SNSS Spectroscopy Module: Tutorial 1 – Exercises

Using FID-A to Process Our Data

By: Peter Truong

FID-A: Basics

FID-A is a spectroscopy processing toolkit written in MATLAB. It contains functions to load in single voxel spectroscopy (SVS) and spectroscopic imaging (MRSI) data from various vendors, such as: Siemens, GE, Philips, and Bruker.

For this tutorial we will be using FID-A from GitHub (CIC-methods/FID-A: Toolbox for simulation and processing of in-vivo magnetic resonance spectroscopy (MRS) data).

Note: There will an update to FID-A to allow it to run in Octave. For now we recommend using MATLAB if you have access.

When you have MATLAB open, add the FID-A folder (along with subfolders) to its path. Navigate to the directory of your FID-A folder.

Some FID-A Tools

FID-A is categorized by a few different set of tools:

- **inputOutput**: Tools to load in and write out data.
- **processingTools**: Tools to process your MRS data that has been loaded in.
- **rfPulseTools**: Tools to create your own RF pulse waveforms.
- **simulationTools**: Tools to simulate *in vivo* MRS experiments.

In addition, there is a folder "**Example Run Scripts**" which provide examples of useful "pipelines" for NMR simulation and data processing.

Since we will be working on processing our raw MRS data, we'll mainly use the processing tools to accomplish this task.

Within "**processingTools**" there's a few functions which will help us out:

- op_combinecoils
- op_averaging
- op_addphase
- op_alignspec
- op_plotspec

In this tutorial, we will go over step-by-step processing of Siemens MEGA-PRESS Data

Step 1: Loading in files

In the folder “**exampleData**”, there is example data from Bruker (PRESS), GE (PRESS and MEGAPRESS), and Siemens (MEGAPRESS and SPECIAL).

Different vendors have different filetypes. Lucky for us, FID-A has load-in tools for most vendors located in the “**inputOutput**” folder:

- Siemens TWIX > **io_loadspec_twix()**
- GE Pfiles > **io_loadspec_GE()**
- Philips SPAR and SDAT > **io_loadspec_sdat()**
- Siemens RDA > **io_loadspec_rda()**
- Siemens IMA > **io_loadspec_ima()**
- Bruker > **io_loadspec_bruk()**

For our example, let’s open up Siemens MEGAPRESS data. Navigate to “**exampleData/Siemens/sample01_megapress**”. Load in TWIX data using the command:

```
raw=io_loadspec_twix(filename);
```

- **filename** is the name of the PFile. In this case it will be “**megapress/megapressDLPFC.dat**”
- If you want information regarding what a function does along with the inputs needed and outputs, simply type in:
    - **help <function name>**
    - **e.g. help io_loadspec_twix**

We also want to load in the water unsuppressed data as well:

```
raw_w=io_loadspec_twix(megapress_w/megapressDLPFC_w.dat);
```

Step 2: Looking at the FID-A Data Structure

You can view the data structure simply by typing in the structure name in the command window.

Have a look at the field “**dims**” and compare it with the size of our data arrays “**fids**” and “**specs**” [2080,32,80,2]. The dimensions indicate:

- 1st dimension = “time” dimension (2080 points)
- 2nd dimension = “coils” dimension (32 channels)
- 3rd dimension = “averages” dimension (80 averages)
- 4th dimension = “subSpecs” dimension (2 subspectra)

As well, you’ll see some important header information, such as: TE, TR, spectral width, transmit frequency (centre frequency), magnetic field strength (Bo), and dwell time.

Step 3: Combining the RF channels

For our acquisition, we used an RF coil with 32 channels. Thus, we need to combine them using **op_combineRcvrs**.

```
[out1, out1_w] = op_combineRcvrs(raw,raw_w);
```

Now look at the "**dims**" field and data size of "**raw.fids**" **and** "**raw.specs**". Note now that the coils dimension is removed.

Step 4: Plotting the data

There are two functions to plot our data:

- **op_plotfid**
- **op_plotspec**

When entering in: **op_plotfid(out1);** you will be asked whether you would like to view "**all averages**" or "**subspectra**".

With "**all averages**" the plot will display only the 1st subspectrum.

With "**all subspectra**" the plot will display only the 1st average.

As you can see from the outputs, "**op_plotfid**" will output display the FIDs. Try again this time using "**op_plotspec**" to display the spectra.

Step 5: Aligning the averages using Spectral Registration and Combining

Before we combine the averages together, it is important to ensure that the averages are properly aligned. This task can be accomplished using "**op_alignAverages**"

```
out2=op_alignAverages(out1);
```

Plot aligned spectra, using the option to see "**all averages**". Compare it to the spectra of **out1**.

Now that we confirmed the averages are properly aligned, you can combine the averages together using:

```
out3=op_averaging(out2);
```

Step 6: Align Subspectra and Subtract

Since our data is from MEGA-PRESS acquisition, we must also align the subspectra to one another using "**op_alignMPSubspecs**". This is different from the previous "**op_alignAverages**" function, because it

```
out4=op_alignMPSubspecs(out3);
```

To then subtract the subspectra:

```
out5=op_combinesubspecs(out4,'diff');
```

Step 7: Final Phase Correction and Filtering

The processing steps can be nested from one another. The nested step will be run first.

The last two steps we need to perform is our final phase correction ("**op_autophase**") to make sure all the peaks are upright, followed by filtering ("**op_filter**") of our data to remove some excess noise.

Nesting the two commands:

```
out6 = op_filter(op_autophase(out5,1.8,2.2,180),2);
```

This will run "**op_autophase**" first, and phase the spectrum with respect to the NAA peak between 1.8 and 2.2 ppm. As well, it will apply an additional 180 degree phase correction, so that our difference spectrum has the edited-GABA peak upright and the edited-NAA peak down.

Then "**op_filter**" will be run on the output from "**op_autophase**", using a 2 Hz exponential decay to line broaden the spectrum.

Step 8: Writing Out the Output for LCModel

Finally we can output our final processed spectrum to a format for LCModel analysis.

```
io_writelcm(out5,'megapress_lcm',68);
```

This will give us an output file "**megapress_lcm**" in our current directory. The "**68**" is the TE, which needs to be specified in the LCModel file. We're using **out5** since LCModel prefers data without any line broadening applied.

Trying out Example Run Scripts

Let's try a complete run script pipeline! These example run scripts will have all the loading and processing steps needed to process your data. You can use them as a reference and modify them for your own data.

run_pressproc_GEauto()

run_megapressproc_auto()

run_specialproc_auto()

Use "**help <function>**" to determine what outputs the scripts generate and the inputs the script needs.

With these scripts you will get a "**report.html**" file which will show you the results from the processing steps run, such as combining RF coils, removal of bad averages, spectral registration, and final output spectra (and subspectra).