# MultiNMRFit

***Release 2.0.0***

**Pierre Millard, Cyril Charlier**

**Mar 28, 2024**

# USAGE

CHAPTER

ONE

# WELCOME TO MULTINMRFIT DOCUMENTATION!

**MultiNMRFit is a scientific software dedicated to the analysis of NMR data**. It is one of the routine tools that we use at the NMR team and MetaSys team of Toulouse Biotechnology Institute.

The code is open-source, and available on GitHub under a *GPLv3 license*.

This documentation is available on Read the Docs (https://multinmrfit.readthedocs.io) and can be downloaded as a PDF file.

**Key features**

- **fit series of 1D spectra** (acquired as individual 1D spectra, as a pseudo 2D spectrum, or provided as tabulated text files),
- can be used with **all nuclei** ($^{1}$H, $^{13}$C, $^{15}$N, $^{31}$P, etc),
- estimation of several parameters for each signal of interest (**intensity**, **area**, **chemical shift**, **linewidth**, **coupling constant(s)**, etc),
- **semi-automated analysis** for **peak picking** and **definition of multiplicity** for each signal,
- account for **overlaps** between peaks and **zero-order baseline correction**,
- **visual inspection of the fitted curves**,
- estimation of **uncertainty** on estimated parameters (standard deviation),
- shipped as a **library** with a **graphical user interface**,
- open-source, free and easy to install everywhere where Python 3 and pip run,
- biologist-friendly.

### See Also

We strongly encourage you to read the *Tutorial* before using MultiNMRFit.

## 1.1 Quick start

### 1.1.1 Installation

MultiNMRFit requires Python 3.8 or higher. If you do not have a Python environment configured on your computer, we recommend that you follow the instructions from Anaconda.

Then, open a terminal (e.g. run *Anaconda Prompt* if you have installed Anaconda) and type:

```
pip install multinmrfit
```

You are now ready to start MultiNMRFit.

If this method does not work, you should ask your local system administrator or the IT department "how to install a Python 3 package from PyPi" on your computer.

#### Alternatives & update

If you know that you do not have permission to install software systemwide, you can install MultiNMRFit into your user directory using the `--user` flag:

```
pip install --user multinmrfit
```

If you already have a previous version of MultiNMRFit installed, you can upgrade it to the latest version with:

```
pip install --upgrade multinmrfit
```

Alternatively, you can also download all sources in a tarball from GitHub, but it will be more difficult to update MultiNMRFit later on.

