

MRS simulation and preprocessing using the FID-A toolkit

MRS Hackathon 2023

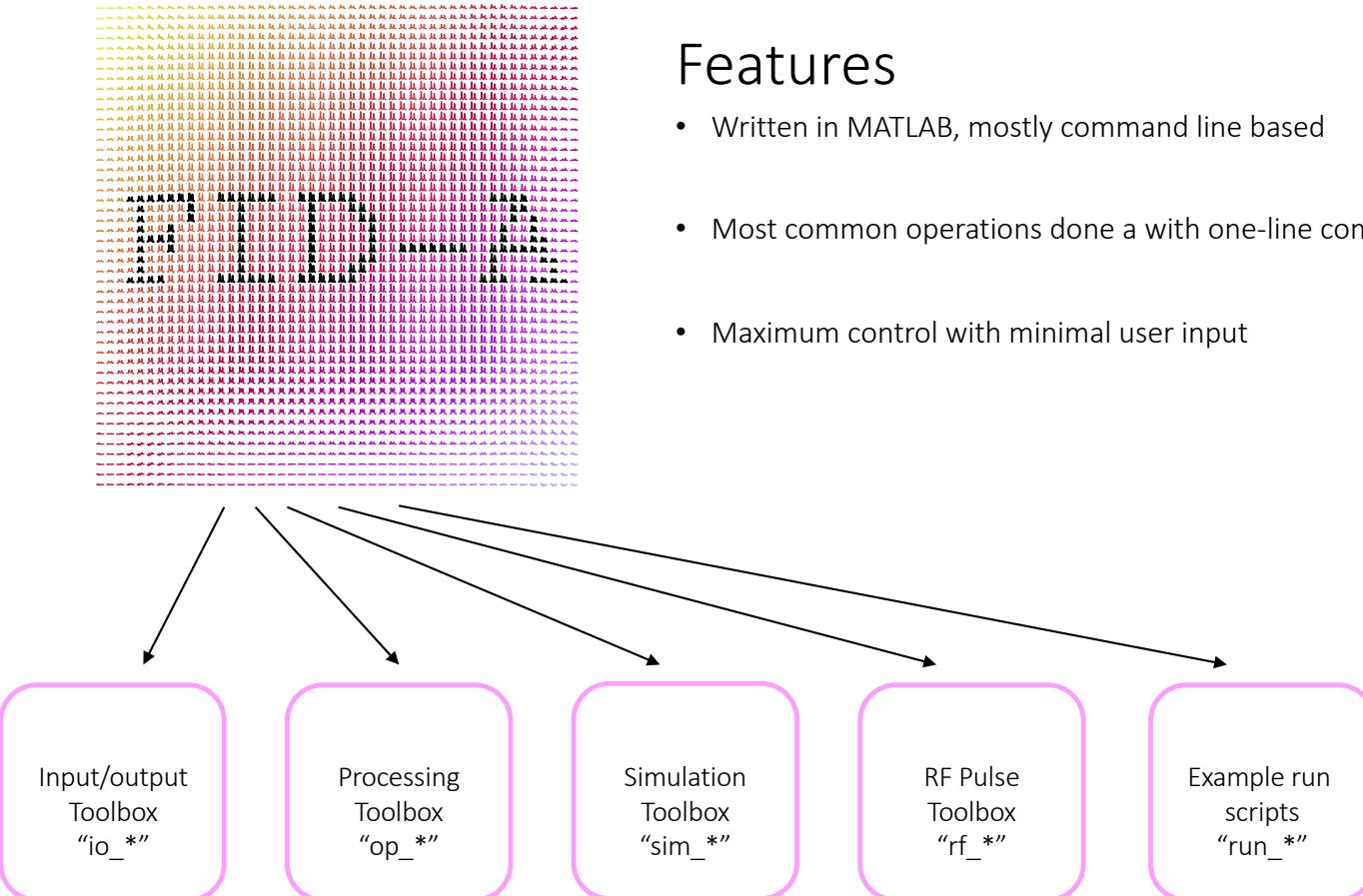
Jamie Near

Outline

- Brief description of the FID-A toolkit
 - What FID-A does
 - What FID-A doesn't do
- Practical examples:
 - Downloading/installing the software
 - Loading and manually processing an MRS dataset
 - Automated processing pipelines
 - Data quality assurance
 - Loading and analyzing an RF pulse waveform
 - Simulating MRS experiments

What is FID-A?

- The FID appliance (FID-A) is an open-source, MATLAB-based software toolkit for advanced MRS simulation and data processing, as well as basic RF pulse design and analysis.
- A framework to access, manage, and visualize all aspects of your MRS data
- This toolkit is freely available from www.github.com/CIC-methods/FID-A.



Features

- Written in MATLAB, mostly command line based
- Most common operations done with one-line command
- Maximum control with minimal user input

Features (Cont'd):

- Designed for raw datasets with multiple dimensions
 - RF coil channels
 - Averages
 - Subtraction sub-spectra, etc.
- Large library of processing tools, simulation tools
- Data structure format contains all relevant header information
- Contains example "run scripts" with pre-implemented:
 - Processing pipelines
 - Simulation of common MRS sequences (both ideal and shaped-pulse versions)
 - Basis set generation
- Users with MATLAB experience can easily modify code to suit their needs.

Full list of FID-A input/output tools:

Siemens

- `io_loadRFwaveform.m`
- `io_loadspec_IMA.m`
- `io_loadspec_twix.m`
- `io_loadspec_rda.m`
- `io_readpta.m`
- `io_writepta.m`

Bruker

- `io_loadRFwaveform.m`
- `io_loadspec_bruk.m`
- `io_loadspec_brukNMR.m`
- `io_loadspec_jrBruk.m`
- `io_readRFBruk.m`

GE

- `io_loadspec_GE.m`

Philips

- `io_loadspec_data.m`
- `io_loadspec_sdat.m`

Varian/Agilent

- `io_loadRFwaveform.m`
- `io_loadspec_varian.m`
- `io_readRF.m`
- `io_writeRF.m`

jMRUI

- `io_loadjmriui.m`
- `io_readjmriui.m`
- `io_writejmriui.m`

LCModel

- `io_loadlcmdetail.m`
- `io_readlcmcord_getBackground.m`
- `io_readlcmcord.m`
- `io_readlcraw_basis.m`
- `io_readlcraw_dotraw.m`
- `io_readlcraw.m`
- `io_readlcmtab.m`
- `io_writelcm.m`
- `io_writelcraw.m`

VeSPA

- `io_readRFtxt.m`
- `io_writeRFtxt.m`

Full list of FID-A processing operations:

Dimensionality manipulation:

- op_ISIScombine.m
- op_addrcvrs.m
- op_averaging.m
- op_combineRcvrs.m
- op_combinesubspecs.m
- op_concatAverages.m
- op_concatFreq.m
- op_concatSubspecs.m
- op_downsamp.m
- op_median.m
- op_takeaverages.m
- op_takesubspec.m
- op_takeextras.m

Phase/Frequency shifting:

- op_addphase.m
- op_addphaseSubspec.m
- op_autophase.m
- op_freqshift.m
- op_freqshiftSubspec.m
- op_makeFreqDrift.m
- op_makePhaseDrift.m
- op_movef0.m
- op_ppmref.m

Filtering:

- op_filter.m
- op_matchLW.m
- op_unfilter.m

Maths operations:

- op_addScans.m
- op_ampScale.m
- op_complexConj.m
- op_integrate.m
- op_subtractScans.m

Truncation:

- op_fqrange.m
- op_leftshift.m
- op_timerange.m
- op_zeropad.m
- op_zerotrim.m

Basic peak fitting:

- op_crefit.m
- op_gaussianPeak.m
- op_integrate.m
- op_lorentz.m
- op_lorentz_linbas.m
- op_lorentzianPeak.m
- op_peakFit.m

Quality assurance:

- op_arsos.m
- op_getLW.m
- op_getSNR.m
- op_relyTest.m
- op_rmbadaverages.m
- op_rmNworstaverages.m
- op_rmworstaverage.m

Spectral Registration:

- op_alignAllScans.m
- op_alignAllScans_fd.m
- op_alignAverages.m
- op_alignAverages_fd.m
- op_alignISIS.m
- op_alignMPSubspecs.m
- op_alignMPSubspecs_fd.m
- op_alignScans.m
- op_alignScans_fd.m
- op_alignrcvrs.m
- op_freqAlignAverages.m
- op_freqAlignAverages_fd.m
- op_getcoilcombos.m
- op_getcoilcombos_specReg.m
- op_phaseAlignAverages.m
- op_phaseAlignAverages_fd.m

Plotting:

- op_plotfid.m
- op_plotspec.m

Other:

- op_addNoise.m
- op_dccorr.m
- op_ecc.m
- op_fddccorr.m
- op_getPeakHeight.m
- op_getcoilcombos.m
- op_getcoilcombos_specReg.m
- op_removeWater.m

Full list of FID-A simulation tools:

Basic Spin Operators

- sim_Hamiltonian.m
- sim_evolve.m
- sim_excite.m
- sim_excite_arbPh.m
- sim_gradSpoil.m
- sim_readout.m
- sim_rotate.m
- sim_rotate_arbPh.m
- sim_shapedRF.m
- sim_spoil.m
- sim_dAdd.m
- sim_dDiv.m
- sim_dMul.m

Pulse Sequences (Ideal)

- sim_laser.m
- sim_lcmrawbasis.m
- sim_megapress.m
- sim_onepulse.m
- sim_onepulse_arbPh.m
- sim_onepulse_delay.m
- sim_press.m
- sim_spinecho.m
- sim_spinecho_xN.m
- sim_steam.m
- sim_steam_gradSim.m

Pulse Sequences (shaped)

- sim_make2DSimPlot.m
- sim_megapress_shaped.m
- sim_megapress_shapedEdit.m
- sim_megapress_shapedRefoc.m
- sim_megaspecial_shaped.m
- sim_onepulse_shaped.m
- sim_press_shaped.m
- sim_sLASER_shaped.m
- sim_spinecho_shaped.m

Full list of FID-A RF pulse tools:

Bloch Simulation

- rf_blochSim.m
- rf_refocusedComponent.m

RF Pulse Design

- rf_addGrad.m
- rf_dualBand.m
- rf_freqshift.m
- rf_gauss.m
- rf_goia.m
- rf_hs.m
- rf_plotWaveform.m
- rf_resample.m
- rf_scaleGrad.m
- rf_sinc.m
- rf_verse.m

Full list of FID-A example run scripts:

Processing

- run_megapressproc_auto.m
- run_megapressproc_GEauto.m
- run_megapressproc.m
- run_pressproc_auto.m
- run_pressproc_brukAuto.m
- run_pressproc_GEauto.m
- run_pressproc.m
- run_specialproc_auto.m
- run_specialproc_fmrsl_slidingWindow.m
- run_specialproc_fmrsl.m
- run_specialproc.m

Simulation

- run_make2DSimPlot.m
- run_simExampleBasisSet.m
- run_simMegaExTEShaped.m
- run_simMegaPressShaped.m
- run_simMegaPressShapedEdit.m
- run_simMegaPressShapedRefoc.m
- run_simMegaPressShapedRefoc_fast.m
- run_simMegaPressShaped_fast.m
- run_simMegaSpecialShaped.m
- run_simPressShaped.m
- run_simPressShaped_fast.m
- run_simsLaserShaped.m
- run_simsLaserShaped_fast.m
- run_simSpinEchoShaped.m

Quality Assurance

- run_getLWandSNR.m

What doesn't FID-A do:

