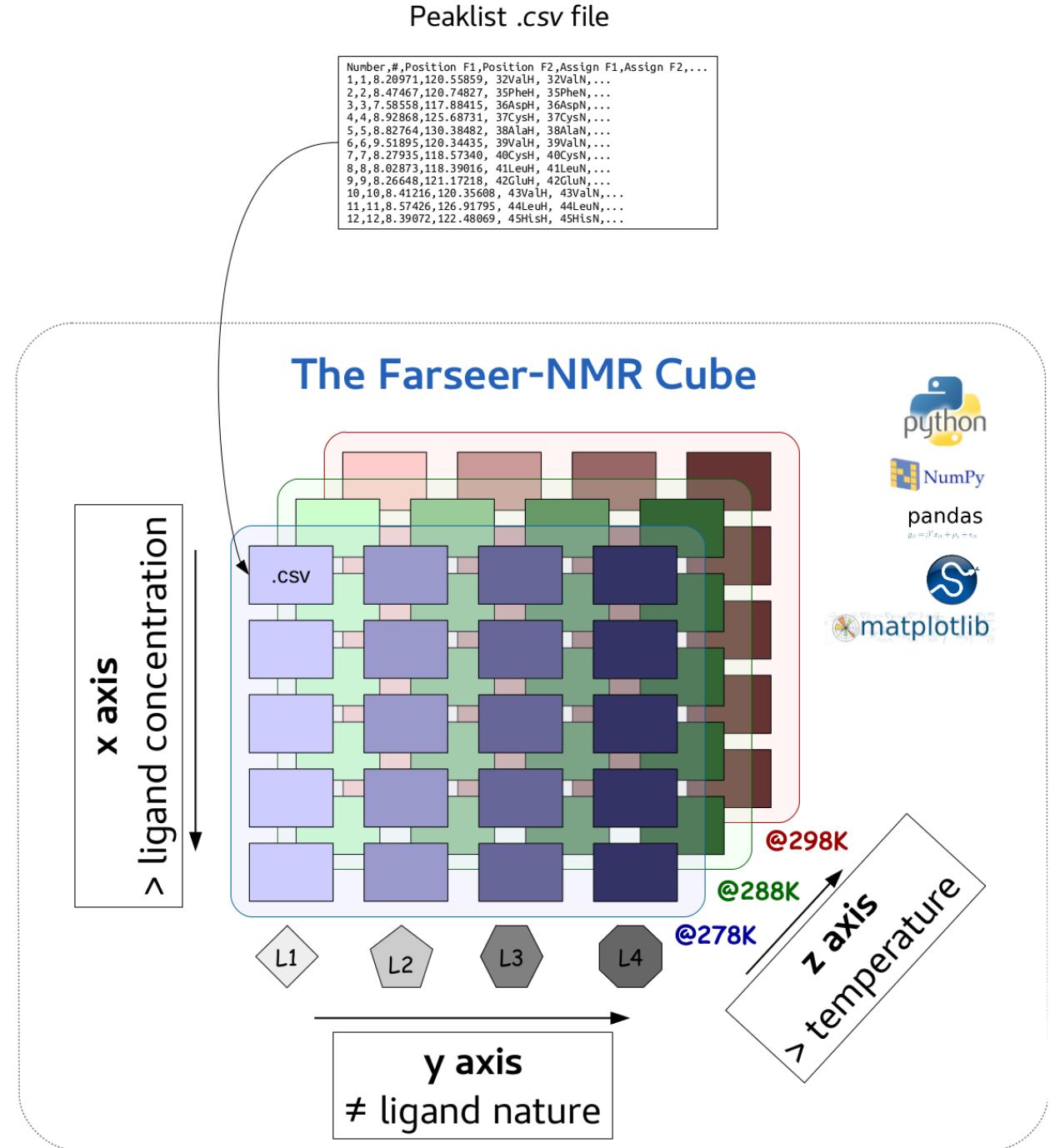


Figure 2: The schematic representation of the Farseer-NMR Cube.

## 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 that are not limited to the acquisition schedule of the multivariate data. [Following the previous example](#), we can query the dataset different questions 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 provoke 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 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 bullets **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 a progressive change of a variable - that can be *continuous* or *discontinuous* - where the first experiment is the reference to which all the others are consecutively compared to.

Following the above rationale, we can fix two points in two given axes (say 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 in ligand concentration (4x3, 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:

- 4x3 (ligand concentration)
- 3x5 (ligand nature)
- 5x4 (temperature dependence)

## Navigating the Farseer-NMR Cube's Axes

The implemented workflow sequentially generates *series of experiments* out of the Farseer-NMR's Cube that result from all possible combinations of X, Y and Z, by fixing datapoints at 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 variable by fixing two datapoints of Y and Z and creating a series of peaklists along the X data points. A full set of series is generated by walking through the Y and Z datapoints (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:

- 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?

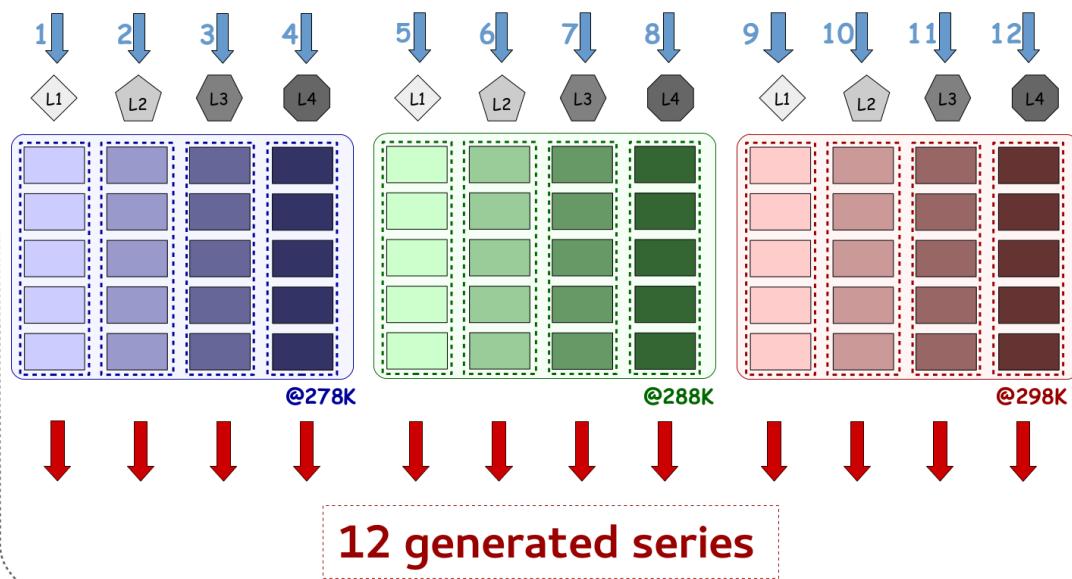


Figure 3: Representation of how the X axis series are generated along the Farseer-NMR Cube.