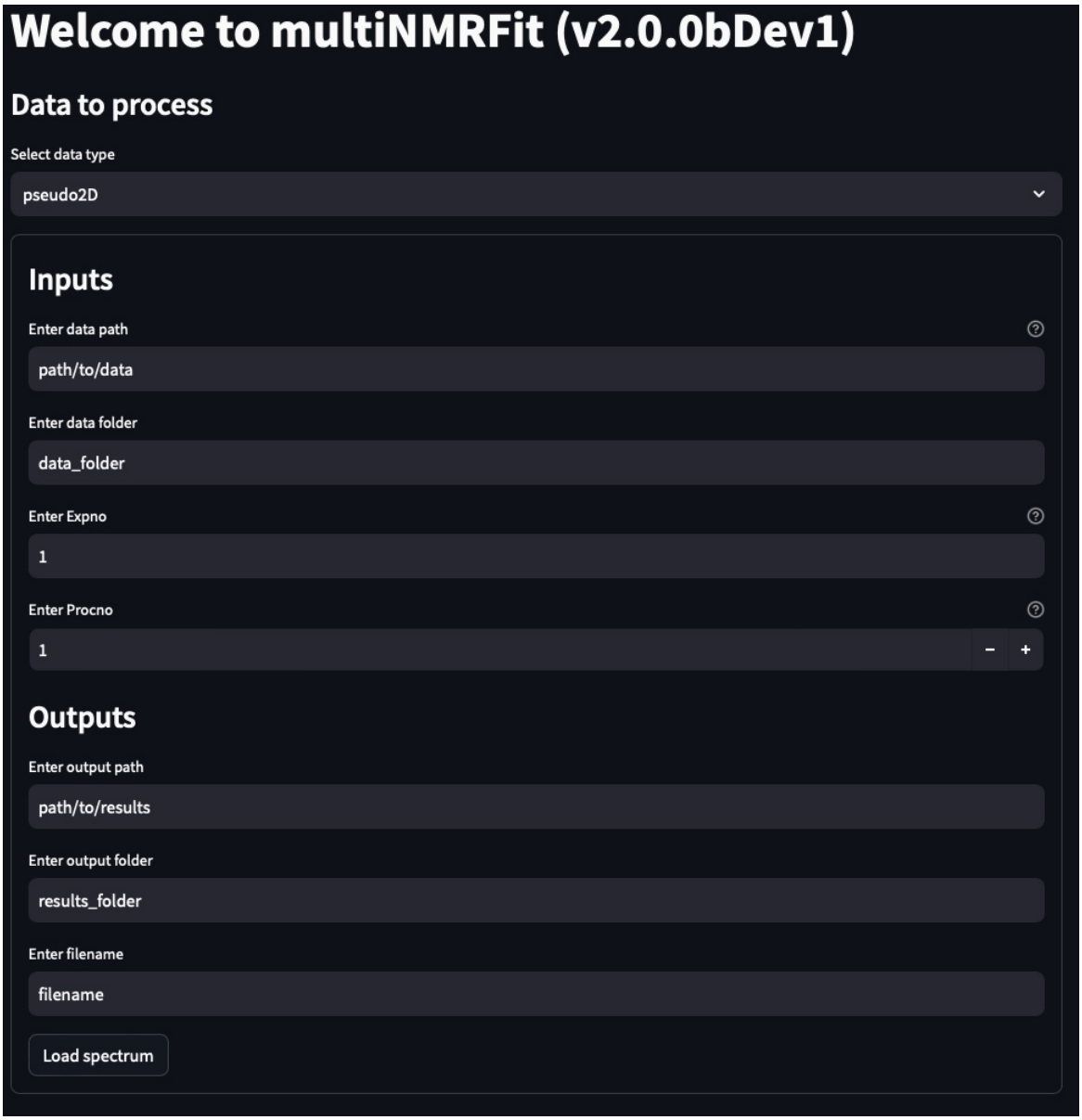
### 1.1.2 Usage

#### Graphical User Interface

To start the Graphical User Interface, type in a terminal (Windows: *Anaconda Prompt*):

```
nmrfit
```

The MultiNMRFit window will open. If the window fails to open, have a look at our dedicated troubleshooting procedure to solve the problem.




The main processing steps can be performed via the menu on the left side bar:

- **Inputs & Outputs**: information required to load the data to process and export results (type of data, input and output directories, etc)
- **Process spectra**: process one or several signal(s) of specific spectra
- **Process from reference**: process a serie of spectra as done on a given spectrum (used as reference)
- **Results visualisation**: view and export processing results

Details on MultiNMRFit usage can be found in the tutorial section.

---

**Note:** The process is continuously and automatically saved as a pickle file in the output folder. To reopen the current processing state, just reopen this file by clicking on "Load a processing file - Browse files" on the side bar at the left.

---

**Warning:** MultiNMRFit silently overwrites (results and processing) files if they already exist. So take care to copy your results elsewhere or to change the output path and/or filename if you want to protect them from overwriting.

#### Library

MultiNMRFit is also available as a library (a Python module) that you can import directly in your Python scripts:

```
import multinmrfit
```

## 1.2 Tutorial

**See also:**

If you have a question that is not covered in the tutorials, have a look at the faq or please contact us.

This tutorial will guide you through the different pages of MultiNMRFit interface.

### 1.2.1 Inputs & Outputs

#### Data type

MultiNMRFit assumes that all the processing (base line correction, phasing, ...) is performed prior its usage. MultiNMRFit can load 1D NMR data in 3 formats:

- **Pseudo2D**: pseudo2D experiment (Bruker format only),
- **list of 1Ds**: list of 1Ds acquired independently (Bruker format only),
- **txt data**: data from a text tabulated file (:file:'.txt' extension) with the following structure:

| ppm | 0 | ... | n |
|---|---|---|---|
| 0 | 1.2e3 | ... | 1.2e6 |
| 0.1 | 1.3e3 | ... | 4e7 |
| 0.2 | 2e8 | ... | 3.6e3 |
| ... | ... | ... | ... |
| 12 | 3e4 | ... | 7.85e3 |

