seaMass Viz Tutorial

1. Get the package

Go to the Github release page (https://github.com/biospi/seamass-viz/releases), locate the '↓ seamass-viz-0.1.1-win.zip' button (see figure 1) and click on it to download the zipped release.

Unzip the file to any directory you prefer, you should have a 'readme' file and the following five subdirectories:

- seamass-pwiz - to convert data from mzML format to smi format
- seamass - to signal decompose smi data into smv data
- seamass-vis - command line visualisation client for smv data
- example_data - example data for all the three applications listed above
- dependencies - collection of 3rd party dependencies

Figure 1

2. Converting data into smi format

The custom ProteoWizard (http://proteowizard.sourceforge.net/) 'msconvert.exe' application inside the 'seamass-pwiz' folder is used to convert MS data from mzML format into smi format.

- Requirements

  64 bit operation system

- Dependencies

.NET Framework 4 -- If not available on your computer, install it from http://www.microsoft.com/en-gb/download/details.aspx?id=17851 .

In a command window, execute 'msconvert' with parameter '--smi' and a data file name to convert the data into smi format:

>***msconvert*** *--smi input.mzML*

An example to convert one of the provided example data is presented in figure 2. The converted data file (in msi format) is saved in the current directory.

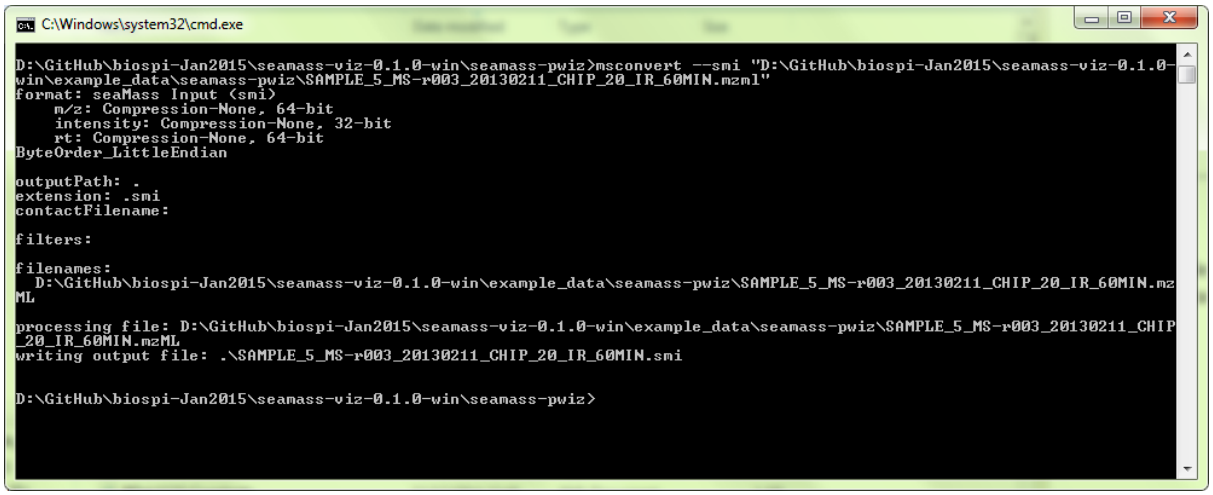

Figure 2

3. Signal decomposing smi data into smv data

The 'restoration.exe' application inside the 'seamass' folder is used to transform smi data into smv data through sparse signal decomposition.

- Requirements

  64 bit operation system

- Dependencies

  Microsoft Visual C++ 2010 x64 Redistributable -- If not available on your computer, install it by execute 'vcredist_x64.exe' in the 'dependencies' folder.

  Intel(R) C++ redistributables on Intel(R) -- If not available on your computer, install it by execute 'w_ccompxe_redist_intel64_2013.1.119.msi' in the 'dependencies' folder.

In a command window, run 'restoration' with the following six parameters to transform smi data in to smv data:

>***restoration*** *<in_file> <mz_res> <rt_res> <shrinkage> <tol> <threads> <out_type>*

<in_file>: Raw input file in seaMass Input format (smi)
Guidelines: Use pwiz-seamass to convert from mzML or vendor format

<mz_res>: MS resolution given as:

"b-splines per Th = 2^mz_res * 60 / 1.0033548378"
Guidelines: between 0 to 2 for ToF, 3 for Orbitrap

<rt_res>: LC resolution given as: "b-splines per minute = 2^rt_res"
Guidelines: around 4

<shrink>: Amount of denoising given as: "L1 shrinkage = 2^shrinkage"
Guidelines: around -4

<tol>: Convergence tolerance, given as: "gradient <= 2^tol"
Guidelines: around -9

<threads>: Number of OpenMP threads to use
Guidelines: set to amount of CPU cores or 4, whichever is smaller

<out_type>: Type of output desired
Guidelines: 0 = just viz_client input; 1 = also smo; 2 = also debug

An example to transform one of the provided example data is presented in figure 3. The converted smv data are saved in a new subdirectory of the folder where <in_file> is located.

