Tutorial 1: Getting started with GIO

1 Accessing the public GIO server

To access the public GIO server, open your web browser and enter the address `gio.sbcs.qmul.ac.uk`. You should immediately see the GIO front page, which will look something like Figure 1. This is the main interface to GIO – instructions for use are given later, in section 3.

You can explore the features of GIO and perform simple analyses right away, but to access GIO's full functionality you will need to create an account. You can do this for free by clicking on the *User* item in the black menu bar at the top of GIO, and clicking *Register*. Follow the instructions to create an account and log in. **All subsequent elements of this tutorial, and others, assume that you are logged in as a registered user.**

2 Using your own GIO server

It is possible to install some or all of GIO on your own Galaxy installation, if you have one. This is necessarily a reasonably complicated process, for which detailed instructions are provided at `https://code.google.com/p/gio-repository/`. Unless you are planning to run particularly large analyses, or want to customise GIO by adding your own tools, we recommend you use the public server, at least to start with.

3 Understanding the GIO user interface

GIO is based on Galaxy (`usegalaxy.org`), so if you've used Galaxy before the interface will be very familiar. The data analysis user interface (see Figure 1) is comprised of three panes. The leftmost pane gives access to available data analysis tools, which are clustered in groups which can be expanded by clicking on them. The rightmost pane is the *History*, which contains a list of available data files – these may have been uploaded to GIO or produced as output of GIO's tools. The central pane initially shows a welcome message, but its main purpose is to set parameters prior to executing tools, and to view the content of data files from the History. The black menu bar along the top of GIO provide a means to switch between the data analysis interface and other pages for accessing workflows, shared data and other functionality.

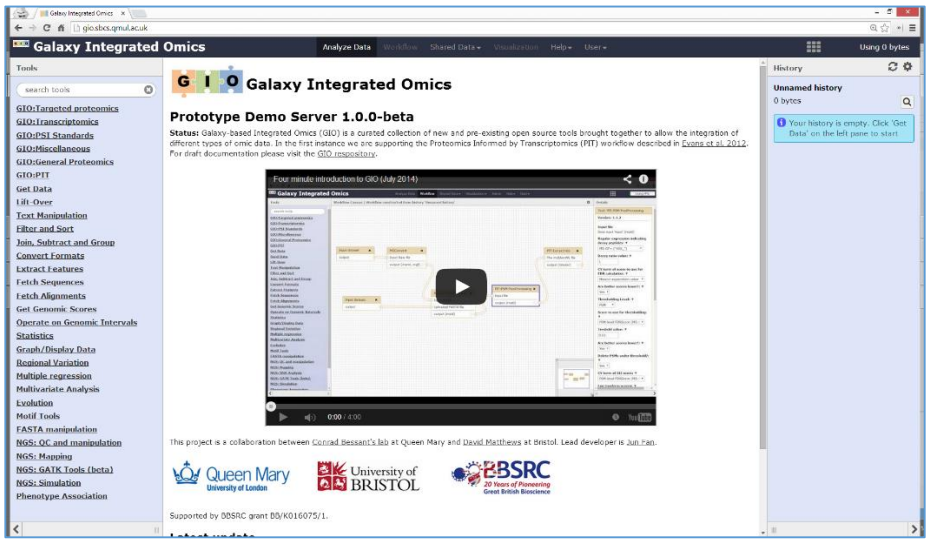

Figure 1: *GIO user interface for data analysis. Tools are accessed via the leftmost pane. The rightmost pane lists any data files present. The central pane is for viewing results and setting tool parameters.*

4 Running simple peptide and protein identification

In this section, we will use GIO to perform a simple protein identification search similar to the one shown in the GIO introductory video (`http://youtu.be/D0lPGycopCk`).

4.1 Add data to history

First of all, we need to get some data to work with. To do protein identification we need a file containing spectral data, and a file containing protein sequences to search this data against. Normally, you would upload these files by clicking on *Get Data* in the Tools pane, then clicking *Upload File* and selecting which file to upload the your History. However, to keep things simple we'll use some files that we have already make available in GIO in a shared data library. Import these files by doing the following:

1. Click *Shared Data* in GIO's menu bar and click *Data Libraries* in the dropdown list that appears.
2. From the list of available data libraries, click *PIT example*
3. From the list of files in this library, select the files `exampleSpectra.mgf` and `human.fasta` by ticking the boxes to the left of their filenames and click *Go* to import to your History.
4. Click *Analyze Data* in the menu bar to return to GIO's main user interface. You should see that the two data files are now in your History (rightmost pane). If you want to check the content of these files, you do so by clicking the eye icon (👁) next to the file name in the History pane.

