

jMRUI2XML Help file

1. Introduction

The present document is aimed at presenting the jMURI2XML plugin for jMRUI. The plugin was designed with two purposes in mind: 1) automating, to a certain degree, MRS processing, and 2) as a platform for outputting exchangeable data for its use in pattern recognition (but not limited to this). The plugin works with any kind of data that can be opened by jMRUI and outputs in XML format.

The rest of the document is split into the following sections – *Interface description* (provides a brief description on how to use the interface and what it does), *Short Tutorial* (describes a complete workflow on a SV and a MV spectra from input to output), and *Technical notes* (special issues to take into consideration when using the plugin).

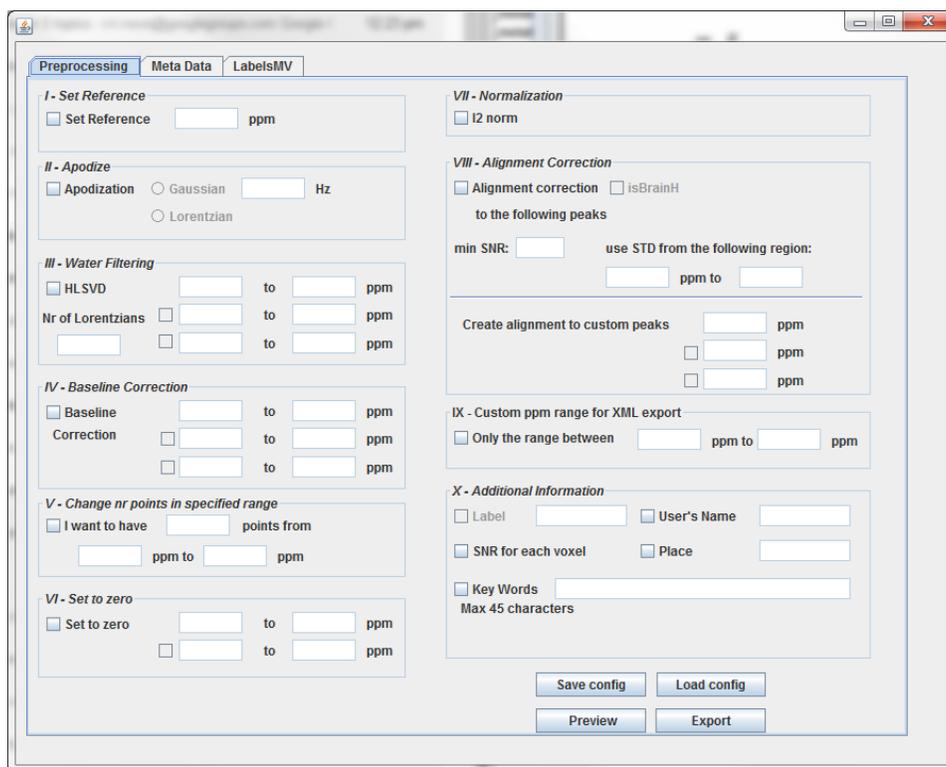
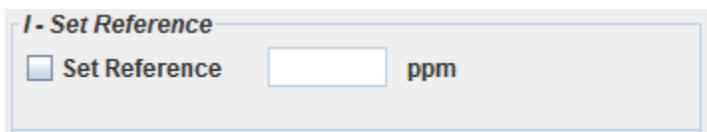


Figure 1: Main Interface. The roman numbering indicates the order of the preprocessing steps. A preprocessing step may be chosen by selecting its' designated checkbox or can be simply skipped.

2. Interface description

Some of the processing steps that can be performed from the jMRUI2XML plugin are characteristic to this plugin (represented by an asterisk at the end of the heading) whereas others leverage already existing jMRUI implementations. This section is dedicated to describing all the steps that can be performed with a more detailed description for the new steps introduced by the plugin. As a general rule, if one wishes to perform a step then the checkbox associated with that step must be ticked. Preprocessing steps are performed sequentially – one after another in the order of their numbering. The interface consists of three tabs: *Preprocessing*, *Meta Data* and *LabelsMV* (see Figure 1). The first tab contains all the preprocessing steps along with some additional information that can be added to the output. The second tab contains a text-editor like interface that can also be added to the output, and the last tab will be used in the case of MV to set labels for each voxel.