The column **ppm** is mandatory and contains the ppm scale assumes to be same for all spectra. The follwing columns here names **0** to **n** correspond to each individual spectra that will be loaded into MultiNMRFit

**Note:** **list of 1Ds**: The list of experiments should be provided as * 1,8,109 : for non-consecutive * 1-5 : sequential se experiments (resulting in 1,2,3,4,5) * 1-5,109 : for incomplete series (resulting in 1,2,3,4,5,109)

**Warning:** **list of 1Ds** All the data needs to have the same number of points (**TD**) and the ppm scale identical. If data were processes with different **SR** parameters in TopSpin it might shift one dataset to another. The ppm scale will be taken from the first experiment in the list.

### Inputs/Outputs

**data_path**
Path to the directory that contain the data

**data_folder**
Folder containing your NMR data

**expno**
List of experiments used in the MultiNMRFit analyis

**procno**
Process number (e.g. procno in Topspin)

---

**Note:** **Inputs**: The different fields will for inputs as described above will appear only for data type (**Pseudo2D** & **list of 1Ds**) For **txt data**, the text file must be loaded using the drag-and-drop menu.

---

---

**Note:** **procno**: If a list of **expno** is provided the **procno** needs to be same for all the **expnos**.

---

**output_path**
path to the folder use to export the outputs

**output_folder**
folder with the outputs

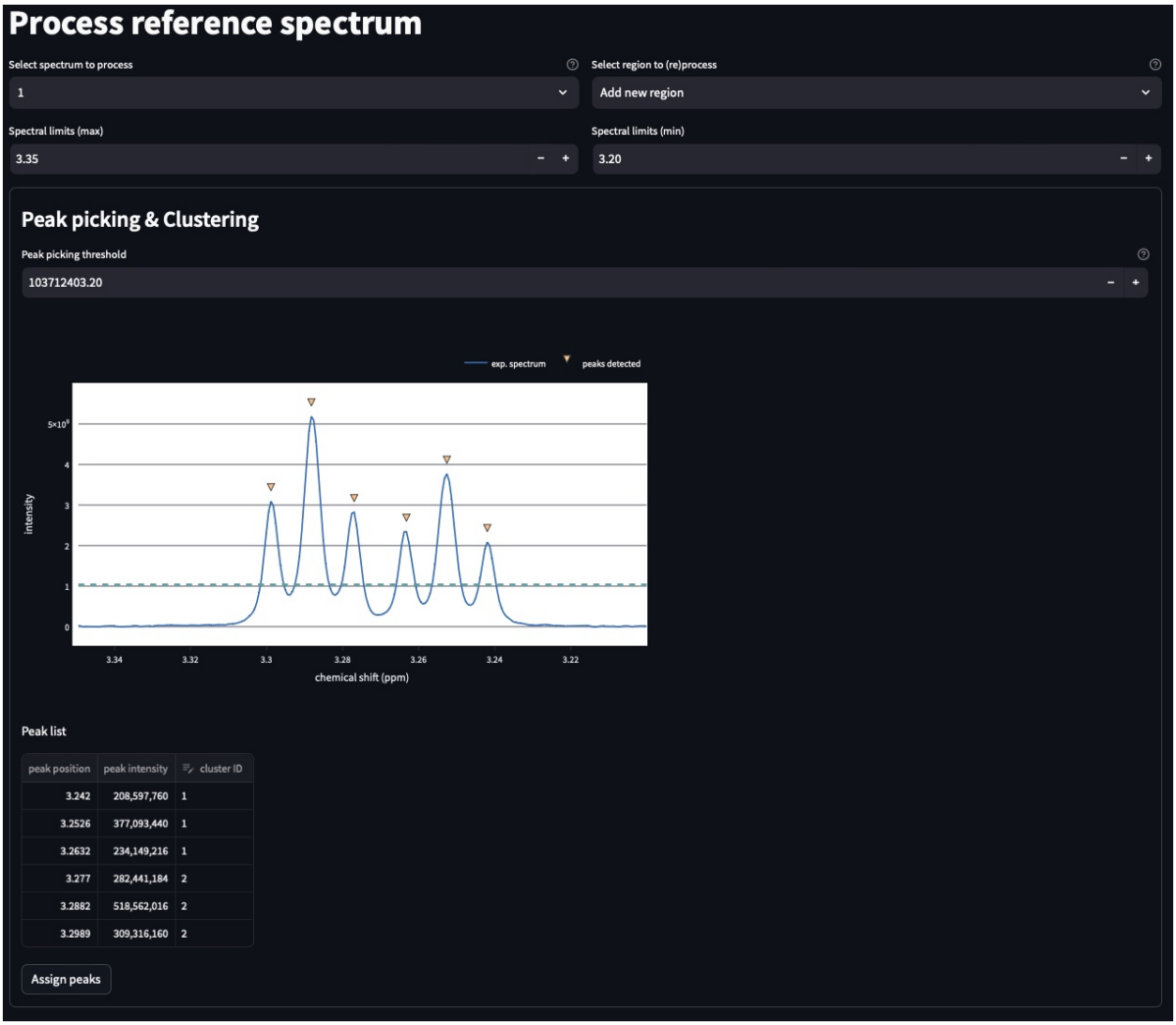
**filename**
name of the pickle file containing the process that will be automatically saved through the workflow.