- Does not have its own graphical user interface.
- Not a standalone piece of software
 - Requires a MATLAB license
 - Some functions require certain MATLAB toolboxes:
 - Statistics and Machine Learning Toolbox (A.K.A. Statistics Toolbox in older MATLAB Versions)
 - Parallel Computing Toolbox

In progress:

- Support for MRSI / CSI data
- Support for X-nuclei

Practical examples:

- Downloading/installing the software
- Loading and manually processing an MRS dataset
- Automated processing pipelines
- Data quality assurance
- Loading and analyzing an RF pulse waveform
- Simulating MRS experiments

Downloading and Installing:

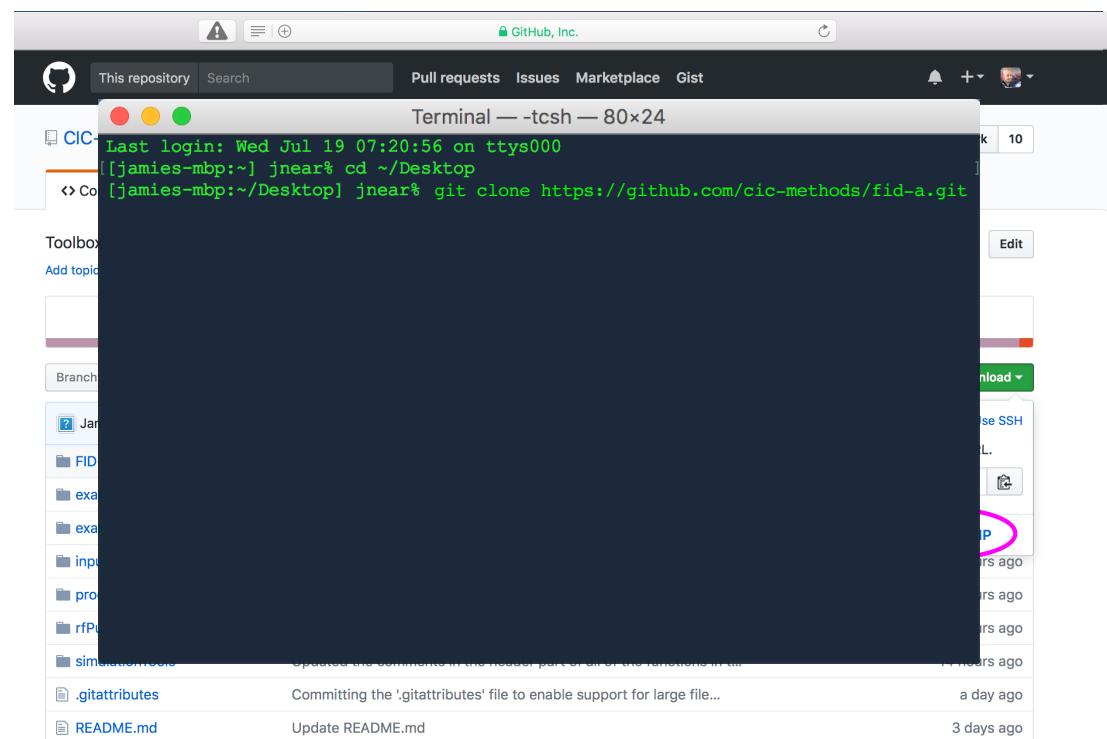
Basic Installation:

1. Go to github.com/cic-methods/fid-a
2. Click on the “Clone or Download” button
3. Click on “Download ZIP”
4. Unzip the ‘FID-A-master.zip’ file and move the resulting folder into the desired location.

Advanced Installation (Requires git and git-lfs to be installed):

1. Open linux terminal
2. Navigate to the desired directory
3. Type:

```
git clone https://github.com/cic-methods/fid-a.git
```



A screenshot of a GitHub repository page for 'fid-a'. A terminal window is open in the sidebar, showing the command 'git clone https://github.com/cic-methods/fid-a.git' being run. The terminal output shows the user's last login information and the command being executed. The GitHub interface includes a sidebar with branches like 'JAR', 'FID', 'example', etc., and a main area showing commit history for files like '.gitattributes' and 'README.md'.

File	Commit Message	Date
.gitattributes	Committing the '.gitattributes' file to enable support for large file...	a day ago
README.md	Update README.md	3 days ago

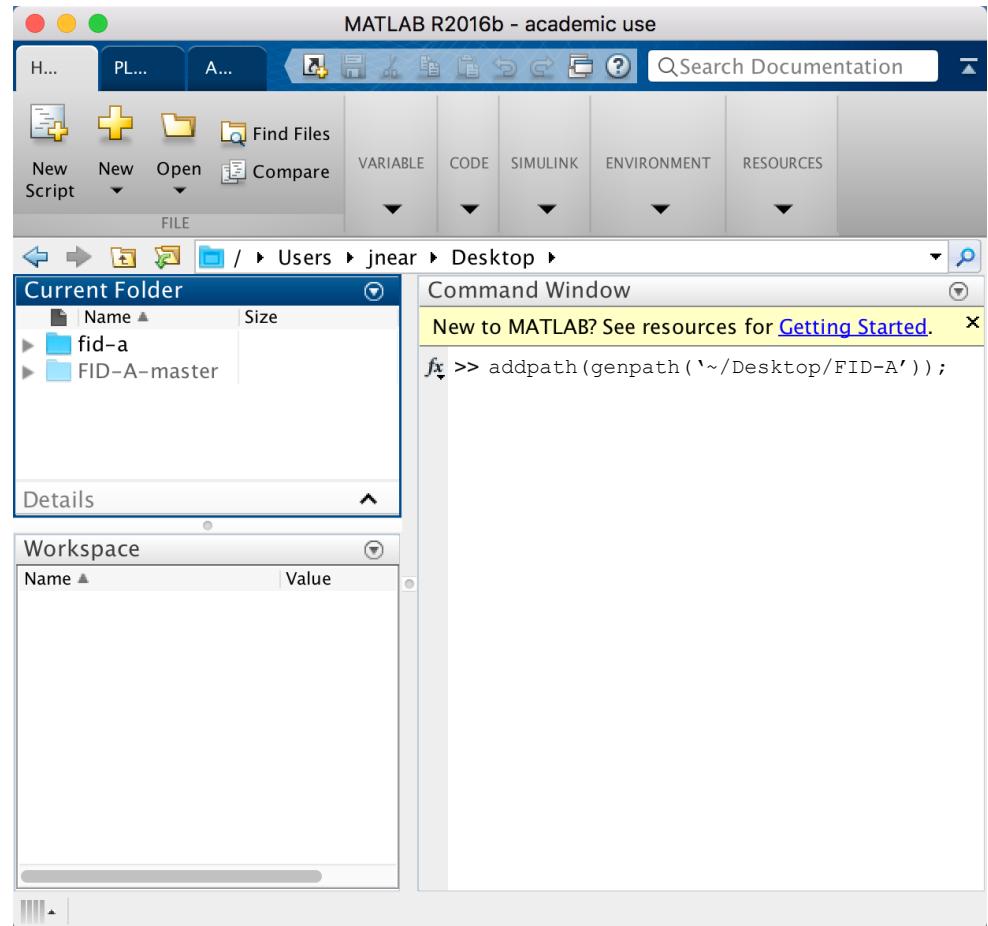
Downloading and Installing (cont'd):

Next:

1. Open MATLAB
2. Add the FID-A directory and all its subdirectories to path using the path tool, or by typing:

```
addpath(genpath('~/Desktop/FID-A'));
```

3. DONE!!



Practical examples:

- Downloading/installing the software
- Loading and manually processing an MRS dataset
- Automated processing pipelines
- Data quality assurance
- Loading and analyzing an RF pulse waveform
- Simulating MRS experiments

Load example data

(Siemens MEGA-PRESS data)

1. Go to 'exampleData' directory containing Siemens raw twix (.dat) dataset:

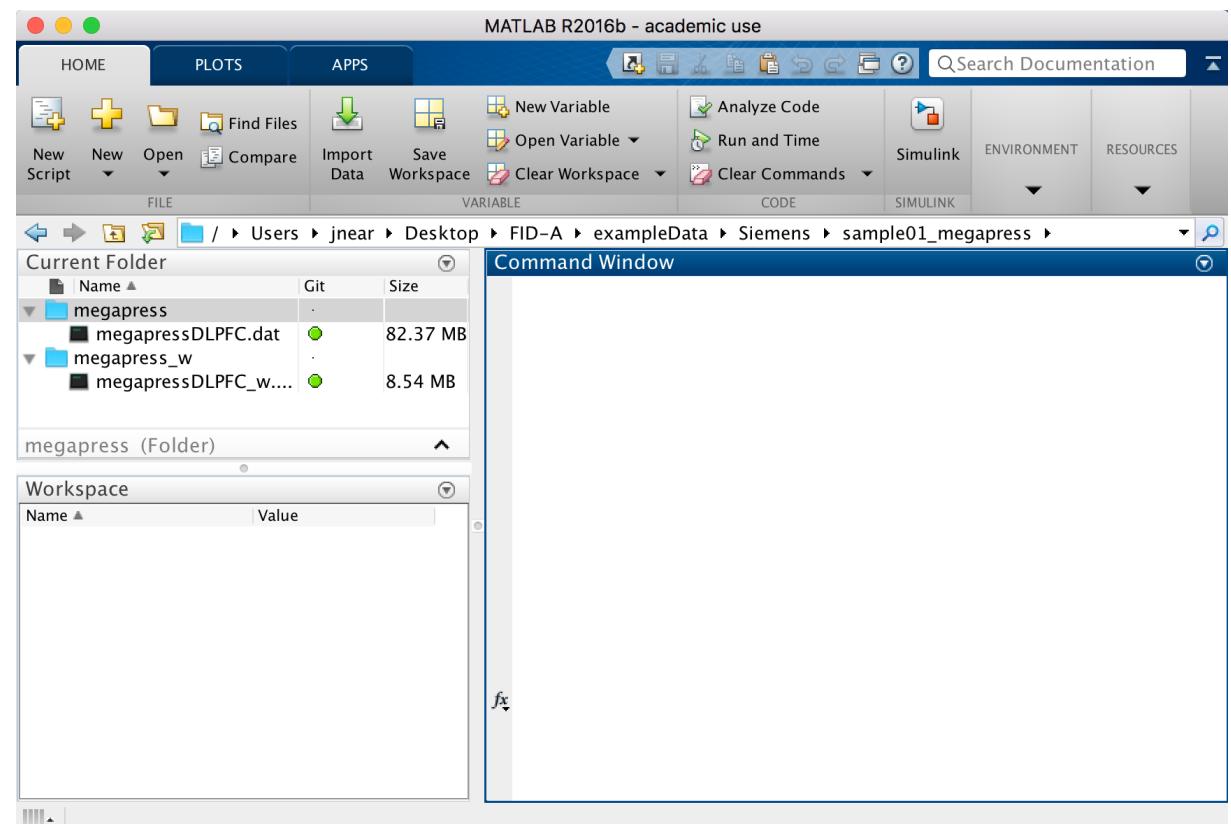
```
cd FID-A/exampleData/Siemens/sample01_megapress
```

2. Load water suppressed data using io_loadspec_twix:

```
raw=io_loadspec_twix('megapress/megapressDLPFC.dat');
```

3. Load water suppressed data using io_loadspec_twix:

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



View the data structure

To view the structure, type the variable name at the command line:

```
raw
```

Data structures contain both data, and header information.

Same format is used for both experimental and simulated data.