## 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?

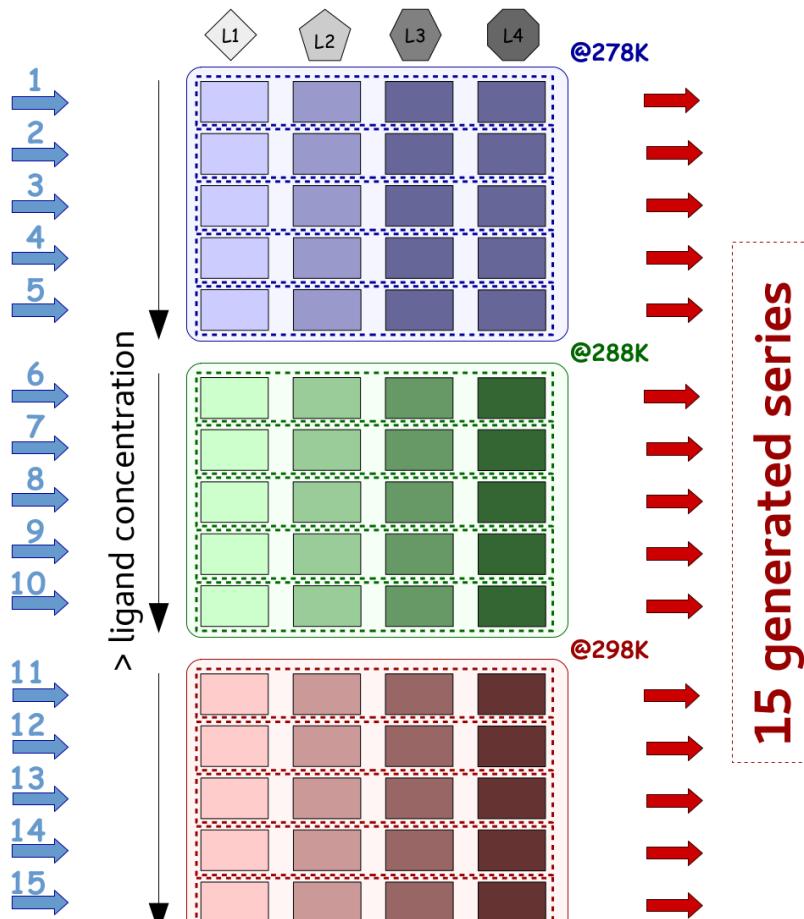


Figure 4: Representation of how the Y axis series are generated along the Farseer-NMR Cube.

## 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?

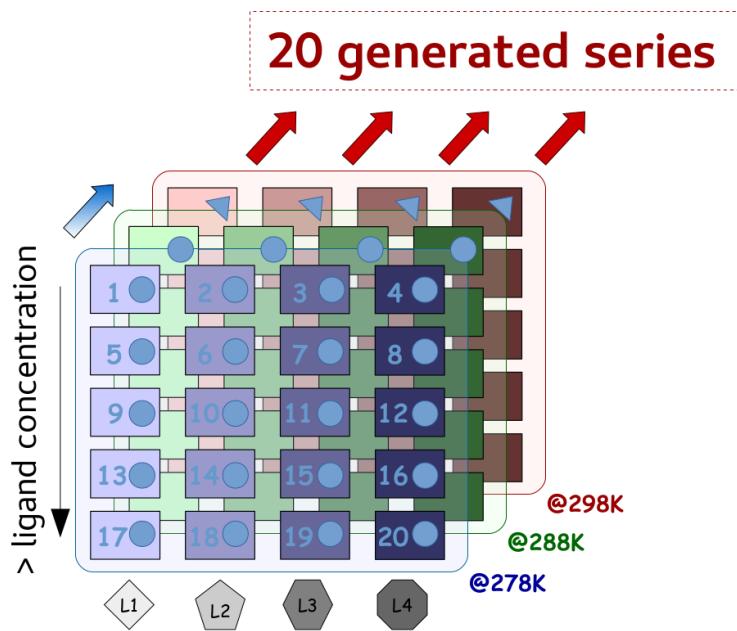


Figure 5: Representation of how the Z axis series are generated along the Farseer-NMR Cube.

## How experimental series are analysed?

Experimental series are analysed by comparing each peaklist in the series to the reference peaklist. What is the output of such comparison depends on what is being analysed, normally, calculations along series generate restraints (chemical shift perturbations, intensity ratios, etc...) that are plot 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][4]