2.1. Set Reference

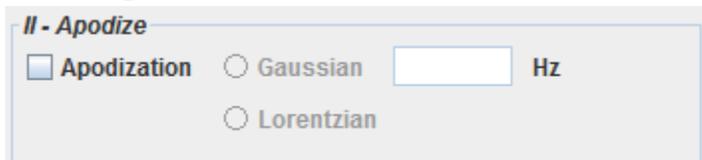


I - Set Reference

Set Reference ppm

This is the first step that can be performed and it takes as input a value (in ppm) that will represent the new reference. For example, we will set the reference to 4.75 ppm for proton spectra.

2.2. Apodize



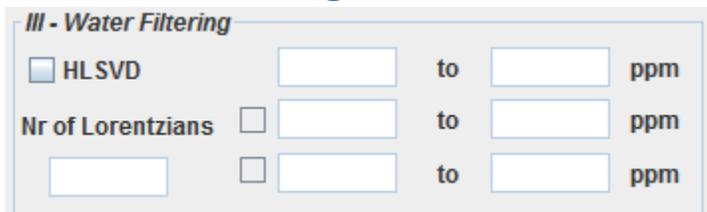
II - Apodize

Apodization Gaussian Hz

Lorentzian

One can choose between using a Gaussian or a Lorentzian function for smoothing the FID. The input value represents the frequency of this function (in Hz).

2.3. Water Filtering



III - Water Filtering

HLSVD to ppm

Nr of Lorentzians to ppm

to ppm

Although this step is named *Water Filtering* you may choose to filter out whatever suits your needs using the HLSVD algorithm. As input values you may name the number of Lorentzians to use, and up to

three areas in which to filter (in ppm; it makes no difference what you input first the higher or the lower range of an area).

2.4. Baseline Correction*

IV - Baseline Correction

Baseline to ppm

Correction to ppm

to ppm

Similarly to the previous step you may choose up to three areas but this time around you use the areas (units in ppm) for performing baseline correction. The algorithm used does the following: take the means of the specified ppm intervals (yes, you may also use only one). Sums up these means and takes their mean followed by subtracting this value out of each element in the FFT vector. As one can see the algorithm can only correct for a baseline that is not sloped.

2.5. Change number of points in specified range*

VI - Change nr points in specified range

I want to have points from

ppm to ppm

As the name implies you input the number of points (an integer) that you would like to have in a specified range. What the algorithm does depends on the number of points you would like to have: if you have more points, then it will skip over values, if you have less it will zero pad.

2.6. Set to zero*

V - Set to zero

Set to zero to ppm

to ppm

Using this step will ensure you have zero values in the areas selected (units in ppm).

2.7. Normalization*

VII - Normalization

l2 norm

Currently the only available implementation is the [l² norm](#) (aka unit length normalization, Euclidean norm).

2.8. Alignment Correction*

VIII - Alignment Correction

Alignment correction isBrainH

to the following peaks

min SNR: use STD from the following region:

ppm to

Create alignment to custom peaks ppm

ppm

ppm

This step will correct for slight (± 10 points in the FFT vector, note that this means a smaller ppm range if for higher resolution data) misalignments of the spectra according to a theoretical value of some peak(s). If you choose this step the first thing is to let the system know if the spectra you are trying to align are brain proton spectra. If it is and you check the checkbox then all the fields are automatically completed and the system will try and align either to choline (3.21 ppm), creatine (3.03 ppm) or lipids (1.29 ppm) - you just have to choose the minimum SNR for the whole spectrum. If the SNR of a peak is lower than the minimum SNR you inputted then it will not be considered in the alignment algorithm. You can look at the minimum SNR as a threshold to perform alignment or not; further along, in our examples we take a minimum SNR of 10 as in [2] In the case of brain proton spectra at 1.5 T we use the alignment algorithm described by Perez-Ruiz et al. in [1] . The SNR is computed using the following formula:

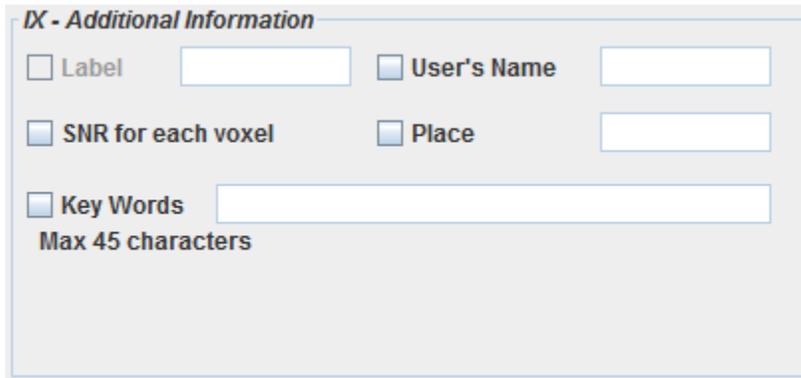
$$SNR = \frac{Max\ Peak}{2 * STD(noise)}$$