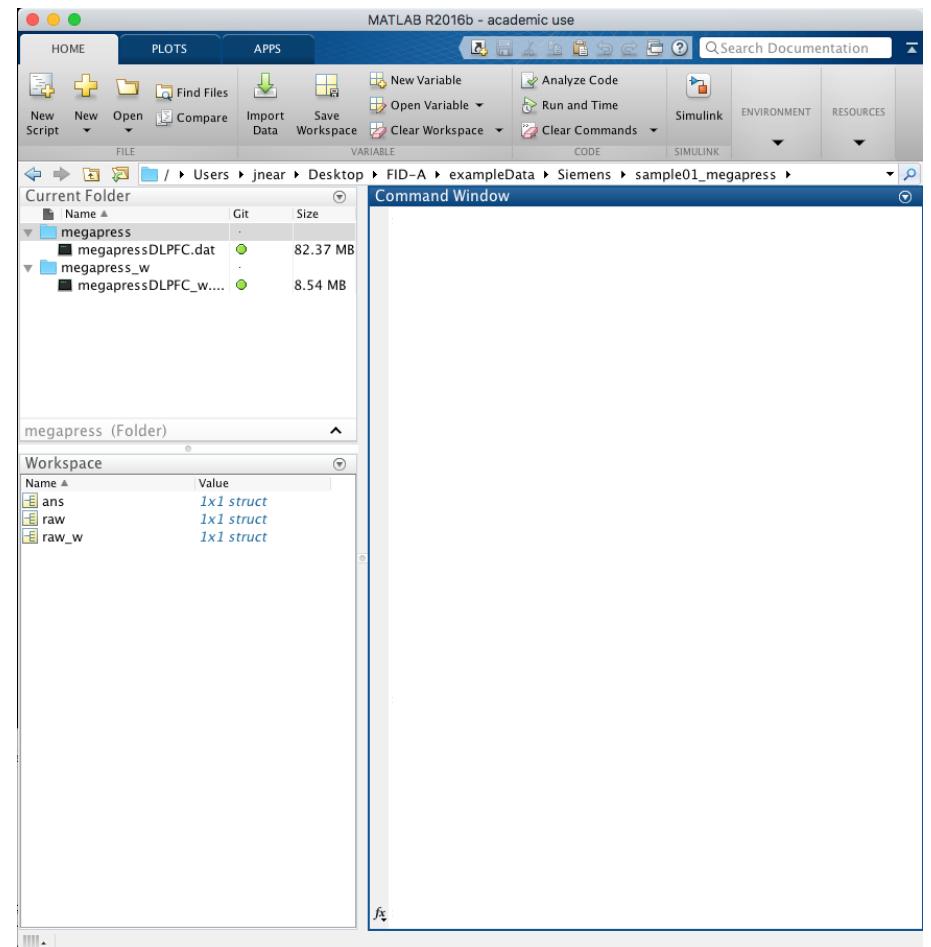
Note size of data arrays, raw.fids and raw.specs (2080 x 32 x 80 x 2). To find out what those dimensions mean, type:

```
raw.dims
```

The raw.dims structure indicates:

- 1st dimension is the "time" dimension (2080 points),
- 2nd dimension is the "coils" dimension (32 channels),
- 3rd dimension is the "averages" dimension (80 averages)
- 4th dimension is the "subSpecs" dimension (2 subspectra).

Next step: combine the rf channels.



Combining the RF channels

Combine RF channels for both the water suppressed and unsuppressed datasets using:

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

Look at the result by querying variable name:

```
out1
```

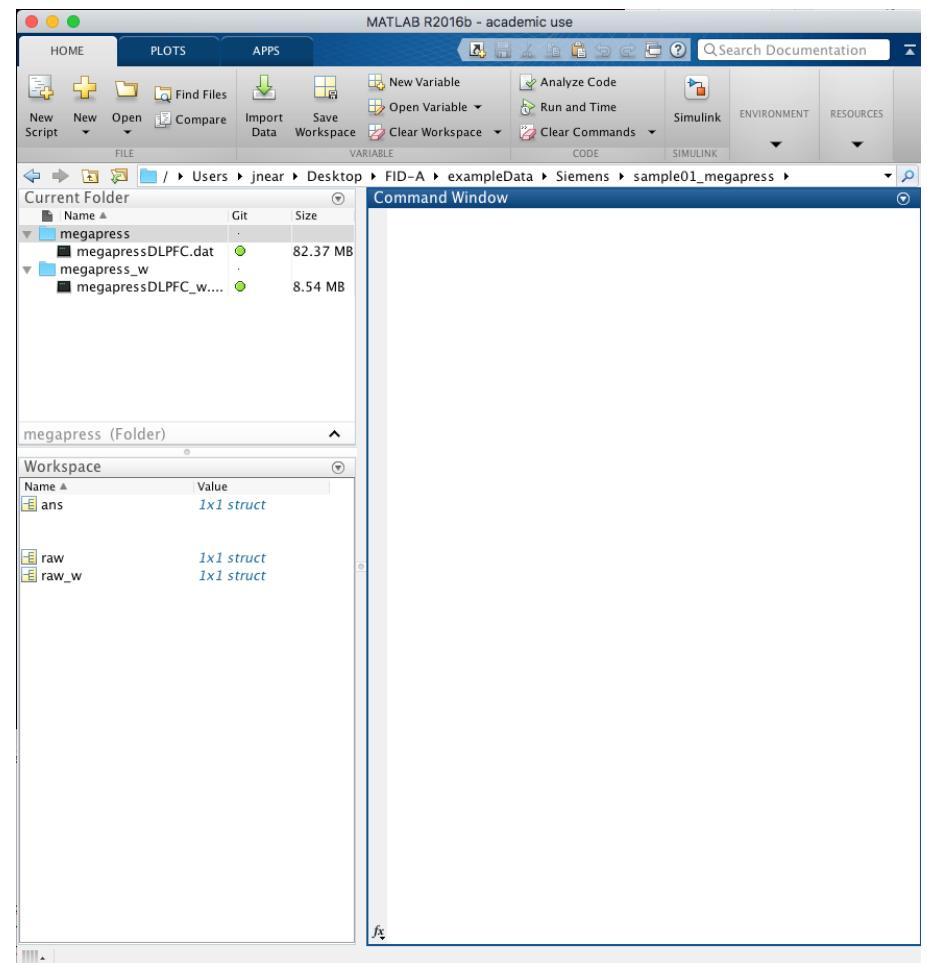
Note the dimensionality of the data arrays, raw.fids and raw.specs has been reduced to 2080 x 80 x 2. The Coils dimension has been removed:

```
out1.dims
```

The out1.dims structure now indicates:

- 1st dimension is the “time” dimension (2080 points),
- 2nd dimension is the “averages” dimension (80 averages),
- 3rd dimensions is the “subSpecs” dimension (2 subSpecs)

Now plot the spectra...



Plotting the data

Plot the time domain data using:

```
op_plotfid(out1);
```

Which dimension would we like to view?

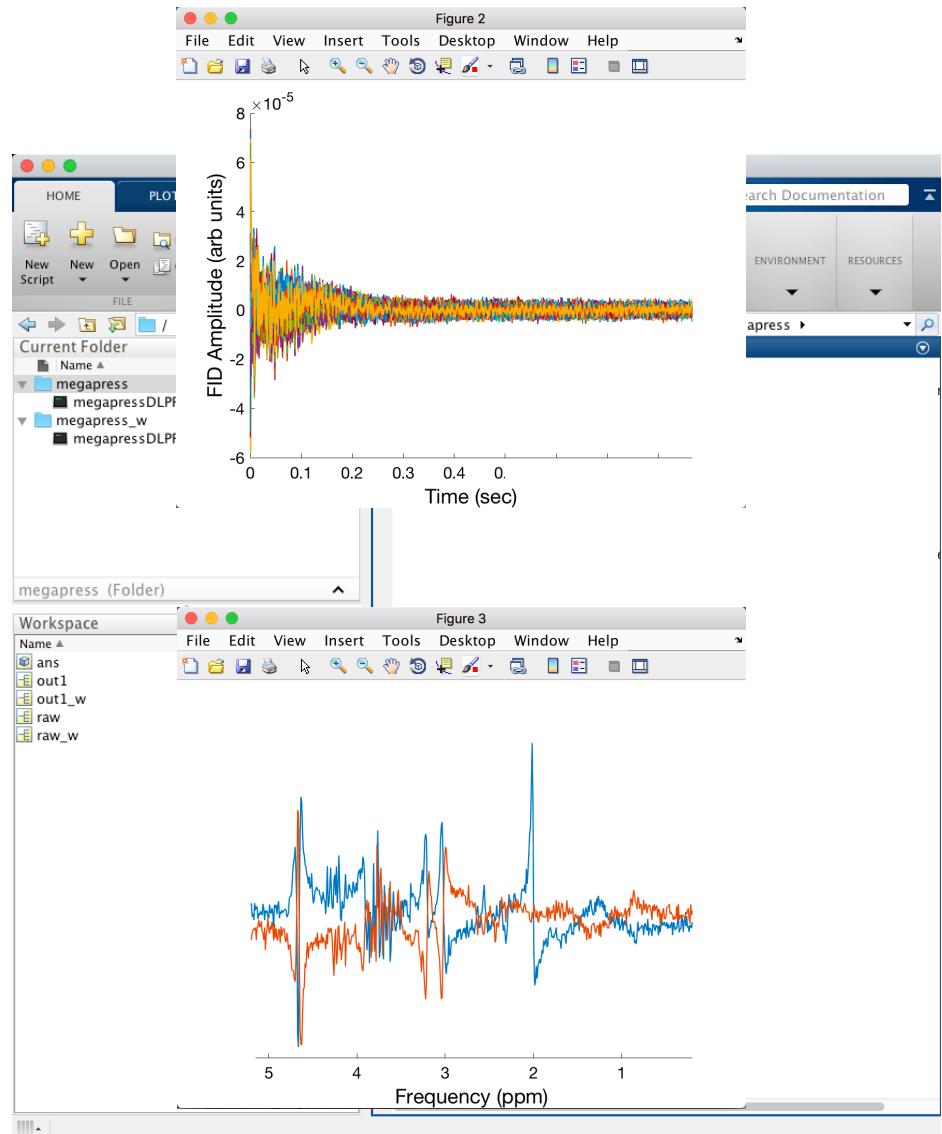
Choose '2' to view all averages (will display only the 1st subspectrum).

We can plot the frequency domain spectra using:

```
op_plotspec(out1);
```

Again, which dimension would we like to view?

This time, choose '3' to view both subspectra (will display only the 1st average).