# The Farseer-NMR Analysis Routines

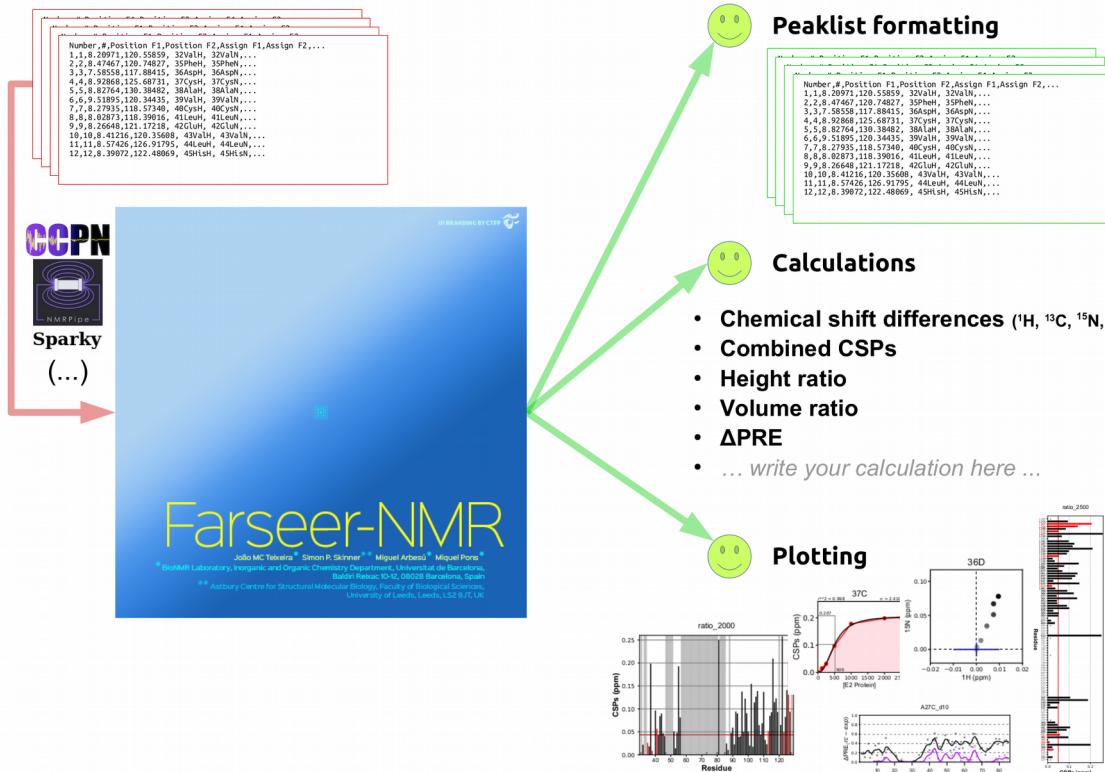


Figure 6: The schematic representation of Farseer-NMR workflow and analysis routines.

The Farseer-NMR routines workflow can be summarized in three main steps:

1. Series of two-dimensional NMR peaklists are extracted from the [Farseer-NMR's 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 organized in dedicated folders.

## The different routines stepwise

### Treating and formatting peaklists

For the sake of clarity the present section is explained at the level of this wiki page. However there is a technical consideration that should be cared about if you wish to fully understand the core of Farseer-NMR workflow.

The following subsections [Reading Assignment Information](#), [Identifying the \*lost\* residues](#), [[Identifying the \*unassigned\* residues](#)][15], are actually performed before the creation of the [Farseer-NMR Cube] and while the peaklists are still organized hierarchically in the [PeakList Tree][16].

Therefore, the following routines serve two purposes:

1. the scientific relevant identification of *lost* and *unassigned* residues
2. adjust all peaklists loaded to the same size (rows), because comparing peaklists of different size is technically not possible as row index won't match.

And for this reason they are performed prior to the creation of the Farseer-NMR Cube. On theoretical terms, the rational can be that which considers these routines to be applied for each series. On 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 [Expand lost residues][12] flags. This option is specially interesting when analysis paramagnetic data where 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 here][11], different .fasta files can only be input along the Y axis.
  - a. again, this can be overcome by activating the corresponding flag in [Expand lost residues][12] menu.

## Reading assignment information

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

In the Farseer-NMR preferred peaklist format ([CCPNMR v2 format][10]), the assigned information is concatenated in a single column (*Assign F1* in the format *1MetH*), it is therefore necessary to split this information first. Three additional columns are created (*Res#*, *1-letter*, *3-letter*) to store information on the residue number, 1-letter and 3-letters aminoacid codes, this information is used to index all the data and resulting calculations (figure 7).

The diagram illustrates the transformation of a single column into three separate columns. On the left, there is a table with one column labeled "Assign F1". This column contains three entries: "0GlyH", "1MetH", and "2AspH". An arrow points from this table to the right, indicating the expansion process. On the right, there is a larger table with three columns: "Res#", "1-letter", and "3-letter". The "Res#" column contains the values "0", "1", and "2". The "1-letter" column contains the values "G", "M", and "D". The "3-letter" column contains the values "Gly", "Met", and "Asp".

Assign F1
0GlyH
1MetH
2AspH

→

Res#	1-letter	3-letter
0	G	Gly
1	M	Met
2	D	Asp

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

## Identifying the lost peaks

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

**Technical note:** The difference in peaklists size greatly difficults the direct and straightforward analysis of the peaklists files in traditional plotting tools, because *rows identity won't match when comparing row by row*. This issue had previously to be handled manually, while now Farseer-NMR does it automatically.

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

### Adds entries for the missing peaks/residues

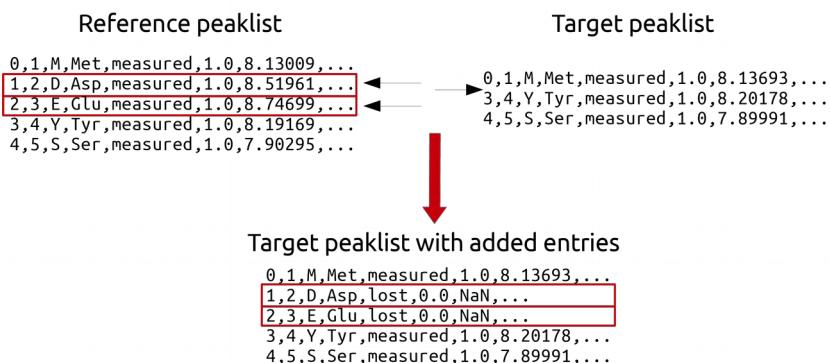


Figure 8: New rows are added to identify the peaks that were lost with respect to the reference experiment.