Where *Max Peak* represents the highest amplitude in the FFT, *STD* stands for standard deviation and in this particular case we consider the noise in the range [minimum ppm to 0 ppm], but you can input your own custom range from where to compute the *STD*. It might be that you do not want to align using the above peaks, don't want to use the above alignment correction algorithm or simply using a different type of spectra- in that case you don't need to check the "isBrainH" checkbox and can proceed to selecting your own parameters. Similarly, the first thing to choose is the minimum SNR followed by the range you would like to be used for computing the *STD* (and implicitly the SNR) and finally the peaks to use for alignment. In this case the algorithm will try to align according to the peak with the highest SNR.

2.9. Custom ppm range for XML export*

Even though the user may choose in the fifth (*Change number of points in specified range*) step to have a specific number of points in a given ppm interval it does not mean that the plugin will output only that range. This step, 2.9, is used when the user only needs a specific ppm range to be present in the output. If this step does not make sense yet please read the examples.

2.10. Additional Information¹



DX - Additional Information

Label User's Name

SNR for each voxel Place

Key Words
Max 45 characters

This step allows for additional information to be added to the XML output; two of the fields, *User's name* and *Place*, are compulsory meaning that an empty XML file will be generated when these fields are not selected or left empty.

In the case one processes a SV then the Label field activates and a label can be attached to the output. You might also want to know the SNR for each voxel in which case you just need to tick the *SNR for each voxel* (works for SV and MV both). The SNR is computed as previously only using the highest amplitude in the spectrum. You can also attach several keywords to save in the XML if one needs a specific reminder for that specific case. Going along these lines I would like to remind the reader that if you need to record extensive notes you may do so in the *Meta Data* tab.

2.11. Control Buttons



Save config Load config

Preview Export

Imagine the following scenario: you have just processed some spectra with your parameters and you'd like to use the same parameters for all the data that you will acquire in the near future. You can do so by

¹ Not marked with a star because there are no algorithms present, but please note that none of the functionality presented here is available anywhere else in jMRUI.

29 JAN 2015
Version 1.1
Victor Mocioiu

pushing *the Save config* button – your parameters will be saved as a an XML file that can later be loaded by clicking the *Load config* button.

Now if you would like to know how your spectra will look under a set of parameters you can use the toggle button *Preview/Original*. This will process the data and show how it was changed on the jMRUI display. After you have finished with the processing you can export by using the *Export* button, which saves the processed spectrum to the directory of your choice.

3. Short Tutorial

3.1. SV example

We start off with a short TE (31 ms) SV spectrum coming from a tumour patient that participated in the INTERPRET project - Figure 2. We will follow the pipeline presented in [2] with the improvements proposed in[3] , namely:

- Set reference to 4.75ppm.
- Line broadening with Lorentzian of 1 Hz.
- Filter water (4.31-5.11 ppm) using the HLSVD algorithm with 10 Lorentzians.
- Baseline Correction (taking into account the 9 to11 ppm and -1 to -2 ppm ranges).
- Ensure the spectrum has 512 points in the 2.7 to 7.1 ppm range.
- Set to zero the values of the FFT in the 4.2 to 5.1 ppm range.
- Normalize according to l^2 (Euclidean) norm.
- Correct for misalignments (min SNR 10, use 0 to 3.07 ppm for computing STD, align to choline, creatine, lipids)
- We will also set a label for the spectrum – namely *tumour*, export its SNR, and give it the key words “Short TE, INTERPRET”;