More processing steps:

Do spectral registration (and plot result):

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

Combine averages:

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

Align MEGA-PRESS subspectra :

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

Subtract MEGA-PRESS subspectra :

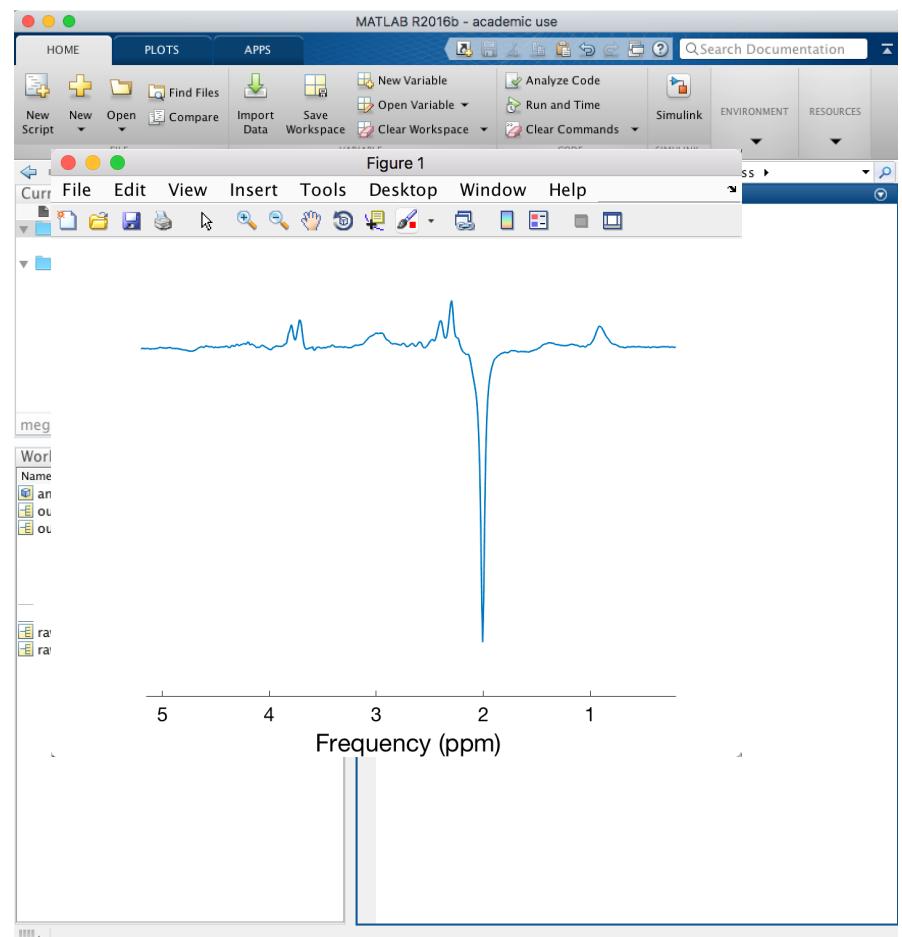
```
out5=op_combinesubspecs(out4,'diff');
```

Zero-order phase, filter (and plot result):

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

Finally, write result into LCModel text format for analysis:

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



Practical examples:

- Downloading/installing the software
- Loading and manually processing an MRS dataset
- Automated processing pipelines
- Data quality assurance
- Loading and analyzing an RF pulse waveform
- Simulating MRS experiments

Automated processing pipeline:

Processing steps can be combined into a single automated script for speed and efficiency.

Some example processing pipelines are provided in the “exampleRunScripts” directory.

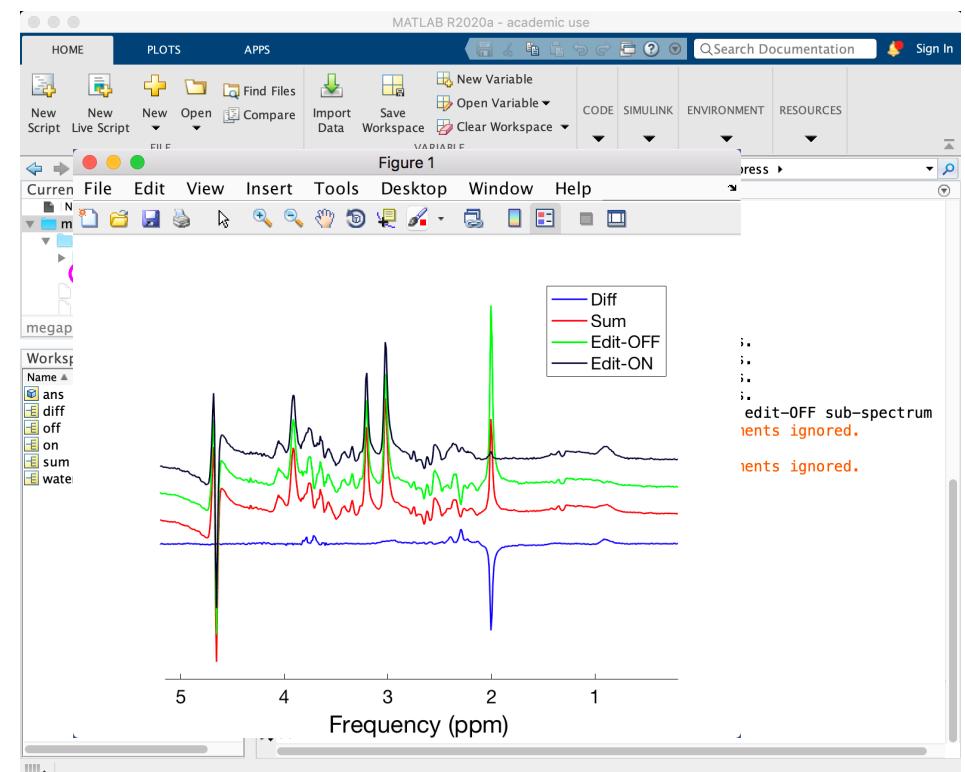
Let's use the “run_megapressproc_auto.m” script to process the same MEGA-PRESS data as before:

```
[diff,sum,off,on,water]=run_megapressproc_auto('megapress');
```

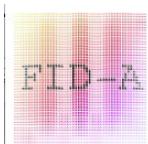
The full processing pipeline takes less than one minute. Now plot all resulting spectra on a single plot using op_plotspec.m:

```
op_plotspec({diff,sum,off,on});  
legend('Diff','Sum','Edit-OFF','Edit-ON');
```

Notice the new “report” directory that appears inside the ‘megapress’ directory. Let's look inside at the ‘report.html’ file.



report.html file



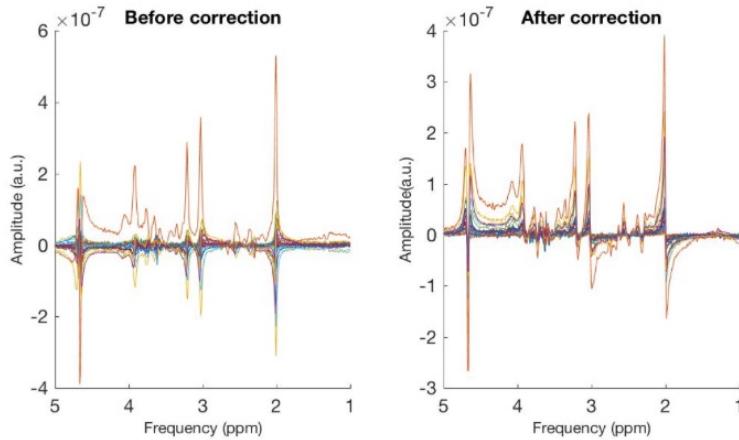
FID-A Processing Report

Processing pipeline applied to MEGA-PRESS data using `run_megapressproc.m`

FILENAME: /Users/jnear/Desktop/FID-A/exampleData/Siemens/sample01_megapress/megapress/megapressDLPFC.dat

DATE: 19-Jul-2017

Results of multi-coil combination:



report.html file

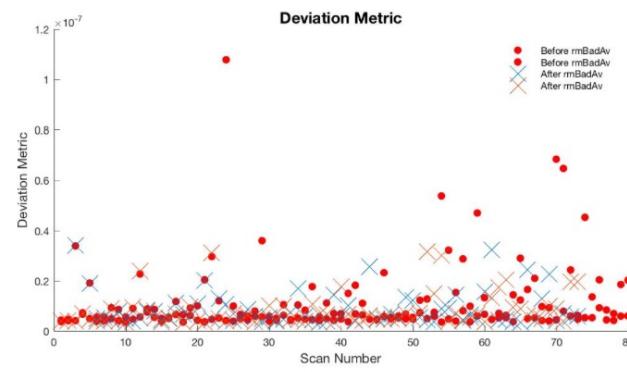
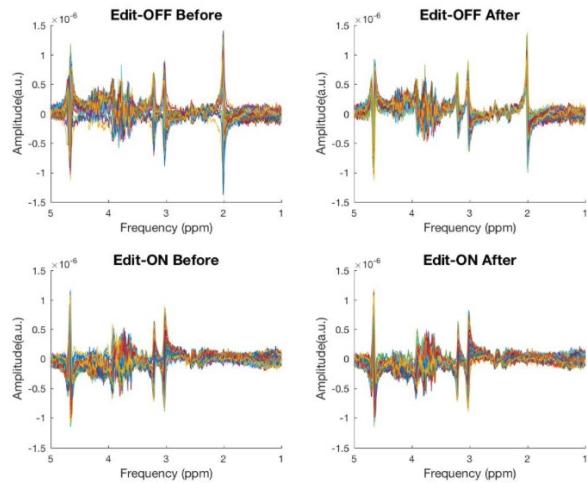
Results of removal of bad averages:

Original number of averages: 160.000000

Number of bad Averages removed: 14.000000

Number of remaining averages in processed dataset: 146.000000

Bad Averages Removal Threshold was: 4.00

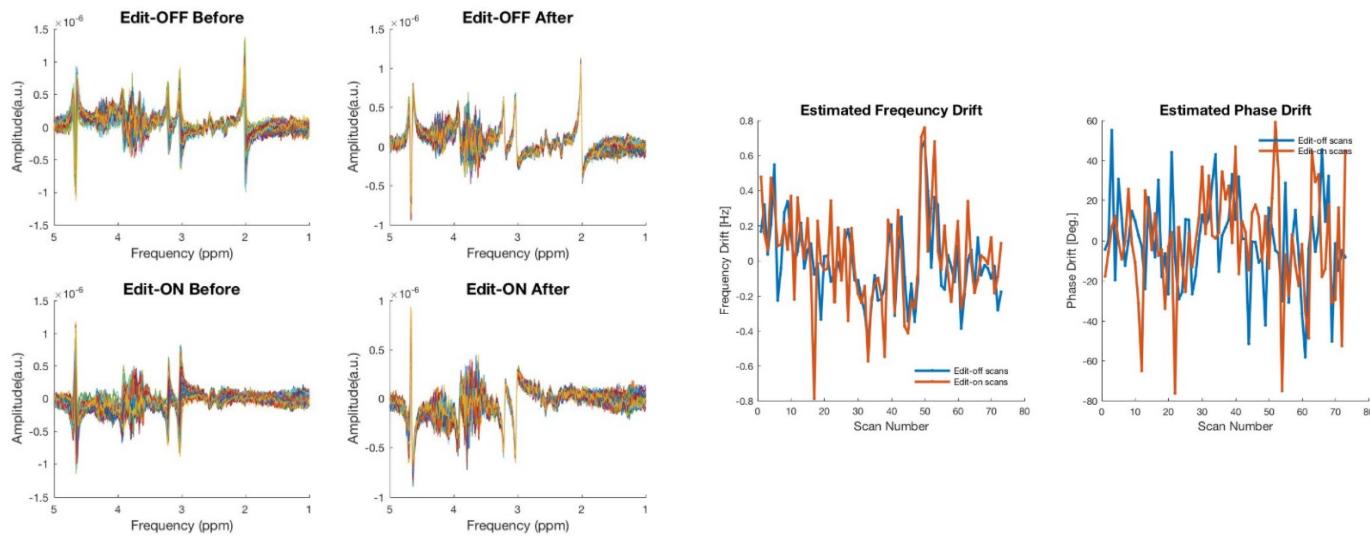


report.html file

Results of spectral registration:

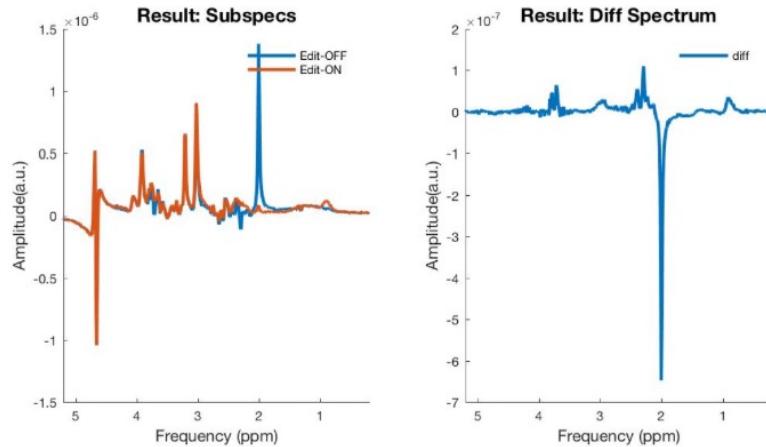
Total frequency drift was: 1.350786

Total phase drift was: 124.315027



report.html file

Final Result:



Practical examples:

- Downloading/installing the software
- Loading and manually processing an MRS dataset
- Automated processing pipelines
- Data quality assurance
- Loading and analyzing an RF pulse waveform
- Simulating MRS experiments

Data quality assurance

After processing, it is a good idea to check spectral quality.