## Identifying the unassigned residues

This feature is optional and is performed similarly to the identification of the *lost* residues with the exception that all the peaklist 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

```
MDEYSPKRHDVAQLKFLCESLYDEGIATLGDSHHGWWNDPT
SAVNQLNLDLIEHIASFVMSFKIKYPDDGDLSELVEEYLDDTY
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 indentify 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 the reference experiment and calculates the user-requested restraints. The results are added to newly generated columns in the peaklists pandas.DataFrame, which are exported altogether at the end of the run. Farseer-NMR can calculate:

1. Chemical shift differences for each nuclei
2. [Combined Chemical Shift Perturbations (CSP)][8]
3. Intensity ratios
4. [Data Fitting][9]
5. [ $\Delta$ PRE][4]

## Plotting the results

Farseer-NMR contains several [publication-quality plotting templates][5] to represent the calculated data. For each series analysed, the calculated restraints can be plot in any and every template available. Each template is highly customizable and aims at adapting the data representation to the different publication requirements.

## Comparative Analysis

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

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

### Example 2:

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

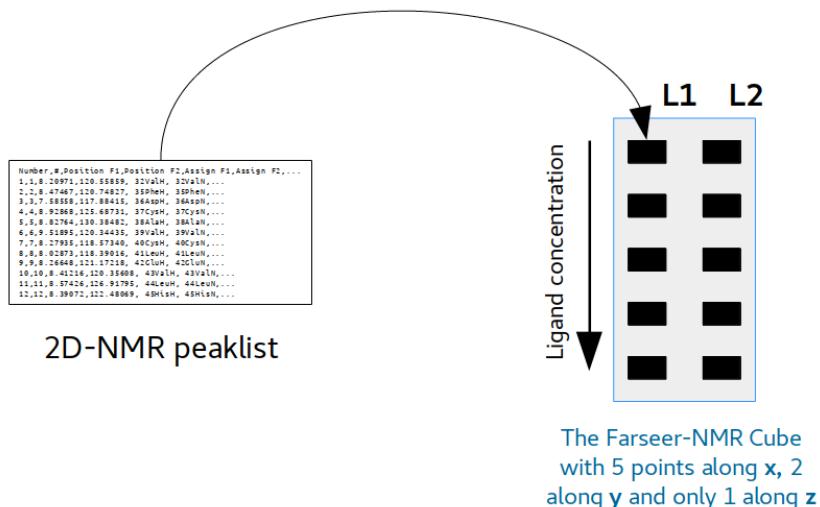


Figure 10: An example of two experimental series stacked along Y axis.

the **z** dimension consists of a single datapoint (298K), and therefore can be disregarded (though technically it exists). Lets consider the user wants to analyse the two experimental series for **L1** and **L2** only along the **x** axis of the Cube, that is, along the increase of ligand concentration. In this case, two folders containing the respective results for **L1** and **L2** are generated ([further reading on folder organization][3]) containing all the tables and plots required by the user ([analysis routines][4]).

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

This requirement could be fulfilled if the user simply opens the two plotting figures (or hard prints) and places them side by side for comparison. This manual procedure may become awkward if one needs to compare large experimental datasets where several points for a particular variable were acquired: say it is a ligand screening of a 20 ligand library.

## The comparative parsing

Following the above example, the generated data can be compared along the **y** axis. It cannot be compared along the **z** axis though (temperature), because there is only one point in this variable: *there is nothing to compare with*.

## 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 obtained **previously** along the **x** dimension (ligand concentration). Figure 11 aims at solving any confusion. **Important:** Comparing along **y** is different from analysing an experimental series along the **y** axis!

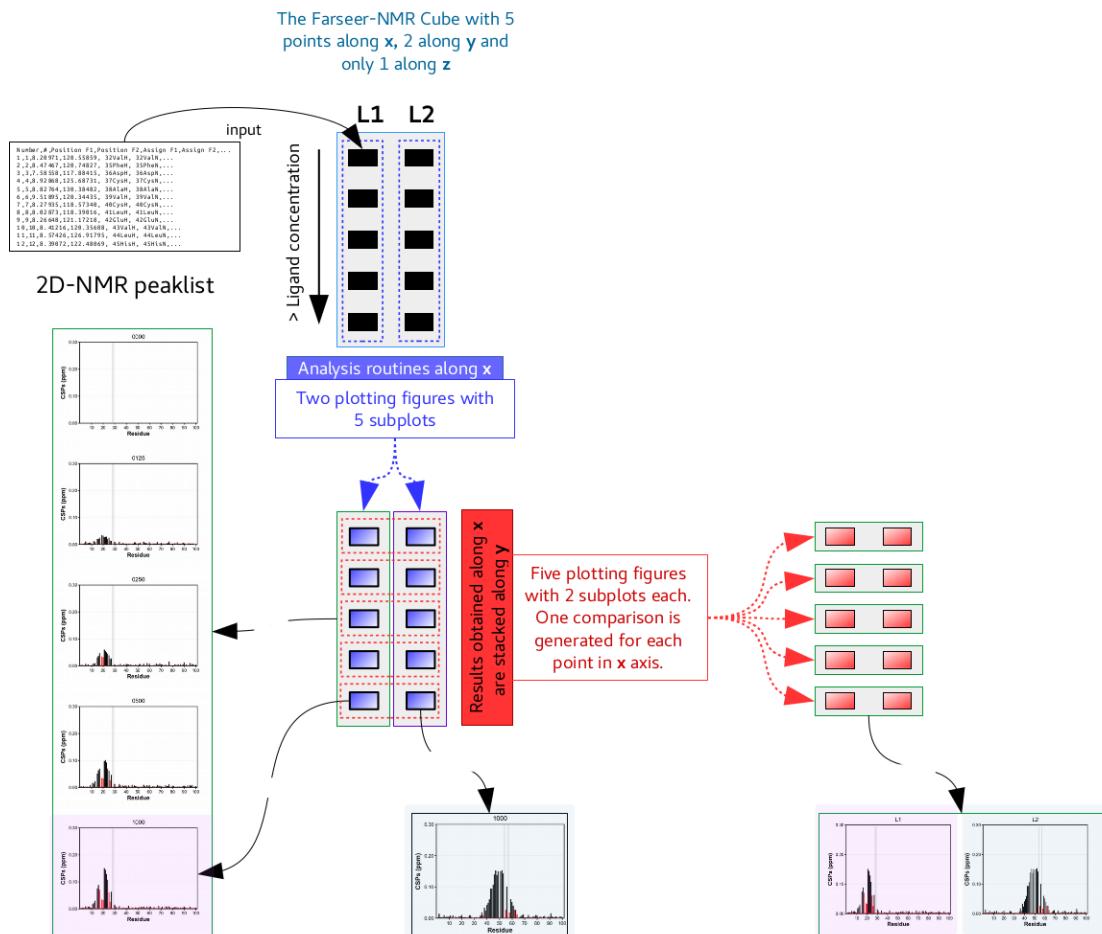