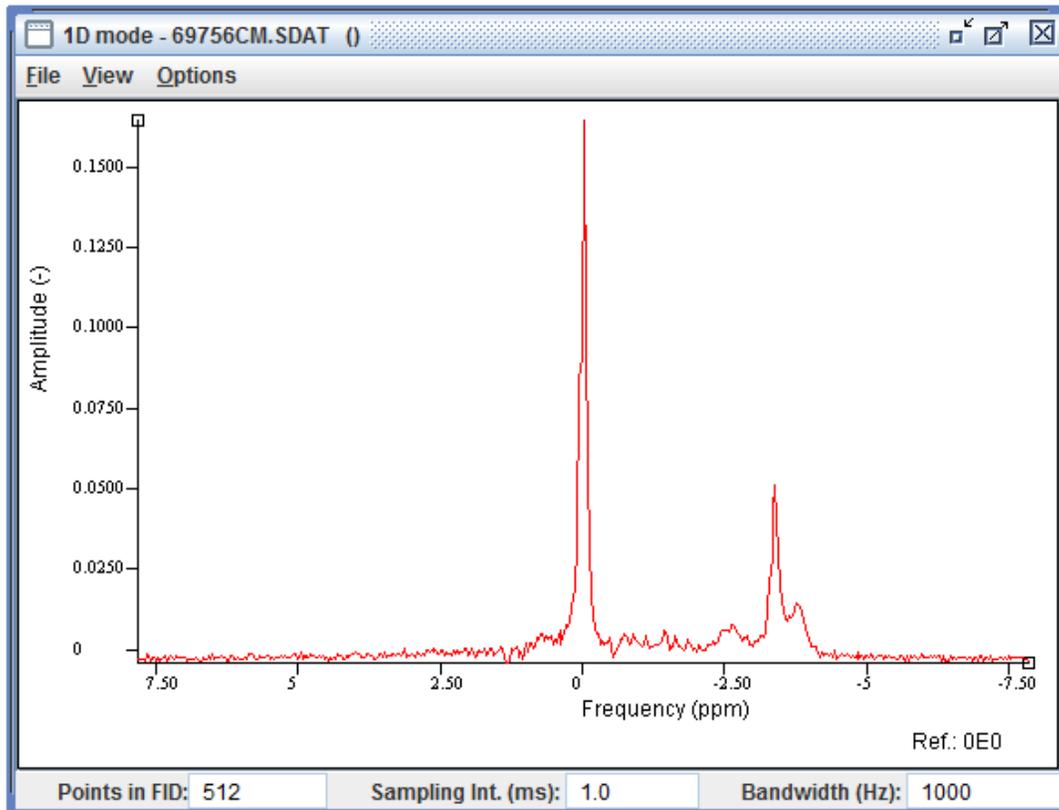


Figure 2: Unprocessed FFT of a 1H Brain spectrum from a healthy volunteer

We will load the following XML template, which we will refer to as the INTERPRET canonical format:

```
<?xml version="1.0" encoding="UTF-8"?>
- <Preprocessing>
  - <SetReference>
    <PPM>4.75</PPM>
  </SetReference>
  - <Apodization>
    - <Lorentzian>
      <Hz>1</Hz>
    </Lorentzian>
  </Apodization>
  - <WaterFiltering>
    <NumberOfLorenzians>10</NumberOfLorenzians>
    <MinPPM>4.31</MinPPM>
    <MaxPPM>5.11</MaxPPM>
  </WaterFiltering>
  - <BaselineCorrection>
    <MinPPM>9.0</MinPPM>
    <MaxPPM>11.0</MaxPPM>
    <MinPPM>-2.0</MinPPM>
    <MaxPPM>-1.0</MaxPPM>
  </BaselineCorrection>
  - <NumberOfPoints>
    <NrPoints>512</NrPoints>
    <MinPPM>-2.7</MinPPM>
    <MaxPPM>7.1</MaxPPM>
  </NumberOfPoints>
  - <SetToZero>
    <MinPPM>4.2</MinPPM>
    <MaxPPM>5.1</MaxPPM>
  </SetToZero>
  <SetToUnitLength>true</SetToUnitLength>
  - <AlignmentCorrection>
    <isBrainH>true</isBrainH>
    <minSNR>10.0</minSNR>
    <minSTD>0</minSTD>
    <maxSTD>-3.07</maxSTD>
    <FirstPeak>3.21</FirstPeak>
    <SecondPeak>3.03</SecondPeak>
    <ThirdPeak>1.29</ThirdPeak>
  </AlignmentCorrection>
  - <CustomPPMRange>
    <FirstPPM>-2.7</FirstPPM>
    <LastPPM>7.1</LastPPM>
  </CustomPPMRange>
  - <AdditionalInformation>
    <Name>Victor</Name>
    <Place>UAB</Place>
  </AdditionalInformation>
</Preprocessing>
```