The two most important and unbiased spectral quality metrics are:

- **Linewidth**, as measured by the full-width at half-maximum (FWHM) of a given peak
- **Signal-to-noise ratio (SNR)**, as measured by the peak height divided by the standard deviation of the noise.

Calculate the linewidth using the 'op_getLW' function. You can measure the linewidth of the water peak as follows:

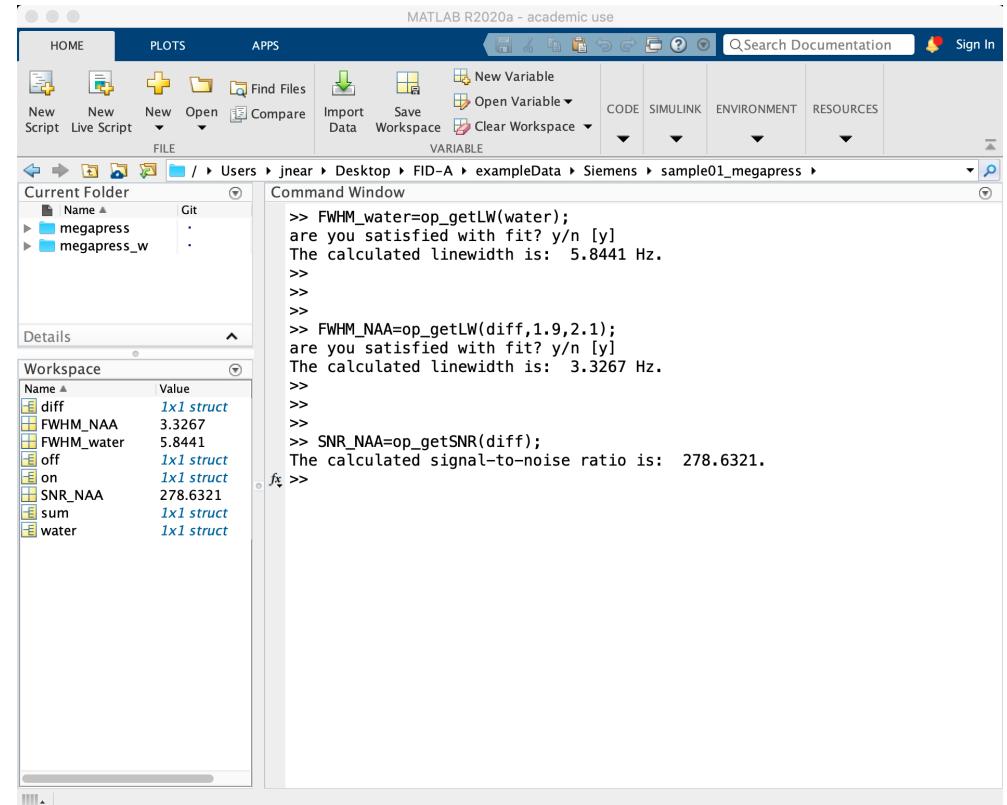
```
FWHM_water=op_getLW(water);
```

Or of a the NAA peak (for example) as follows:

```
FWHM_NAA=op_getLW(diff,1.9,2.1);
```

Lastly, calculate the SNR using 'op_getSNR' as follows:

```
SNR_NAA=op_getSNR(diff);
```



Practical examples:

- Downloading/installing the software
- Loading and manually processing an MRS dataset
- Automated processing pipelines
- Data quality assurance
- Loading and analyzing an RF pulse waveform
- Simulating MRS experiments

Loading and simulating

Some sample RF pulses are located in:

```
FID-A/rfPulseTools/rfPulses/
```

Load an RF pulse:

```
refoc=io_loadRFwaveform('sampleRefocPulse.pta','ref');
```

View the RF pulse structure by querying the RF pulse variable name:

```
refoc
```

RF pulse structure contains both the waveform, and header information.

Plot the amplitude and phase waveforms using:

```
rf_plotWaveform(refoc);
```

Do bloch simulation to determine the inversion profile (3 ms pulse duration):

```
[mv,sc]=rf_blochSim(refoc,3);
```

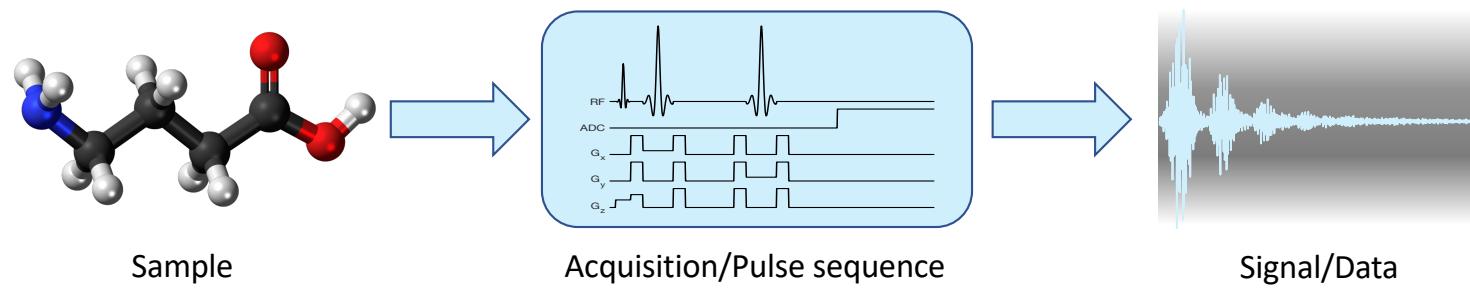


Practical examples:

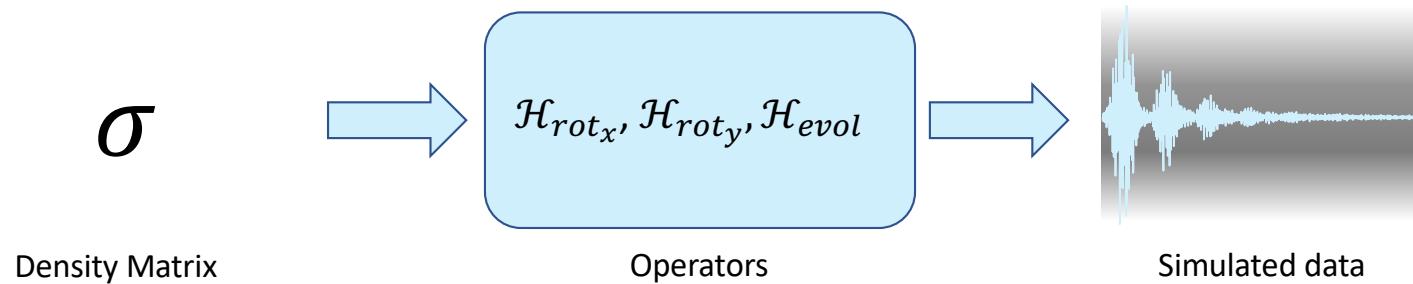
- Downloading/installing the software
- Loading and manually processing an MRS dataset
- Automated processing pipelines
- Data quality assurance
- Loading and analyzing an RF pulse waveform
- Simulating MRS experiments

Simulation Overview

Experiment:



Density Matrix Simulation:



Spin systems in FID-A

- Many spin system definitions are included:

- 2-hydroxyglutarate
- Alanine
- Ascorbate
- Aspartate
- Beta-hydroxybutyrate
- Citrate
- Creatine
- GABA
- Glucose
- Glutamine
- Glutamate
- Glycine
- Glycerophosphocholine
- Glutathione
- Lactate
- Myo-inositol
- NAA
- NAAG
- Phosphocholine
- Phosphocreatine
- Phosphorylethanolamine
- Phenylalanine
- Scyllo-inositol
- Serine
- Taurine
- Tyrosine

FID-A spin system definitions

Metabolite spin system definitions are stored in:

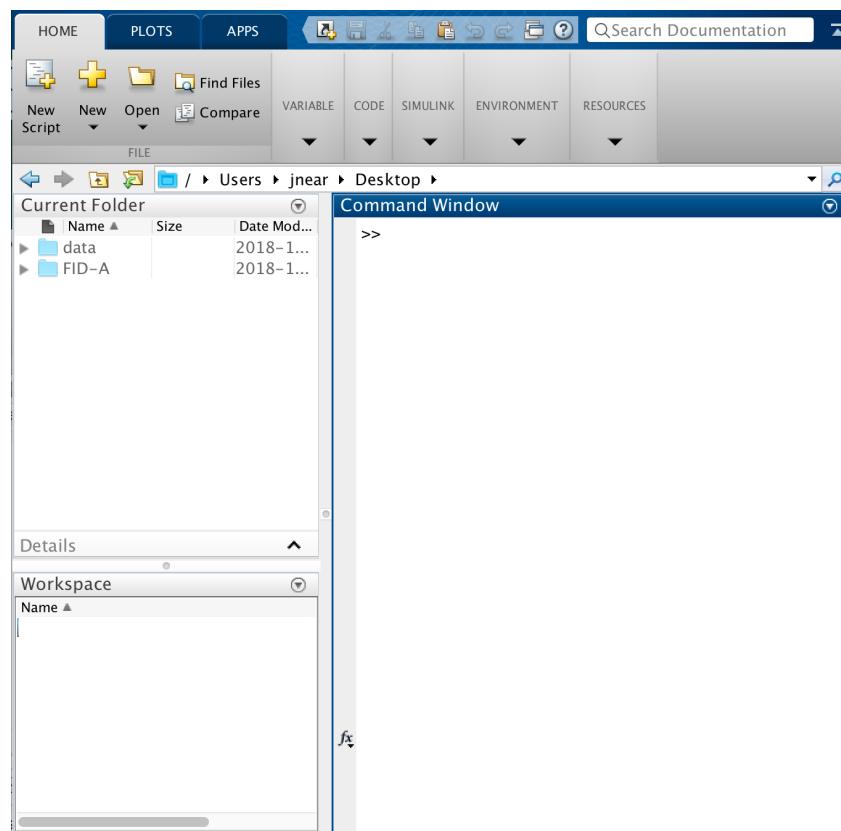
`FID-A/simulationTools/Metabolites/`

Load the Lactate spin system:

```
load Lac
```

The resulting variable “sysLac” is a structure array with 4 fields:

- J
- shifts
- name
- scaleFactor

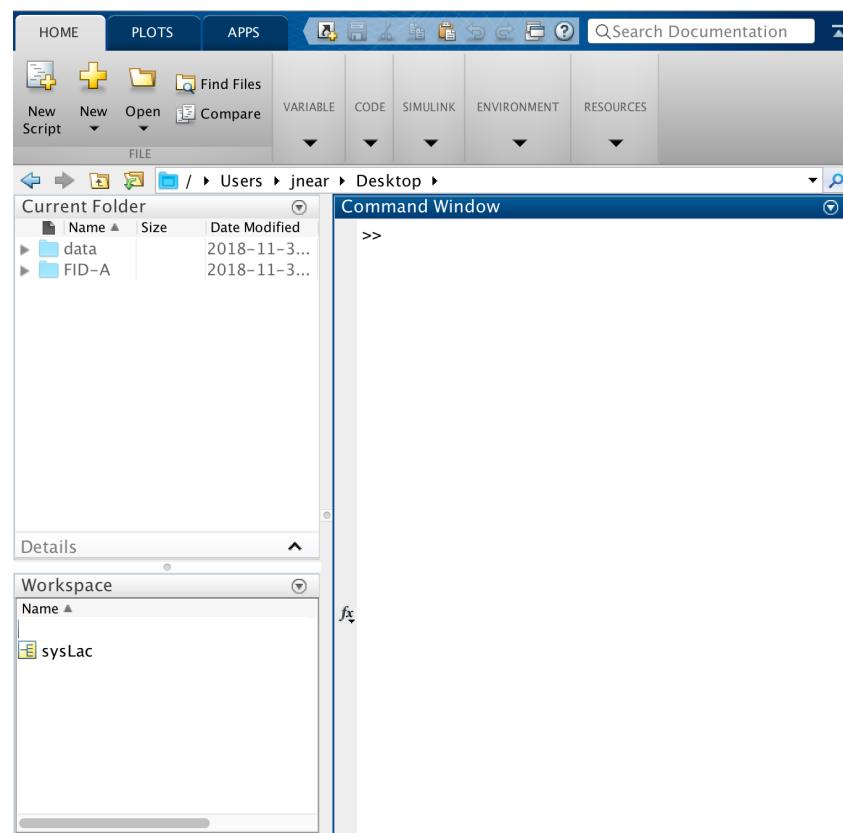


FID-A spin system definitions

Let's have a look at the values of the chemical shifts and coupling constants:

```
sysLac(1).shifts
```

```
sysLac(1).J
```



Full list of FID-A simulation tools:

Basic Spin Operators

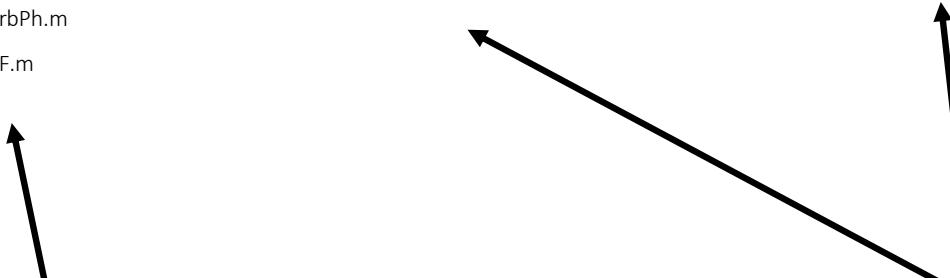
- sim_dAdd.m
- sim_Hamiltonian.m
- sim_evolve.m
- sim_excite.m
- sim_excite_arbPh.m
- sim_gradSpoil.m
- sim_readout.m
- sim_rotate.m
- sim_rotate_arbPh.m
- sim_shapedRF.m
- sim_spoil.m

Pulse Sequences (Ideal)

- sim_laser.m
- sim_lcmrawbasis.m
- sim_megapress.m
- sim_onepulse.m
- sim_onepulse_arbPh.m
- sim_press.m
- sim_spinecho.m
- sim_steam.m
- sim_steam_gradSim.m

Pulse Sequences (shaped)

- sim_make2DSimPlot.m
- sim_megapress_shaped.m
- sim_megapress_shapedEdit.m
- sim_megapress_shapedRefoc.m
- sim_megaspecial_shaped.m
- sim_onepulse_shaped.m
- sim_press_shaped.m
- sim_spinecho_shaped.m



The Basic Spin Operators form the building blocks for all of the Pulse Sequences.

Example simulation CODE:

Pulse-and-acquire sequence (sim_onepulse.m):

Usage:

```
out=sim_onepulse(np, sw, B0, lw, sys);
```

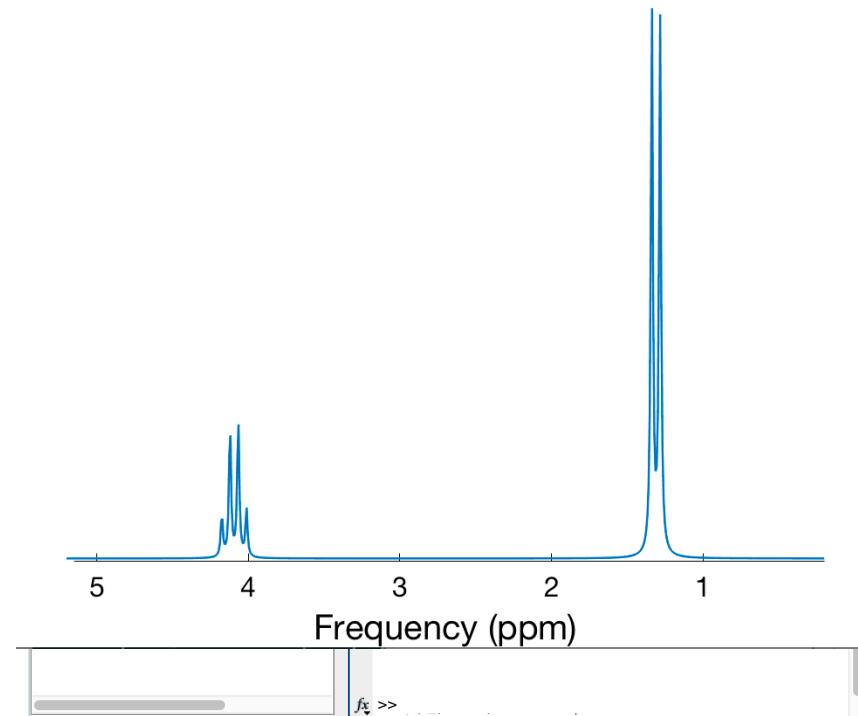
```
%Calculate Hamiltonian matrices and starting density matrix.  
[H,d]=sim_Hamiltonian(sys,Bfield);  
|  
%BEGIN PULSE SEQUENCE*****  
d=sim_excite(d,H,'x'); %EXCITE  
[out,dout]=sim_readout(d,H,n,sw,linewidth,90); %Readout along y (90 degree phase);  
%END PULSE SEQUENCE*****
```

Pulse-and-acquire Lactate simulation



Let's simulate a basic pulse-and-acquire sequence on the Lactate spin system:

And plot the result:



Example simulation CODE:

PRESS sequence (sim_press.m):

Usage:

```
out=sim_press(np, sw, B0, lw, sys, te1, te2);
```

```
%Calculate Hamiltonian matrices and starting density matrix.
[H,d]=sim_Hamiltonian(sys,Bfield);

%BEGIN PULSE SEQUENCE*****
d=sim_excite(d,H,'x'); %EXCITE
d=sim_evolve(d,H,tau1/2); %Evolve by tau1/2
d=sim_rotate(d,H,180,'y'); %First 180 degree refocusing pulse about y' axis.
d=sim_evolve(d,H,(tau1+tau2)/2); %Evolve by (tau1+tau2)/2
d=sim_rotate(d,H,180,'y'); %second 180 degree refocusing pulse about y' axis.
d=sim_evolve(d,H,tau2/2); %Evolve by tau2/2
[out,dout]=sim_readout(d,H,n,sw,linewidth,90); %Readout along y (90 degree phase);
%END PULSE SEQUENCE*****
```

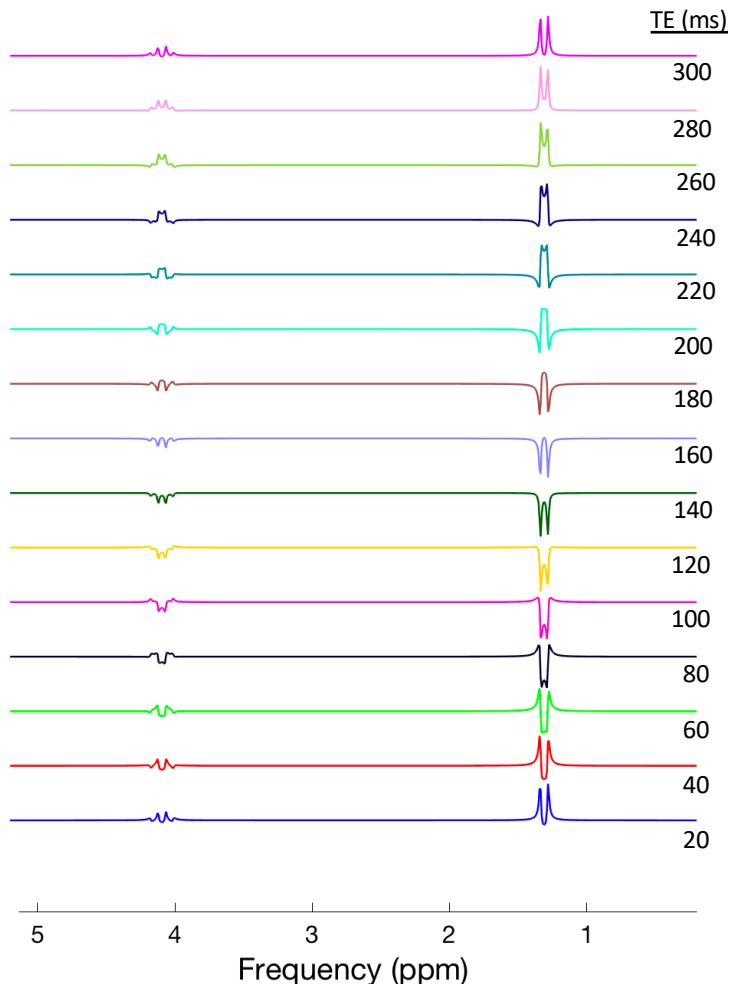
PRESS Lactate sin

Now let's quickly generate some PRESS spectra of Lactate with a series of different echo times:

And plot the results:

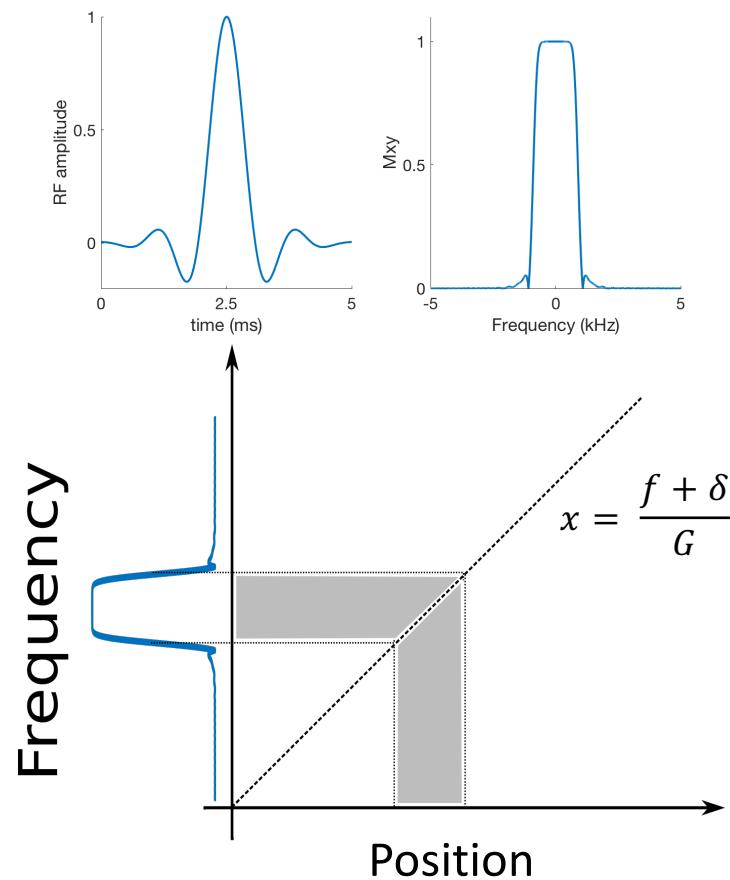
Problem: Simulations using ideal (instantaneous) RF pulses do not agree well with in-vitro experiments