Figure 11: Schematic representation of how results generated along the X axis are compared along the Y axis.

## Calculating along Y and comparing along X

Similarly to the above, subplots resulting from calculations along the y axis are stacked along X.

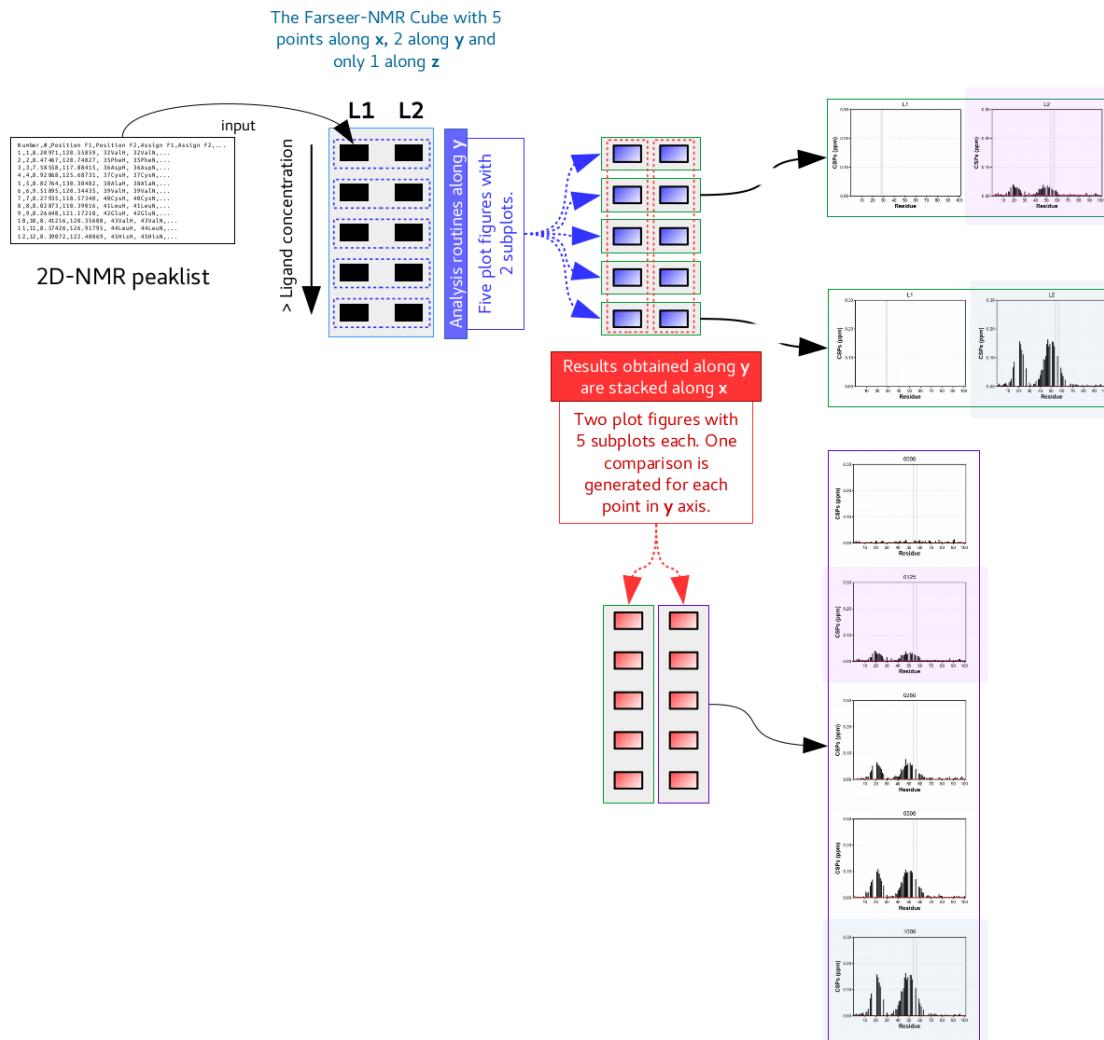


Figure 12: Schematic representation of how results generated along the Y axis are compared along the X axis.

## 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 that investigate the system's dependence on 3 different variables – Farseer-NMR Cube with dimensions 5x4x3.

We have seen that in total 4x3 series can be analysed along the x axis. Applying a comparative analysis along the y axis will generate 5x3 new sets of parsed data (not new data). Consistently, comparisons performed along the z axis sum a total of 5x4 sets of parsed data.

## **II. Technical Documentation**

## Axes restrictions

Farseer-NMR loads all the peaklist data into a 5-dimensional array that can be thought of as a cube made of 2-dimensional datapoints – the [Farseer-NMR's 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.

**Farseer-NMR has been designed to enable any kind of variable to be analysed**, however, come 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 respective section].

Parameter fitting is only available along the **x** axis. It is possible to set a concentration range series along **y** and **z** and perform all the [analysis routines](#), 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 5x4 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 on the data points of the **y** axis. Consequently, swapping the tow 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 were 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 datapoints of a second axis where the *concentration range* would fit 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 analysing along the *z* axis. Therefore, when analysing paramagnetic NMR data, e.g. PREs, input data should be organized such that the first point in the *z* axis corresponds to the *diamagnetic* series and the second point to the *paramagnetic* series.

See [here] for comprehensive descriptions of the Paramagnetic NMR analysis routines available.

## How to set a calculation Run

### The initial input structure

Please read the previous section about [Farseer-NMR Cube](#). As explained in [Multidimensional Analysis Workflow](#), Farseer-NMR performs complex analysis on NMR data by generating series of experiments from permutation of the experimental variables. However, to initiate the Farseer-NMR Run, and hence the Farseer-NMR Cube, data should be input in a hierachic fashion. A logical tree of experiments should be created where the top hierarchy level corresponds to the z axis of the Farseer-NMR Cube, followed by the y axis datapoints and finally the x axis datapoints.

### Setting up the experimental tree - Peaklist Selection Tab

Examples in pictures