### Load a processing file

Along the way the process is saved in a pickle format containing the entire process that was perfomed. The pickle file can be loaded using the drag-and-drop menu available in side bar of the Inputs & Outputs page.

Once you are ready to load the spectrum, clicked the **Load Spectrum** buttom.

Once the data are correctly loaded the second page of the interface becomes available and allows use to perform the fitting of the reference spectrum:




The top part of this page automatically performs the peak picking on the reference spectrum within the region displayed in the graph: * **Select reference spectrum**: Select one the spectrum of the list. Tis specturm (called reference spectrum) will be used for automatic peak detection and initial fitting. * **Select region to (re)process**: Multiple independent regions can be processed. Here, it will give you the choice of all regions added to the process. * **Spectral limits (max)**: Maximum of the spectral window (default is the maximum of the ppm scale) * **Spectral limits (min)**: Minimum of the spectral window (default is the min of the ppm scale)

**Note:** **reference spectrum**: The signal that you to analyze needs to be seen in the reference spectrum.

**Note:** **spectral limits**: The difference betwwen the max and min should be at least 0.25 ppm.

You can adjust the **Peak picking threshold** to detect the desired peaks on the displayed spectrum.

While adjusting this threshold the software will automatically display a dataframe **Peak list** with the detected peaks in the region (marked with a yellow triangle on the spectrum). The peaks are displayed in the ascending order (e.g. from right to left on the spectrum).

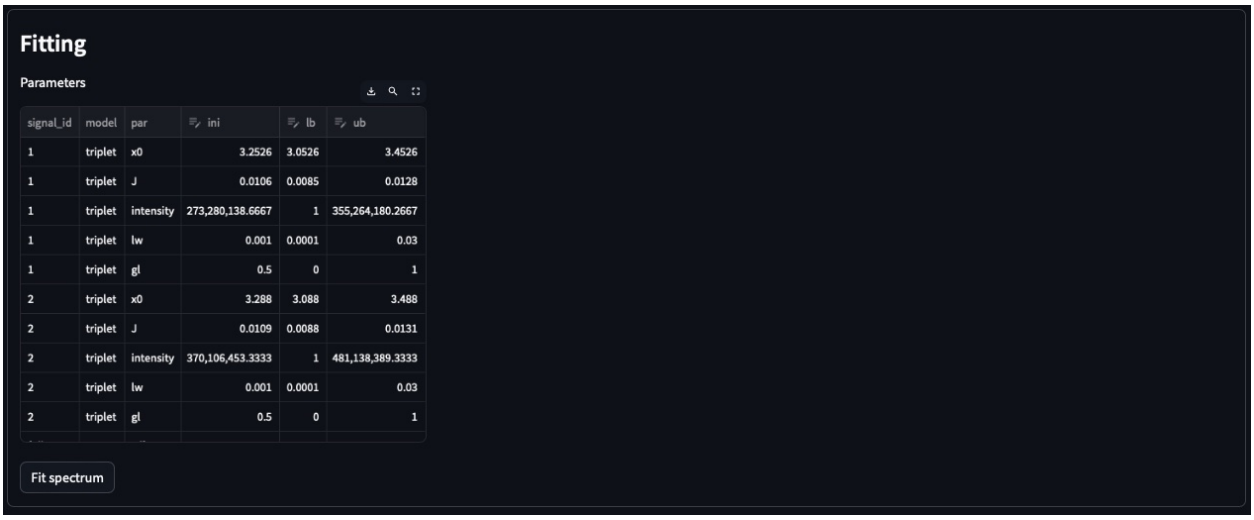
You can now proceed with the clustering steps that consists in filling out the **cluster ID** column of the **Peak list** to

group peaks together. Peaks that belongs to the same multiplets must have the same names.