Solution: Take into account chemical shift effects



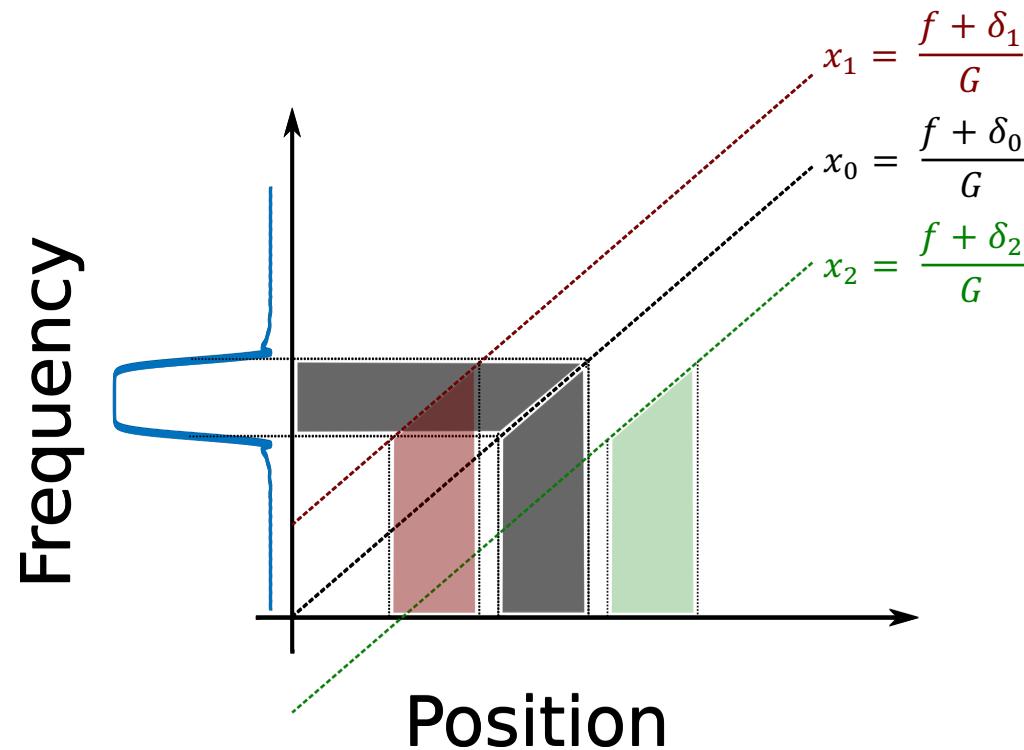
Slice selective excitation

- Spatial localization is performed using “slice selective excitation”:
 - Frequency selective RF pulse + gradient.
- Gradient creates a mapping (M) between position and frequency.
- Problem: This mapping depends on chemical shift, δ !



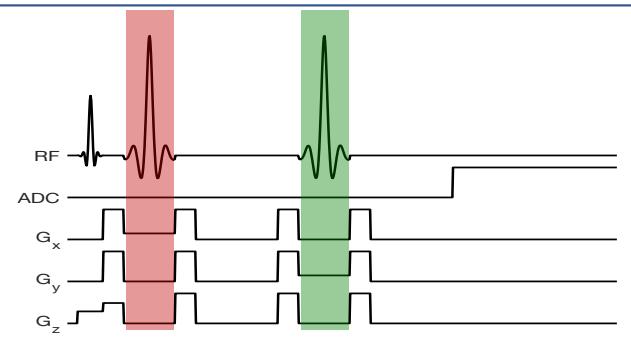
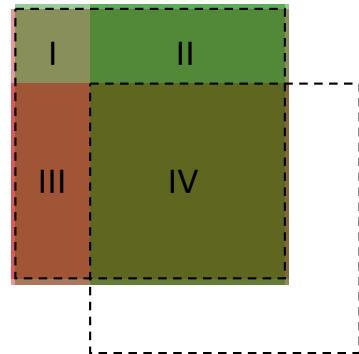
Chemical shift displacement (CSD)

- Result: spins with different chemical shifts are excited in different locations



CSD in localized MRS for coupled spin systems

- I – Spin X experiences neither refoc pulse
- II – Spin X experiences only the 2nd refoc pulse
- III – Spin X experiences only the 1st refoc pulse
- IV – Spin X experiences both refoc pulses

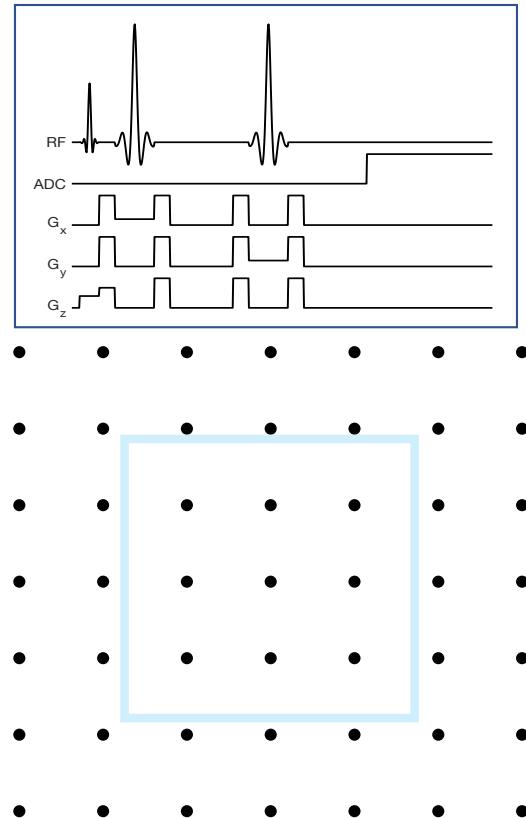


This phenomenon has a few consequences:

- The spectral shape varies within the region of interest (true for both standard PRESS and MEGA-PRESS experiments)
- Accurate simulations must take into account the rf pulse shapes and chemical shift displacement

Accounting for CSD

- Chemical shift displacement is accounted for by repeating the simulation at an array of spatial positions.
- The exact pulse waveform and gradient strength are taken into account.
- Outer volume signals removed either by phase cycling or spoiler gradients.



Advanced simulation with shaped RF waveforms:

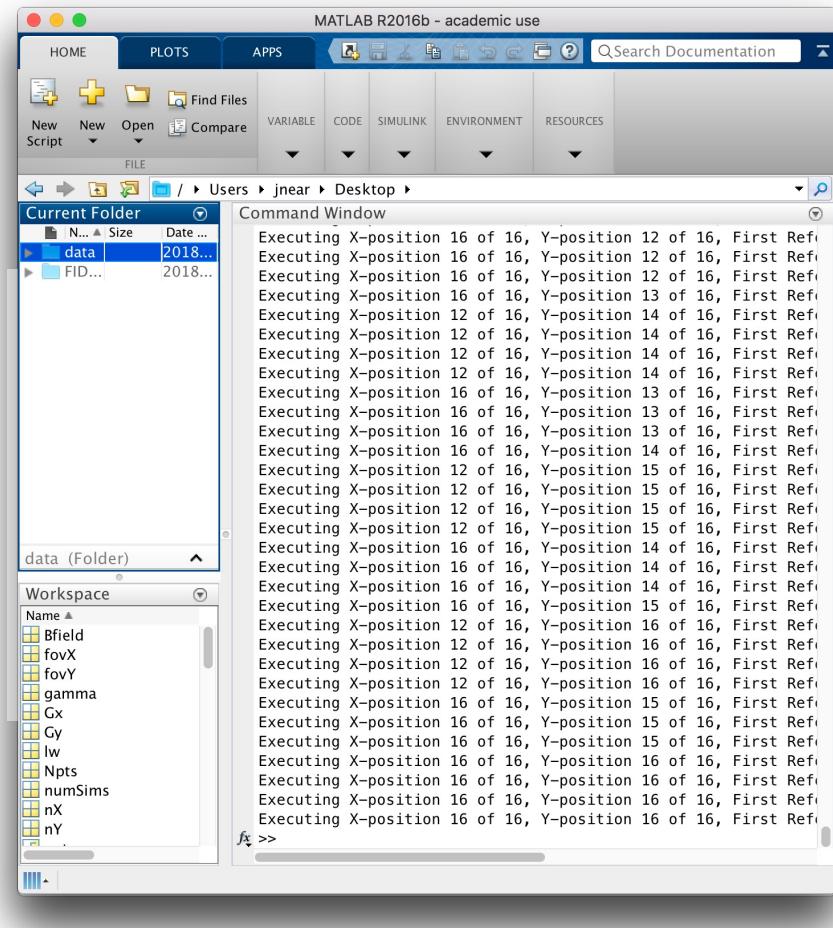
FID-A has example scripts for such spatially resolved simulations. To view the example script for shaped PRESS simulations, type:

```
edit run_simPressShaped.m
```

To run, first adjust input parameters to desired values. We will now run a 3T PRESS simulation for Lactate with TE=135 ms, and a 16 x 16 spatial grid. To run, type the name of the script at the command line:

```
run_simPressShaped
```

Run time < 3 minutes with two-fold parallelization. Two main outputs of interest: the spatially resolved simulated spectra ('out_posxy'), and the result after summing over all space ('out'). Plot both:



Plotting the simulation results

Plot spatially resolved simulation results using `sim_make2DSimPlot.m`. First, focus on the 1.3 ppm Lactate resonance:

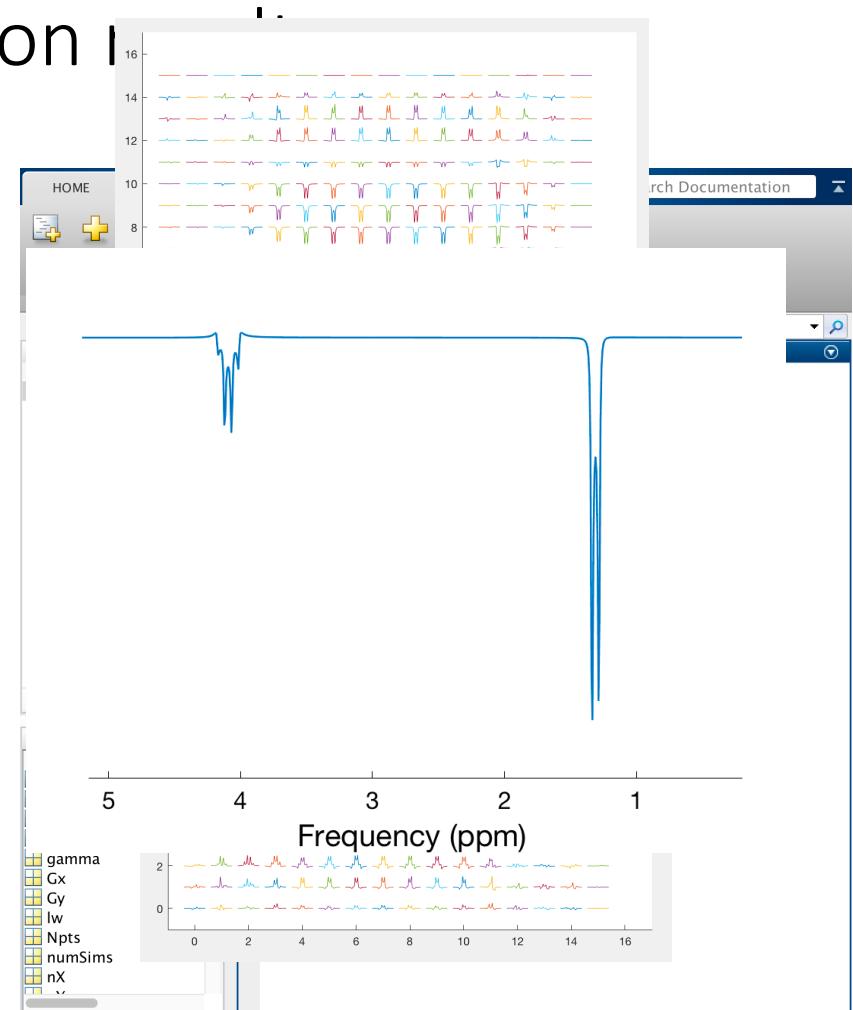
```
sim_make2DSimPlot(out_posxy,1.1,1.5);
```