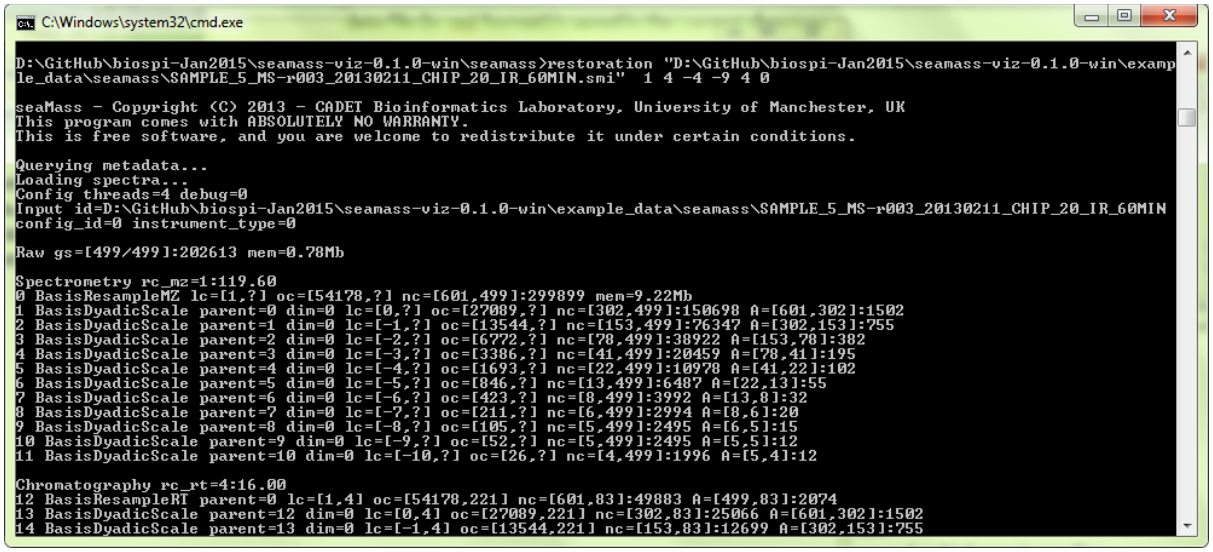

Figure 3

4. Visualizing smv data

The 'viz.exe' application inside the 'seamass-viz' folder is a command line client to visualise seamass transformed smv data.

- Dependencies

  Microsoft Visual C++ 2010 x86 Redistributable -- If not available on your computer, install it by execute 'vcredist_x86.exe' in the 'dependencies' directory.

In a command window, run 'viz' with the following eight parameters:

*>**viz** <in_idx> <mz_min> <mz_max> <rt_min> <rt_max> <out_w> <out_h> <chunk_size>*

<in_file>: Input .idx file

<mz_min>: Minimum m/z to display

<mz_max>: Maxmimum m/z to display

<rt_min>: Minimum retention time to display

<rt_max>: Maximum retention time to display

<out_w>: Output width in pixels

<out_h>: Output height in pixels

<chunk_size>: Number of coefficients to stream per output (e.g. 40000)

Among these eight parameters, <mz_min> <mz_max> <rt_min> and <rt_max> define the visualization viewport; <out_w> and <out_h> define the size of output images.

To visualise data with m/z range and/or RT ranges unknown, it is suggested to execute 'viz.exe' with a big viewport (e.g. m/z from 0 to 2000 Daltons, RT from 0 to 180 mins), then the real m/z and RT ranges (inside the specified big viewport) will be displayed in the command window, and further zooming/panning can be conducted using those ranges as guidelines.

An example to visualize one of the provided example data is presented in figure 4. The streaming output images are saved in a new subdirectory of current directory.

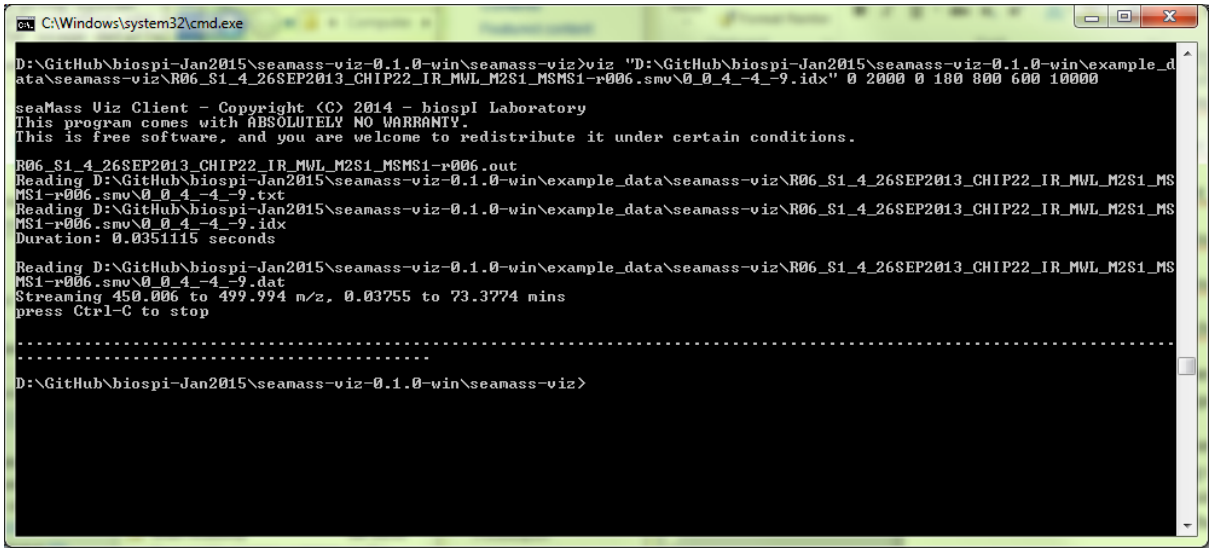

Figure 4