The interface will now have all the fields autocompleted; if we want to see what the spectrum will look like after processing we just need to click the *Preview/Original* button. Of course, we could play with the parameters of the preprocessing steps and see how they affect the spectra, but that is outside of the scope of this tutorial. If the alignment was successful a message pops up letting you know what peak was used. In the case of proton brain spectra the possibilities are *Choline*, *Creatine* and *Lipids* as for

other spectra the possibilities are *First Peak*, *Second Peak*, and *Third Peak*. In our example Alignment was performed using the lipids peak. Figures 3 and 4 illustrate the interface with the completed fields with the aforementioned alignment popup, and the pre-processed spectra.

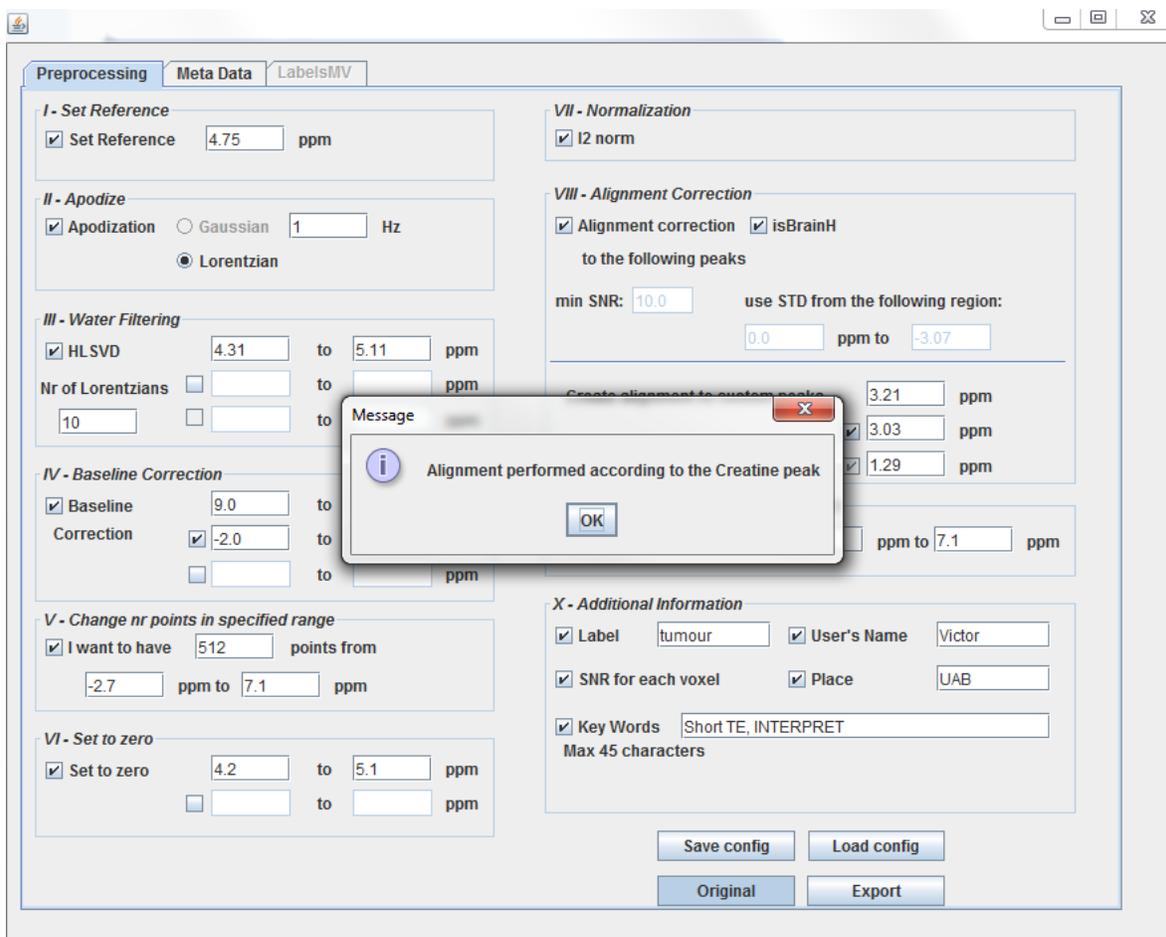


Figure 3: Filled out interface. After pressing the Preview/Original button the spectrum gets preprocessed (see the following figure) and we are informed that the spectrum was aligned taking into account the creatine peak.