Now view the 4.1 ppm Lactate resonance:

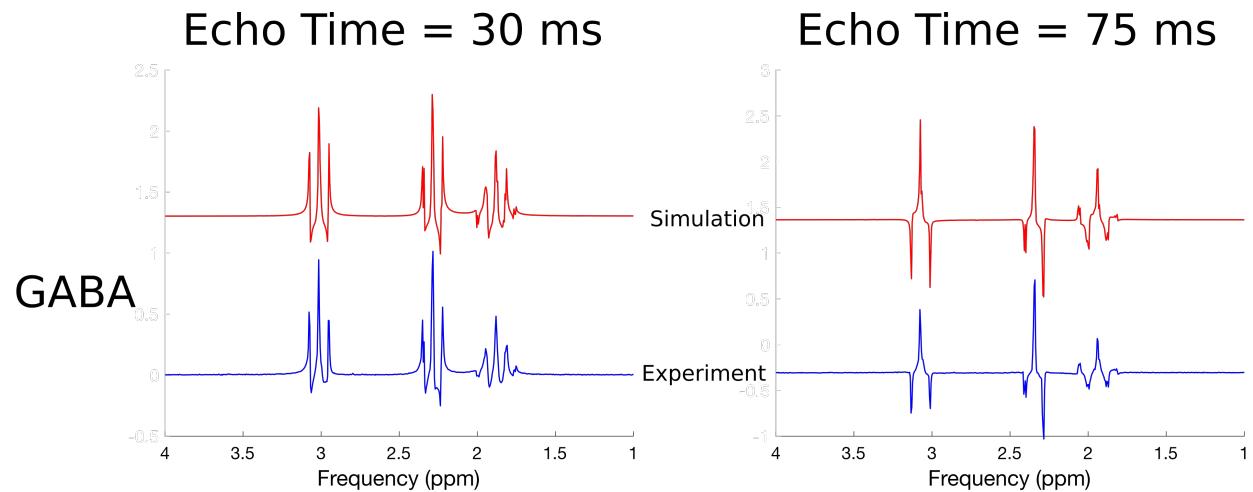
```
sim_make2DSimPlot(out_posxy,3.9,4.3)
```

Finally, view the result after summing over all space:

```
op_plotspec(out);
```



Simulation accuracy:



Simulating MEGA-PRESS

- Requires subtraction of an edit-off and an edit-on simulation.
- FID-A has a few scripts for MEGA-PRESS simulations.
- For non-spatially-resolved simulations:
 - `sim_megapress.m`
 - `run_simMegaPressShapedEdit.m`
- For spatially-resolved simulations:
 - `run_simMegaPressShaped.m`
 - `run_simMegaPressShaped_fast.m`
 - `run_simMegaPressShapedRefoc.m`

Simulating mega-press

run_simMegaPressShapedEdit.m:

```
% *****INPUT PARAMETERS*****
editWaveform='sampleEditPulse.pta'; %name of editing pulse waveform.
editOnFreq=1.88; %frequency of edit on pulse[ppm]
editOffFreq=7.4; %frequency of edit off pulse[ppm]
editTp=20; %duration of editing pulses[ms]
Npts=2048; %number of spectral points
sw=2000; %spectral width [Hz]
Bfield=3; %magnetic field strength [Tesla]
lw=2; %linewidth of the output spectrum [Hz]
taus=[5,... %Time from excitation to 1st refoc pulse [ms]
    17,... %Time from 1st refoc pulse to 1st editing pulse [ms]
    17,... %Time from 1st editing pulse to 2nd refoc pulse [ms]
    17,... %Time from 2nd refoc pulse to 2nd editing pulse [ms]
    12]; %Time from 2nd editing pulse to ADC onset [ms]
spinSys='GABA'; %spin system to simulate
centreFreq=3.0; %Centre Frequency of MR spectrum [ppm];
editPhCyc1=[0 90]; %phase cycling steps for 1st editing pulse [degrees]
editPhCyc2=[0 90 180 270]; %phase cycling steps for 2nd editing pulse [degrees]
% *****END OF INPUT PARAMETERS*****
```

sim_megapress_shapedEdit.m:

```
%BEGIN PULSE SEQUENCE*****
d=sim_excite(d,H,'x');
d=sim_evolve(d,H,delays(1)/1000);
d=sim_rotate(d,H,180,'y');
d=sim_evolve(d,H,delays(2)/1000);
d=sim_shapedRF(d,H,editPulse,editTp,180,90+editPh1);
d=sim_evolve(d,H,delays(3)/1000);
d=sim_rotate(d,H,180,'y');
d=sim_evolve(d,H,delays(4)/1000);
d=sim_shapedRF(d,H,editPulse,editTp,180,90+editPh2);
d=sim_evolve(d,H,delays(5)/1000);
[out,dout]=sim_readout(d,H,n,sw,linewidth,90); %Readout along y (90 degree phase);
%END PULSE SEQUENCE*****
```

%EXCITE
 %Evolve by delays(1)
 %1st instantaneous 180 degree refocusing pulse ;
 %Evolve by delays(2)
 %1st shaped editing pulse rotation
 %Evolve by delays(3)
 %2nd instantaneous 180 degree refocusing pulse ;
 %Evolve by delays(4)
 %2nd shaped editing pulse rotation
 %Evolve by delays(5)

Simulating MEGA-PRESS

Let's run a basic MEGA-PRESS simulation using `run_simMegaPressShapedEdit`. This function runs a quick MEGA-PRESS simulation with instantaneous refocusing pulses, and shaped editing pulses.

As before, adjust the parameters in the script as desired, and then type the name of the script at the command line:

```
run_simMegaPressShapedEdit;
```

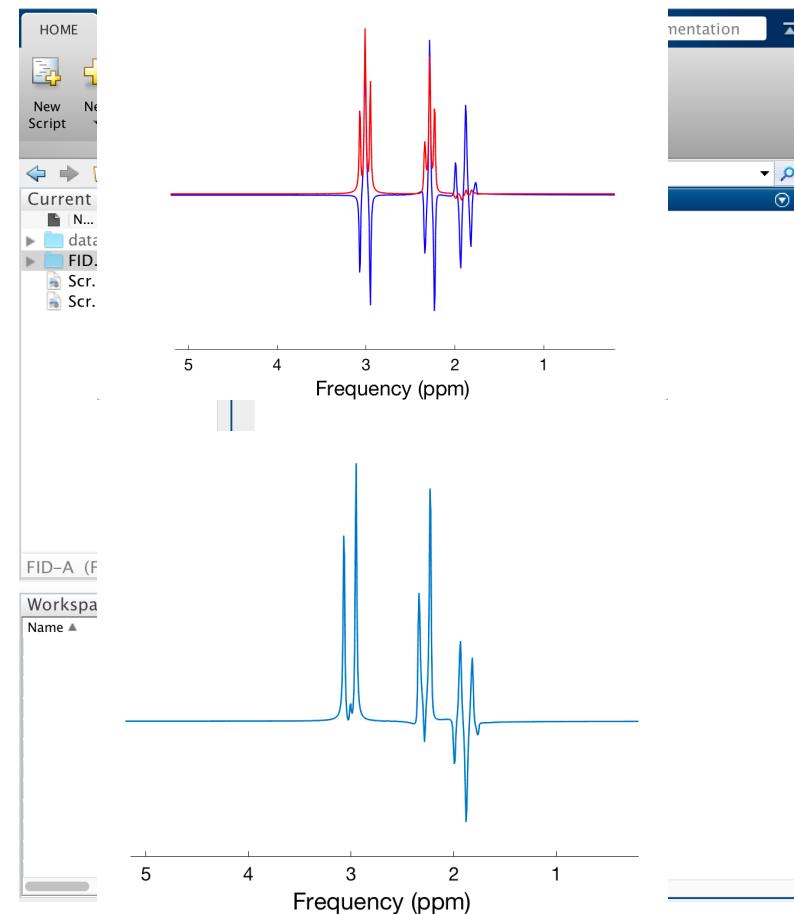
Run time < 1 minute. Three main outputs of interest: the edit-off spectrum ('outOFF'), the edit-on spectrum ('outON'). Plot both:

```
op_plotspec({outOFF,outON});
```

We can also now generate the MEGA-PRESS difference spectrum by subtracting the OFF spectrum from the ON spectrum:

```
diff=op_subtractScans(editON,editOFF);
```

```
op_plotspec(diff);
```



Generating LCModel basis spectra

- For basis sets, use the most accurate possible simulation given your time constraints.
 - Spatially resolved with high-resolution (at least 32 x 32)
 - Using the exact pulse waveforms and timings to be used experimentally
- Once you have generated a simulated spectrum that you are happy with, you can write it into LCModel “.RAW” format using:

```
[~]=io_writelcmraw(diff,'GABA.RAW','GABA');
```

- Repeat the above process for every metabolite of interest. Then, collect the .RAW files on your LCModel server, and run the LCModel command ‘makebasis’ to convert the .RAW files into an LCModel .basis file.

Summary

- The FID-A Toolkit provides a command line environment for MRS data processing, simulation and basic RF pulse design.
- User has high degree of control, flexibility and efficiency.
- Open source provides a platform for users to build upon.

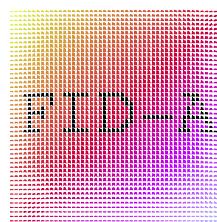
Thank You!!

- My lab
 - Jay Hennessy
 - Chathura Kumaragamage
 - Dan Madularu
 - Caitlin Fowler
 - Katrina Cruickshank
 - Dana Goerzen
 - Masoumeh Dehghani
 - Mallar Chakravarty
 - Gabriel Devenyi
- MPI Tübingen
 - Philip Ehses
- UWO
 - Martyn Klassen
- JHU
 - Richard Edden
 - Kimberly Chan
 - Mark Mikkelsen
 - Georg Oeltzschner
- University of Calgary
 - Ashley Harris

- Brenden Kadota
- Sneha Senthil
- Peter Truong
- Jason Rock
- Emily Hiles
- Lubna Burki
- Alejandra Castilla
- Colleen Bailey

- Oxford
 - Phil Cowen
 - Peter Jezard
 - Robin Simpson
 - Beata Godlewska
 - Matthew Taylor

- Funding Sources
 - NSERC
 - CIHR
 - FQRNT



Any Questions?

FID-A Simulation Tools

- Contains:
 - List of common spin-systems for in-vivo MRS
 - Basic spin operators
 - Pulse sequences with instantaneous RF pulses
 - Pulse sequences with shaped RF pulses
 - Tools for spatially resolved simulations