## Settings Tab

All the settings can be configured directly in the Settings tab of the Farseer-NMR user interface. Settings are categorised in individual boxes according to their nature. Additional, and specific, settings may be available by pop-up menus. Box description is listed below.

### *Spectrum Path*

### *Calculation Output Folder*

### *Experimental Series Analysis*

#### *Expanding lost residues to other dimensions*

#### *Sidechains*

#### *Chemical Shift Normalization*

#### *Using FASTA files*

#### *Restraint Calculation*

#### *CSP Specific Settings*

#### *Series Plotting*

#### *Residue Evolution Plotting*

#### *PRE Analysis*

#### *Figure*

## **Load/Save/Run Buttons**

## 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

It is explained that Farseer-NMR can [load different and multiple .fasta files](#) to compare parameter evolution along different mutants.

However, to perform analyses along the **y** axis with different .fasta files Farseer-NMR requires that .fastas have the same length, otherwise row numbers won't match when comparing peaklists. If no analysis is to be performed along **y**, and only several constructs want to be analysed along **x**, there is no requirement to maintain fastas to the same length.

## Others

### The .fasta file

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

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

## **III. Results and output**

## The log file

Farseer-NMR prints to the `Terminal` the progression of the run showing in detail all the performed operations. At the end, the full log is exported to an external file in [Markdown syntax](#).

## The results folder hierarchy

### The main Run folder...

... is the folder that contains the *spectra/* folder (which contains all the input peaklists) and is where where all the Farseer-NMR generated output will be stored:

1. copy of the version used
2. copy of the user defined variables
3. log file
4. folder 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, by 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...

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

## The conditions subfolders...

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

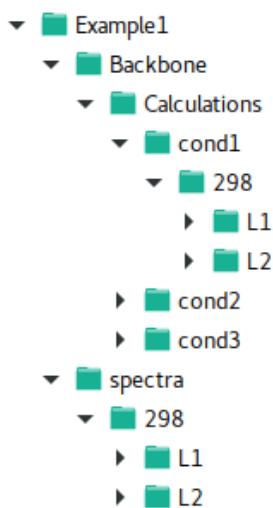
## The different series

The series generated for each axis (*cond\** subfolder) are hierarchically stored according to the data points fixed for each of the other two axes, where the previous axis is parent to the next axis, where the next axis of X is Y and the previous is Z.

## Some examples:

- The analysis generated for 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 13: An example of how results for the different dimensions are organized under the Farseer-NMR data structure.*