**Note:** **cluster ID**: Cluter IDs can be anything (numbers or string).

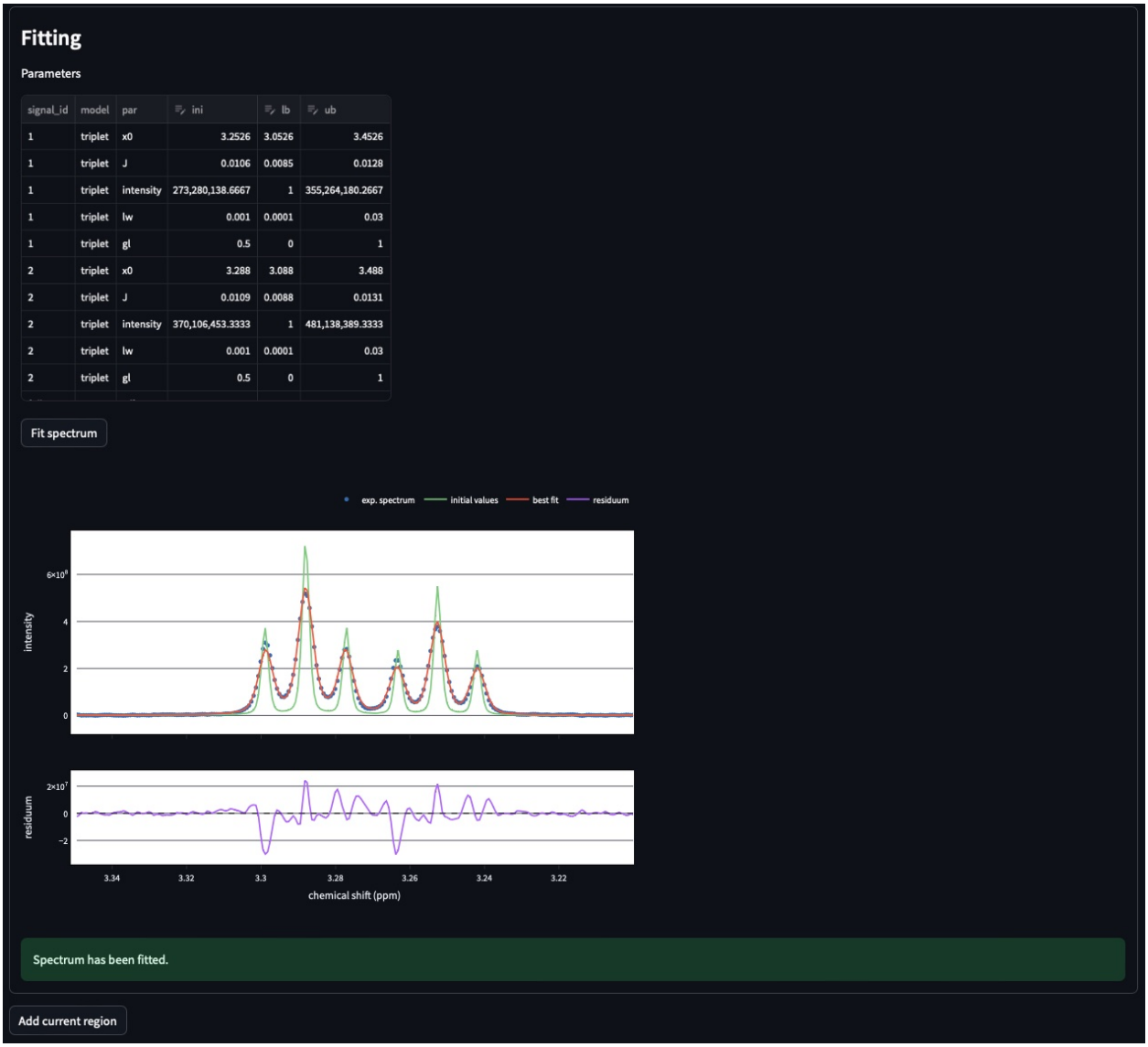
Once this clustering is performed press the **Assign peaks** button to move towards the model construction:


For each cluster MultiNMRFit will provide a choice of all the models containing this number of peaks and will give you the choice to add a offset to fit. This offset is equivalent to a linear phase correction on the selected window. Once this step is done, you can click on the **Build model** button that will automatically creates the fitting model and initially display the table of fitting parameters (at this step initial values along with boundaries).




Intitial values are calculated based on [i] the results of the peak picking (intensities and peak position) [ii] the default parameters of the each model (look at models.rst for more details on the default parameters). If no changes are required press the **Fit spectrum** button to proceed with the minimization of the reference spectrum.

**Note:** **Parameters**: All parameters are shwon in **ppm** units.




The fitted reference spectrum will be automatically displayed on the resulting graph. This plots will show [i] the experimental data as dots [ii] the best fit as red a curve and [iii] the initial values used in the minimzation in green. This is supplemented with the residum plot below.

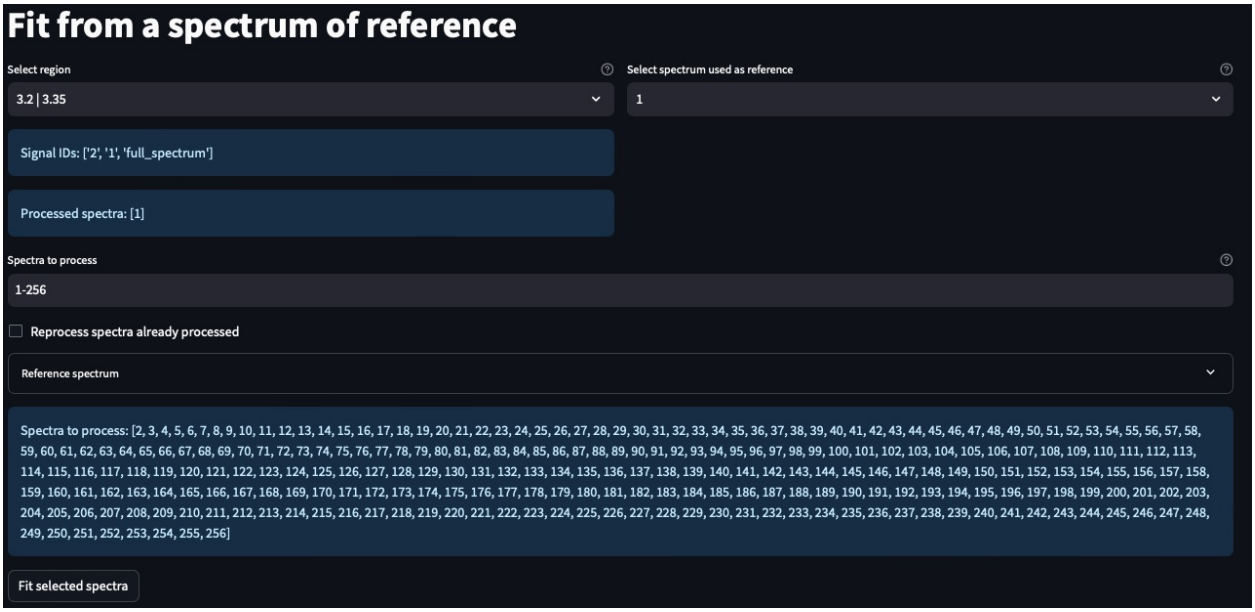
---

**Note:** **Parameters**: In the case of evident mismatch between the data and the best fit, you can adjust manually adjust the initial values in the former **parameters** table ()

---

If the results are satisying press the the **Add current region** button to save this region and eventually to the same workflow for another region of the spectra. For this you will need to go back to the top of page and select **add new region** in the field **Select region to (re)process**. Otherwise move to next page **Fit from reference**.

This page contains the wrapper that allows you to fit the desired data.