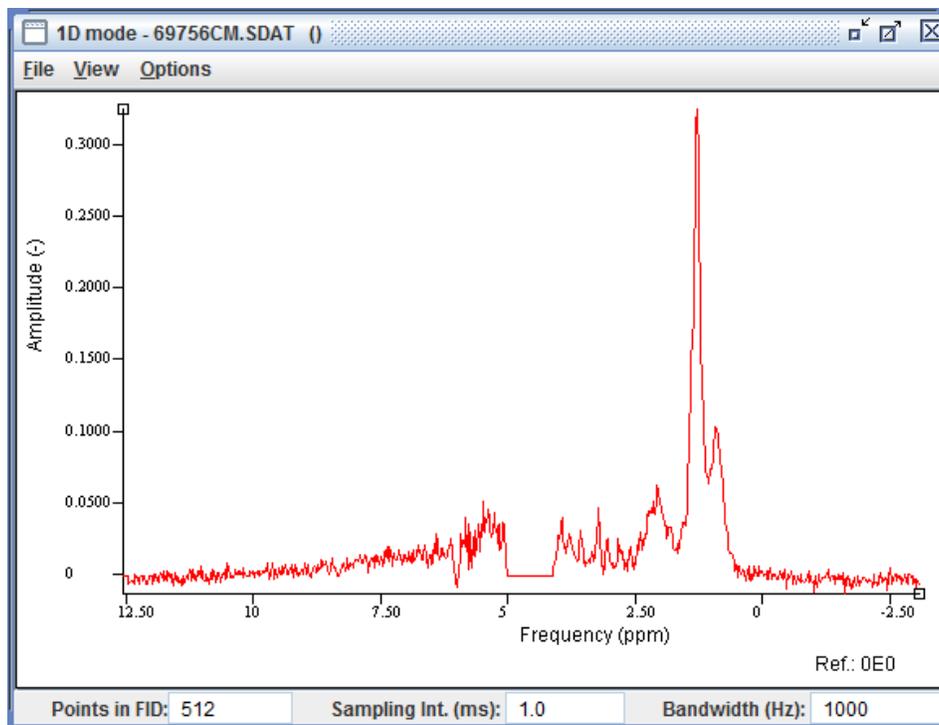


Figure 4: The spectrum after undergoing the INTERPRET canonical format pipeline.

Seeing as we are satisfied with the results we'd like to save them and later apply some statistical analysis to a set of such spectra. In order to do so a label would be advisable, and because we want to remember that this spectrum was of short TE and from the INTERPRET project some key words will also prove useful. The exported XML file looks like this:

```

<?xml version="1.0" encoding="UTF-8"?>
- <DATASET Version="1.0" Date="2015-02-02 11:36:40" CreatedBy="jMRUI2XML">
  - <Preprocessing>
    - <SetReference>
      <PPM>4.75</PPM>
    </SetReference>
    - <Apodization>
      - <Lorenzian>
        <Hz>1</Hz>
      </Lorenzian>
    </Apodization>
    - <WaterFiltering>
      <NumberOfLorenzians>10</NumberOfLorenzians>
      <MinPPM>4.31</MinPPM>
      <MaxPPM>5.11</MaxPPM>
    </WaterFiltering>
    - <BaselineCorrection>
      <MinPPM>9.0</MinPPM>
      <MaxPPM>11.0</MaxPPM>
      <MinPPM>-2.0</MinPPM>
      <MaxPPM>-1.0</MaxPPM>
    </BaselineCorrection>
    - <NumberOfPoints>
      <NrPoints>512</NrPoints>
      <MinPPM>-2.7</MinPPM>
      <MaxPPM>7.1</MaxPPM>
    </NumberOfPoints>
    - <SetToZero>
      <MinPPM>4.2</MinPPM>
      <MaxPPM>5.1</MaxPPM>
    </SetToZero>
    <SetToUnitLength>true</SetToUnitLength>
    - <AlignmentCorrection>
      <isBrainH>YES</isBrainH>
      <minSNR>10.0</minSNR>
      <minSTD>0.0</minSTD>
      <maxSTD>-3.07</maxSTD>
      <FirstPeak>3.21</FirstPeak>
      <SecondPeak>3.03</SecondPeak>
      <ThirdPeak>1.29</ThirdPeak>
    </AlignmentCorrection>
    - <CustomPPMRange>
      <FirstPPM>-2.7</FirstPPM>
      <LastPPM>7.1</LastPPM>
    </CustomPPMRange>
    - <AdditionalInformation>
      <Name>Victor</Name>
      <Place>UAB</Place>
      <KeyWords>Short TE, INTERPRET</KeyWords>
    </AdditionalInformation>
  </Preprocessing>
  - <Case ID="2015-02-02 11:36:40">
    <Tissue Type="tumour"/>
    - <Spectrum>
      <Parameters SNR="110.6479113787693" PointsNumber="512" LastPPM="7.1" FirstPPM="-2.7"/>
      <Points>0.01382904863660728 0.007877293178865015 0.00985034272039255 0.01428977