## Comparisons Folder

*Comparisons/* folder is present at the same hierarchical 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, named *C1*, *C2* or *C3*, respectively to the *X*, *Y* and *Z* axes. 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 under these folder hierarchy you will find the that series are organized in the same scheme as described above for the *Calculations/* folder. The *Backbone/Comparisons/C1/cond2/* structure is followed by a *<previous axis>/<next 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 datapoints along *X* axis and *298K* is the other datapoint long the *Z* axis. This series runs along the *Y* axis data points.

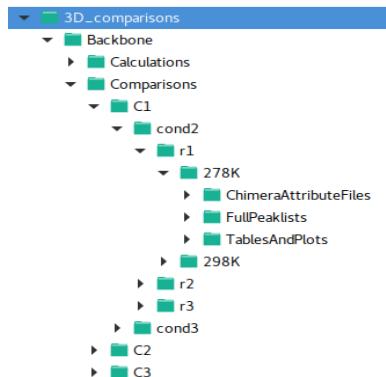


Figure 14: An example of how comparative analyses are organized under the Farseer-NMR data structure.

## Reading the results

Inside each titration result folder you will find different subfolders that organize the output results data: tables, parsed files and plots.



Figure 15: The folders where results from the Farseer-NMR Analysis routines are stored.

The **FullPeaklists/** folder store the parsed peaklist that constitute the titration analysed in *tab separated files*. These peaklists have the same information originally input for the calculation plus additional 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 aminoacid code*, the *3-letter aminoacid code* and a *Peak Status* information.
4. A column for each of the calculated restraint.

The **TablesAndPlots/** subfolder stored the plots drawn. One subfolder is created for each restraint calculated (*H1\_delta*, *15N\_delta*, *CSP*, *Height\_ratio*, ...). Inside each subfolder there is a figure file for each plotting template drawn and a *.tsv* file with the calculated restraints data used for representing those plots. See here the list of [all the plotting templates available].

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.

## Plotting Templates

Farseer-NMR contains a set of plotting templates that represent the calculated data in an organized, simple and **publication-ready** manner. There plots that represent commonly used styles and other which we have specially implemented and developed to improve data representation.

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

The structure of the figures, subplots organization in columns and rows, colours, font types and several other plotting style options are highly customizable under the corresponding GUI menu or the farseer\_user\_variables.py file.

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

## Bar Plots

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

### General features:

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

## Compacted Bar Plot

The compacted bar plots are designed to fit half-page width in a scientific reviewed publication and are generally drawn in an overall figure of cols per rows matrix of subplots (figures 16 and 17).

### Specific features:

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

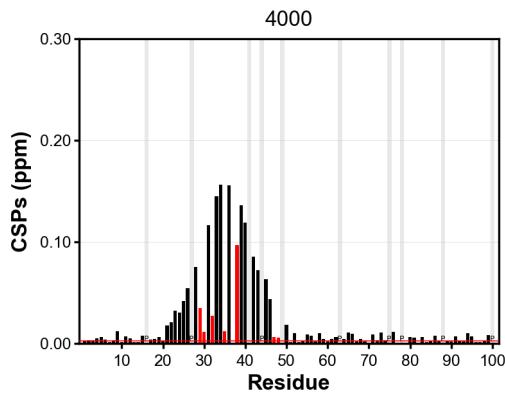


Figure 16: Compacted Bar template subplot

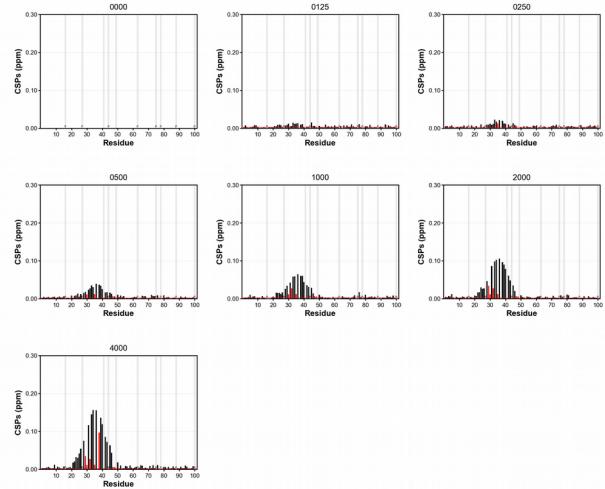