4.2 Perform peptide spectrum matching

Now we'll try to match spectra in `exampleSpectra.mgf` to peptides from `human.fasta` using the popular MSGF+ search engine. To do this:

1. Click on the *GIO:General Proteomics* section in the Tools pane. The section will expand to reveal the tools within this section, one of which is MSGF+ (named *MSGF+ MSMS Search*).
2. Click on *MSGF+ MSMS Search*. The central pane will then display the input parameters for MSGF+.
3. Because the spectrum file that we just added to the History is in MGF format, we need to select MGF as the *Input Type* (the first parameter). This second parameter will change to *Input mgf* and you will then be able to select the `exampleSpectra.mgf` file from the dropdown list there.
4. To search against the protein sequence file in your History, change the *Database source* parameter to *Your Upload File* and select `human.fasta` from the *Uploaded FASTA file* dropdown list that appears below it.
5. Normally we would carefully set the other parameters according to the experimental methods used, but for the purposes of this tutorial we will leave them at their default values and start the search by clicking the *Execute* button below the parameters. This will start the MSGF+ search, which will be indicated by the addition of a new yellow History item called something like "MSGF+ MSMS Search on data 1 and data 2". This box will turn green when the search is complete – wait until this happens before continuing to the next step.

4.3 Perform post processing on the peptide spectrum matches (PSMs)

The MSGF+ output only contains basic PSM information. More work is needed to determine which of these PSMs are likely to be valid and which proteins they map to. To do this, we can use a GIO tool called *PIT:PSM PostProcessing*. This tools combines three individual tools from mzIdentML-lib to determine the false discovery rate (FDR) or each PSM, remove all PSMs below a user-defined threshold and map the remaining PSMs to proteins. To perform this post processing:

1. Expand the *GIO:PIT* tool section by clicking its name in the *Tools* pane, then click *PIT:PSM PostProcessing* to show the parameters for that tool.
2. For the *Input file* parameter, select the MSGF+ output file that you just created (this will be listed as something like "3: MSGF+ MSMS Search on data 1 and data 2".
3. Set the *CV term of score to use for FDR calculation* parameter to *MS-GF:SpecEValue*, so that the post processing tool knows which search engine the input file came from.
4. For the purposes of this tutorial you can leave the other parameters as they are (note that the *Threshold value* is FDR threshold) and click *Execute* to begin the post processing.

4.4 View the results

You can view the output of the post processing by clicking the relevant 👁 icon in the History pane, once processing is complete and the item has turned green. This output is in mzIdentML format, so not easily human readable. We can get a more digestible view by applying GIO's *PIT:Extract hits* tool as follows:

1. If it's not already expanded, click *GIO:PIT* in the Tools pane to expand that section, and click *PIT:Extract hits* to open the parameter window for this tool.
2. Set the *Input file type* parameter to *mzIdentML*, then select the mzIdentML file to process (it will be called something like "4: Post Processing on data 3").
3. Click the *Execute* button and wait until processing is complete (i.e. result file turns green).

Now if you click the 👁 icon next to the freshly created result file you should see, in the central pane, an easily readable table listing the identified peptides, the proteins they map to and some identification scores (unless you have a huge monitor you will have to scroll horizontally and vertically to see all this information).

5 Summary and next steps

You've completed your first proteomic data analysis within GIO. Although this was only a simple example with a small toy data set and default parameters, it demonstrates the ease with which data can be processed within GIO. You should find it easy to analyse your own proteomics data by following similar steps. (Note that MSGF+ accepts spectral data in MGF and mzML format – if you have data in raw or other formats you can convert it within GIO using the *MSConvert* tool from the *GIO:General Proteomics* section.)

Using individual tools via GIO's *Analyse Data* interface is a lot easier that working with the same tools using the command line, and because every step is recorded in the *History* it is very easy to double check analyses and share them with colleagues. It's worth spending a few minutes looking at the various sections in the *Tools* pane to see what else you can do in GIO.

However, the real power of GIO comes from *workflows* – analysis pipelines created by joining multiple tools together. A selection of ready-made workflows can be found by selecting *Published Workflows* from GIO's *Shared Data* menu. Among the workflows listed you will see one called *Example from introductory video*. This combines all the analysis steps from this tutorial into a single workflow, allowing the whole process to be executed by a single click. We'll show you how to do this in Tutorial 2.

Last updated 14 July 2014.