```

Please note the fields in red - the XML file starts with the root, DATASET, having the attributes Version, Date, and CreatedBy. The value of the Version attribute corresponds to the current version of the plugin. In the current case the DATASET has two children: Preprocessing and Case. The Preprocessing field is the same as the preprocessing template; the Case field has an ID attribute (currently the same as the Date attribute of DATASET) and has two sub-children: Tissue (with the attribute Type) and Spectrum. The Spectrum field has three sub-children: Parameters (with attributes SNR, PointsNumber, LastPPM, and FirstPPM), Points (where the value of the field represents the processed spectrum), and MapPosition (with attributes Zaxis, Yaxis and Xaxis - not shown in the above XML, will come back to it in the MV example).

3.2. MV Example

We continue our set of examples off with a healthy volunteers' short TE (30 ms) MV spectrum - we will follow a similar pipeline to the above one insisting only on the distinctive aspects for MV data.

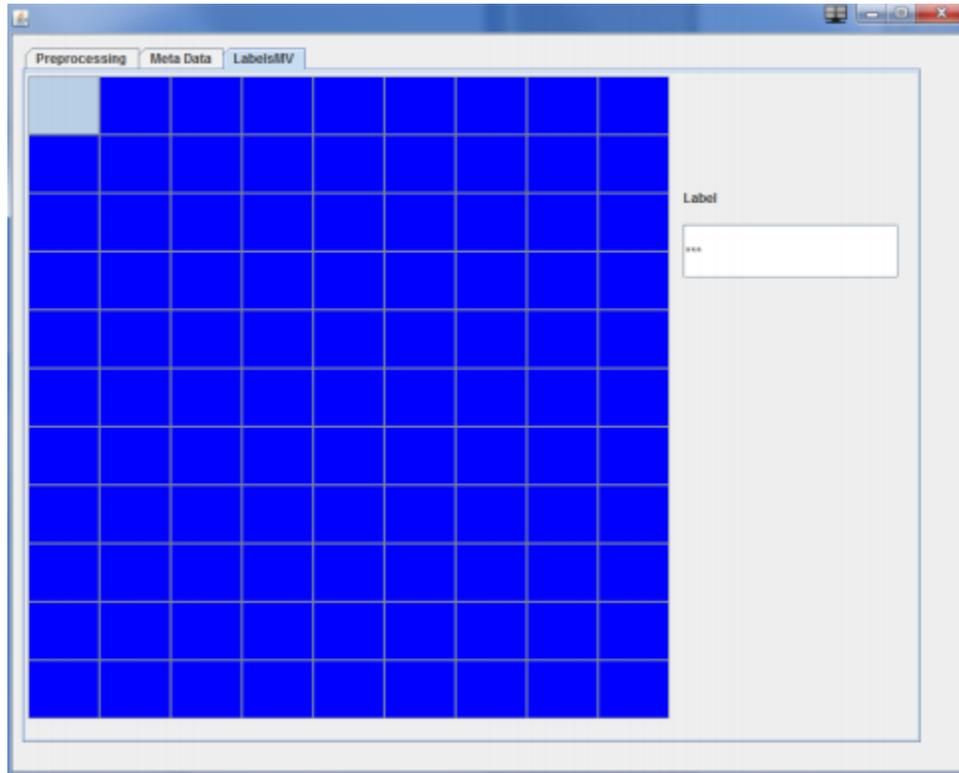


Figure 5: The LabelsMV tab holding a 9 by 11 grid. Each cell corresponds to a voxel and has its' own associated label. The label of a cell can be viewed/edited by clicking on it. In the figure the voxel top left has been selected - by default all the labels have the value of "***".