First select the region that needs to be fitted (**Select region**). Automatically MultiNMRFit will display the list of **Signal IDs** present in the selected region along with the **processed spectra** already analyzed (e.g in the first run this nunmber will correspond to the number of the reference spectrum)

MultiNMRFit will give the choice of the spectra you want to process, By default it shows the complete dataset (here 1-256 as the pseudo2D contains 256 in the example). However if you want to analyze the first ten spectra one can write 1-10 and it will update the list **spectra to process** automatically. Click the **Fit selected spectra** to run the fitting of the selected spectra. The progress of the fitting will be displayed by a progress bar and once complete a message **All spectra have been fitted** will appear.

---

**Note: Fitting**: This procedure can be repeated for the different regions defined in the previous pages upon selection in **Select region**. By default MultiNMRFit do not reprocess spectra that have been already been fitted so clicked the option if necessary. The reference spectrum associated with the slected region can be visualized on this page.

---

Once you have fitted all the data you can move to last page

This page provides several visualization options of the results. On top, you can inspect every fitted spectrum. If multiple signals were fitted on the the same region, you can observe individual ones by clicking on the different signal IDs in the figure caption.

### Spectra visualisation

Users can select the spectrum and the region to display.


### Parameters visualisation

For the corresponding spectra shown above users can find the table of paramters. A particular attention must me given to the **opt** that contains the optimal values of the fitting routine. If one value is highlited in red it means that is value is stuck to either the lower or higher bound. If this the case the spectra need to re-analyzed in leaving more freedom tho the parameter.

**Parameters**

| signal_id | model | par | ini | lb | ub | opt | opt_sd | integral |
|---|---|---|---|---|---|---|---|---|
| 1 | triplet | x0 | 3.252585 | 3.052585 | 3.452585 | 3.252662 | 0.000021 | 4704194.511116 |
| 1 | triplet | J | 0.010629 | 0.008504 | 0.012755 | 0.010786 | 0.000040 | 4704194.511116 |
| 1 | triplet | intensity | 273280138.666667 | 1.000000 | 355264180.266667 | 195309442.400065 | 1061182.371098 | 4704194.511116 |
| 1 | triplet | lw | 0.001000 | 0.000100 | 0.030000 | 0.002171 | 0.000011 | 4704194.511116 |
| 1 | triplet | gl | 0.500000 | 0.000000 | 1.000000 | 0.421393 | 0.038608 | 4704194.511116 |
| 2 | triplet | x0 | 3.288016 | 3.088016 | 3.488016 | 3.288015 | 0.000020 | 6480995.999202 |
| 2 | triplet | J | 0.010942 | 0.008754 | 0.013130 | 0.010727 | 0.000052 | 6480995.999202 |
| 2 | triplet | intensity | 370106453.333333 | 1.000000 | 481138389.333333 | 267387616.487518 | 2280498.653494 | 6480995.999202 |
| 2 | triplet | lw | 0.001000 | 0.000100 | 0.030000 | 0.002130 | 0.000010 | 6480995.999202 |
| 2 | triplet | gl | 0.500000 | 0.000000 | 1.000000 | 0.531790 | 0.016501 | 6480995.999202 |

Finally, users can observe the variation of a given paramters as function of spectra IDs.


### Export results

Users can export their results tabulated text file in 2 different manners: **all data** or **specific data** In the first case (**all data**) all the parameters of all the regions and spectra will be saved in the **output** location defined in the first page of the interface. If the second case (option **specific data** selected), you can select one region, one parameter that will exclusively saved in the file.

### Warning and error messages

Error messages are explicit. You should examine carefully any warning/error message. After correcting the problem, you might have to restart MultiNMRFit (to reload files) and perform the analysis again.

## 1.3 Models

The models used in MultiNMRFit can be found in the models folder. All models follow the same format defined in:

Users can add additionnal custom models following this format. We'll provide detailed information on the construction of new models soon! In the meantime, do not hesitate to grab us a message or open an issue in our GitHub repository.

## 1.4 How to cite

Thank you for using MultiNMRFit and citing us in your work! It means a lot to us and encourage us to continue its development.

> Manuscript in preparation

## 1.5 License

MultiNMRFit is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MultiNMRFit is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with MultiNMRFit. If not, see https://www.gnu.org/licenses/.