Figure 17: Full picture of a 3x3 subplot table representing the CSPs evolution of the whole series.

## Extended Bar Plot

The extended bay plot is designed to fit whole page width in a scientific reviewed publication and are generally drawn in an overall figures vertically stacked subplots representing the titration evolution (figures 23 and 24).

### Specific features:

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

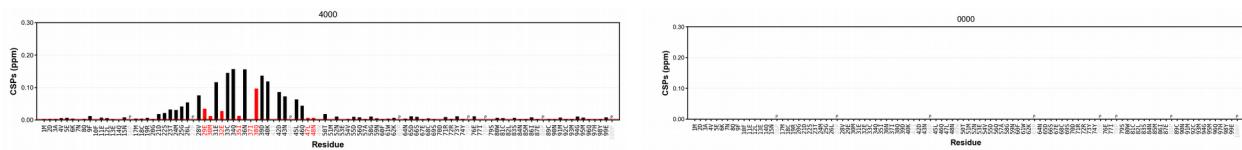


Figure 18: Extended Bar template subplot

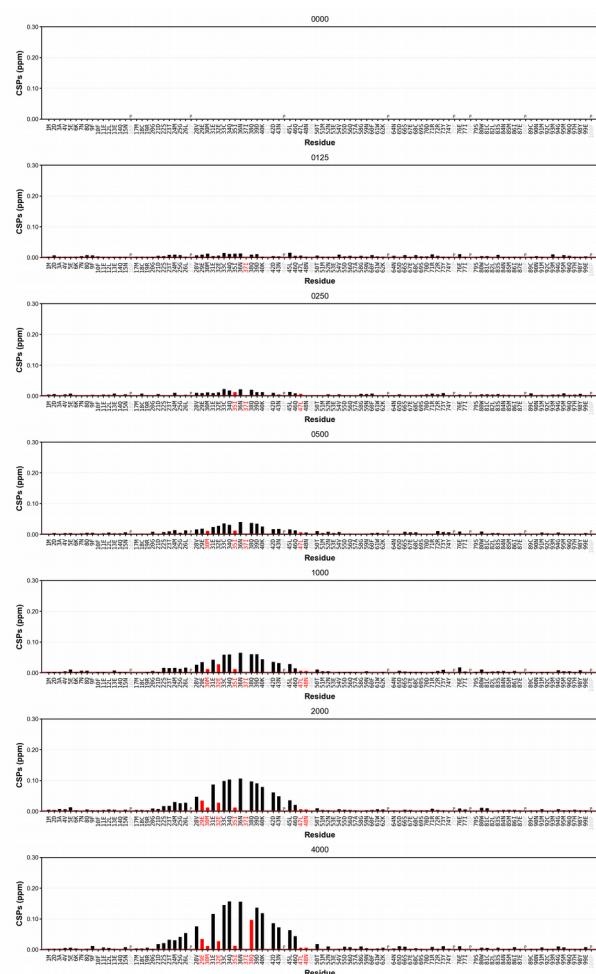
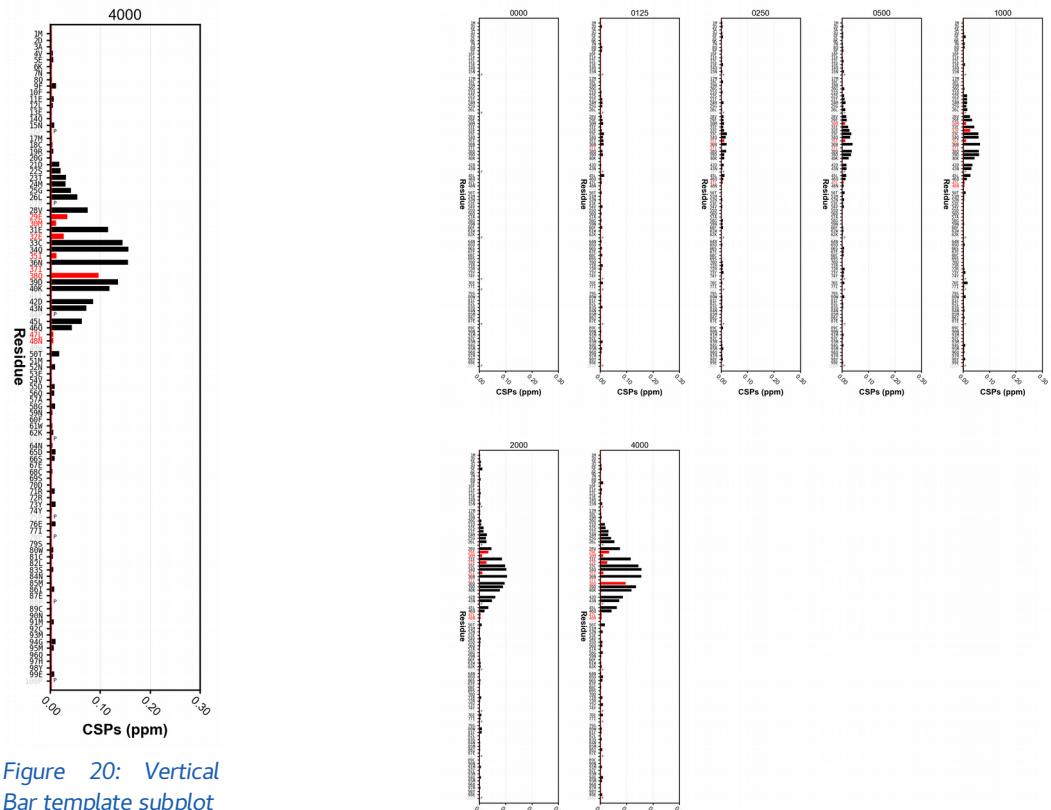


Figure 19: Full picture of a 7x1 subplot table representing the CSPs evolution along the whole series.

## Vertical Bar Plot

The vertical bar plot is designed to fit narrow spaces and column organization styles in a scientific reviewed publication and are generally drawn in an overall figures of horizontally stacked subplots representing the titration evolution (figures 20 and 21).



## Residue Evolution Plots

## **Restraint Evolution**

Residue evolution plots represent the evolution of a given restraint over the whole titration for each individual residue. The generated figure is amasses one subplot for each residue in a  $M \times N$  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. The data represented this manner can be fit to a given equation.

## General features:

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

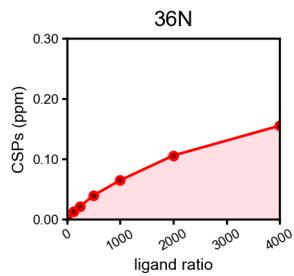


Figure 22: Subplot template of the restraint evolution representation per residue.

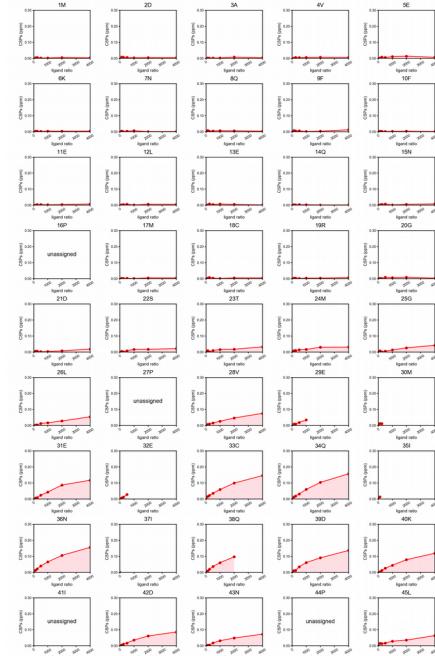
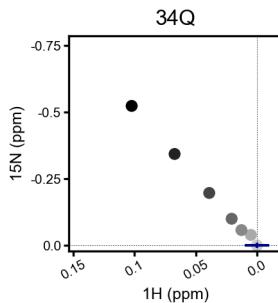


Figure 23: The full figure as generated by Farseer-NMR where a subplot is drawn for each residue.

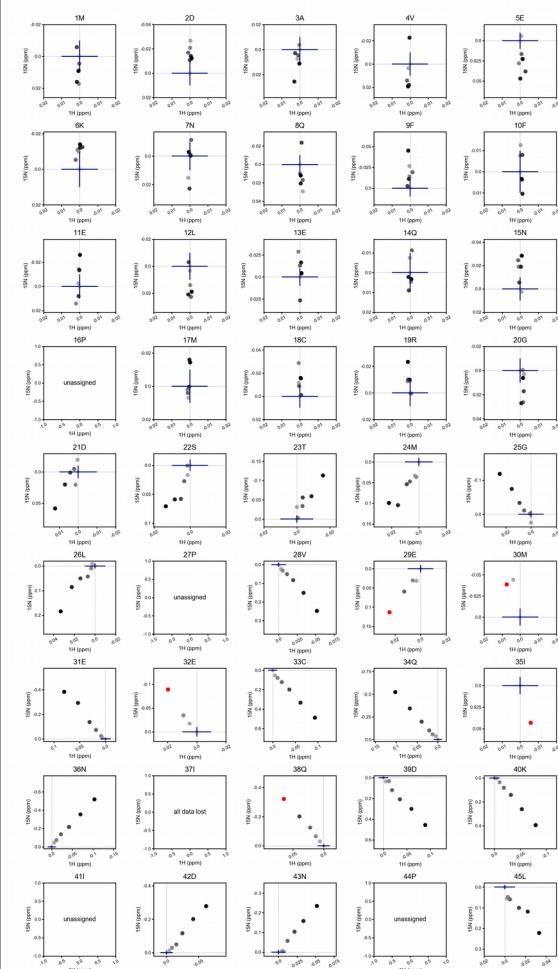
## Chemical Shift Scatter Plot

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 the two observed dimensions (general  $^1\text{H}$  and  $^{15}\text{N}$ ) for every residue separately. The generated figure is amasses one subplot for each residue in a  $M \times N$  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 24 and 25).

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



**Figure 24:** A subplot template representing the chemical shift perturbation of a specific peak where the reference is centered at 0.0.



**Figure 25:** 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 *Flower* plot amasses all that information on a single plot. The spreading of the chemical shifts away from the centre resemble a flower petals, allowing to easily discriminate those affected residues and to group them according to their changing nature (figure 26).

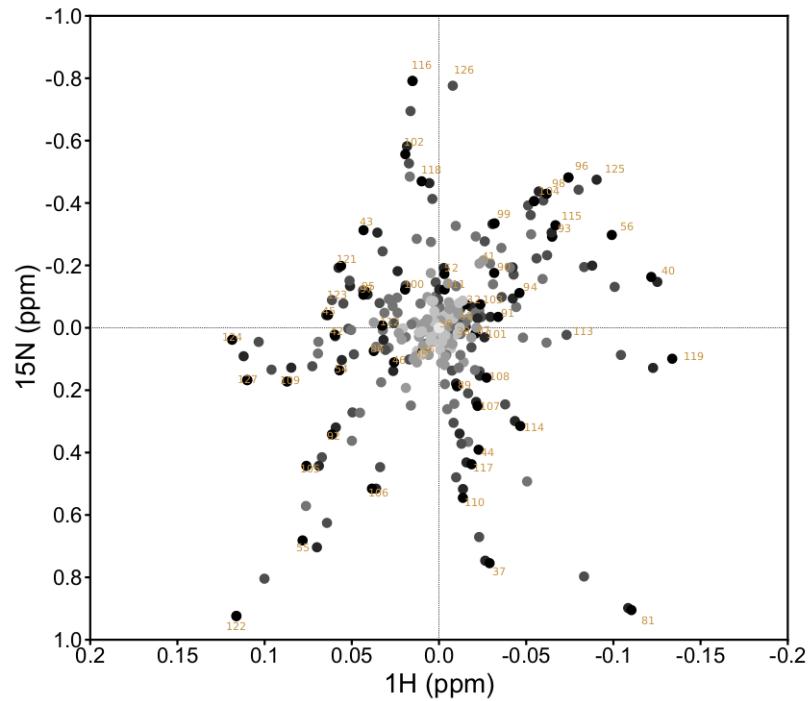


Figure 26: The CSPs Scatter Flower template.

## The $\Delta$ PRE Analysis

## IV. Miscellaneous

## Warnings, Errors and Troubleshooting (WET) list

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

## Frequently Asked Questions

This page is updated in <https://github.com/joaomcteixeira/FarSeer-NMR/wiki/FAQ>.