We use the same XML in INTERPRET canonical format to process the data with the mention that now we can set a label for each voxel using the *LabelsMV* tab (see Figure 5). Each cell in the grid represents a voxel; each cell has a label assigned with the default value of "***". The labels can be viewed and edited by clicking on a cell. The grid follows matrix notation, meaning the top left corner corresponds to position [1,1] and the bottom right corner corresponds to position $[m,n]$, where m is the number of rows and n is the number of columns. For illustration purposes we will label the [1,1] voxel as "First Voxel". Afterwards we follow the same steps as in the previous example. Below you can find (part) of the XML output:

```
- <Case ID="2014-10-24 13:50:59">  
  <Tissue Type="FirstVoxel"/>  
  - <Spectrum>  
    <Parameters SNR="5.3868396132169" PointsNumber="512" LastPPM="7.1" FirstPPM="-2.7"/>  
    <Points>0.05422064587665461 -0.01789127550881019 -0.040869873410109 -0.049989  
      0.02332033350734645 0.01463176012323406 0.015994738575578076 -0.00310076  
      ⋮  
    -0.012338323561276994 -0.0229297210171179 -0.048202357220069145  
    .03516064815445516 -0.02391649723823769 -0.04434765726741924  
    5 0.020206380542219397 0.0021369564152637043 0.005040225044479008</Points>  
    <MapPosition Zaxis="0" Yaxis="1" Xaxis="1"/>  
  </Spectrum>  
</Case>
```

Note that the Preprocessing element is not illustrated above to save space, but it is still present in the XML file. The red boxes highlight the label of the voxel, the SNR (as an attribute of the Parameters field) and the position of the voxel on the grid (again, according to the jMRUI convention). Also taking a swift look at the SNR attribute of the Parameters node we can tell that this voxel has not undergone the alignment algorithm because it has an SNR below the value of 10.

4. Technical Notes

The jMRUI2XML plugin uses the data it “sees” when you call it; for example if you have filtered some region of the spectrum then toggling the Preview/Original button will always return to the ALREADY filtered data - NOT the original input data. In the current version of the plugin, regarding spectral alignment on MV: you will get a popup message only for the last processed voxel (which is the top right corner). The algorithm will work for all the voxels, but visual inspection would be the only solution to be sure at the time.

While exporting a file, if you did not “preview”, the XML file gets created before the actual preprocessing so please do not try to access it in any way until you notice the processing has finished. It should however not take long – on a 16x16 MV grid with 1024 points per spectrum the full pipeline takes about 22 seconds on a 2.6 GHz processor.

When using the MV grid, if you have finished labeling voxels then remember to “de-click” the last edited cell. This way you will not lose the label of the last edited voxel. (It happens that at times one voxel isn’t saved and it is usually the last edited voxel, even if you de-clicked it. The workaround to this is to click on another previously edited voxel or just a non-edited voxel).

In my opinion the MetaData panel really comes into use if you want to save details about processing steps applied to your data that are not included in this plugin. Let’s say you want to phase the data after you have processed it and you want to export it. Any modifications of the signal, prior to or after processing with the jMRUI2XML plugin, will also be present in the exported file. In this case you might want to write the extra steps in the MetaData panel. An example isn’t provided in the tutorial but one might put in the processing that was applied to the spectrum before/after going through the pipeline – i.e. “manual phasing was performed before the pipeline”.

Bibliography

- [1] A. Perez-Ruiz, M. Julia-Sape, G. Mercadal, I. Olier, C. Majos, and C. Arus, "The INTERPRET Decision-Support System version 3.0 for evaluation of Magnetic Resonance Spectroscopy data from human brain tumours and other abnormal brain masses," *BMC Bioinformatics*, vol. 11, p. 581, 2010.
- [2] A. R. Tate, J. Underwood, D. M. Acosta, M. Julia-Sape, C. Majos, A. Moreno-Torres, *et al.*, "Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra," *Nmr in Biomedicine*, vol. 19, pp. 411-434, Jun 2006.
- [3] M. van der Graaf, M. Julia-Sape, F. A. Howe, A. Ziegler, C. Majos, A. Moreno-Torres, *et al.*, "MRS quality assessment in a multicentre study on MRS-based classification of brain tumours," *NMR Biomed*, vol. 21, pp. 148-58, Feb 2008.