Comparing Skyline With Other Tools

By Dario Amodei with Brian Pratt

*[This document is based on an internal Skyline developer document written by Dario Amodei, and has been edited for general use by Brian Pratt. Descriptions of the various tools were current as of November 2014. – ed.]*

This tutorial explains the basics of how to use compare Skyline with these three tools:

* Peak View (from SCIEX)
* Spectronaut (from Biognosys)
* OpenSwath (from the Aebersold lab)

This is not intended to be an exhaustive tutorial in the use of these tools. The focus is on how to take a DIA experiment from a Skyline document and use these other tools to run exactly the same assay with the same raw mass spec files, and then re-import these results to Skyline for comparison. Installation of these tools is beyond the scope of this document, and it is assumed that the reader has access to the tools of interest.

This mainly involves getting transitions, spectral libraries, and retention time information from Skyline into the other tool, running that tool, and then getting picked peaks back from the tool to Skyline. Thus, this tutorial is most directly useful for comparing the peak picking of external tools to that of Skyline, but it may also be useful as a very basic guide to the general operation of these external tools.

Data for this tutorial can be found at  
[TODO URL]\ExternalSoftwareTools.zip

Table of Contents

[Peak View](#_Toc403469518)

[Exporting Data from Skyline](#_Toc403469519)

[Preparing the Exported Skyline Report for Peak View](#_Toc403469520)

[Importing To Peak View](#_Toc403469521)

[RT Calibration in Peak View (iRT-Based Workflows Only)](#_Toc403469522)

[Processing the Data in Peak View](#_Toc403469523)

[Exporting Data From Peak View](#_Toc403469524)

[Converting Data to Skyline Format](#_Toc403469525)

[Importing Data Back to Skyline](#_Toc403469526)

[Spectronaut](#_Toc403469527)

[Exporting Data from Skyline](#_Toc403469528)

[Preparing the Exported Skyline Report for Spectronaut](#_Toc403469529)

[Importing and Processing in Spectronaut](#_Toc403469530)

[Exporting Data from Spectronaut](#_Toc403469531)

[Converting Data Back to Skyline Format](#_Toc403469532)

[Importing Data Back to Skyline](#_Toc403469533)

[Customizing Spectronaut Settings](#_Toc403469534)

[Open Swath](#_Toc403469535)

[Exporting Data from Skyline](#_Toc403469536)

[Preparing the Exported Data for OpenSwath](#_Toc403469537)

[Running Open Swath](#_Toc403469538)

[Running mProphet on the Output Data](#_Toc403469539)

[Converting Data Back to Skyline Format](#_Toc403469540)

[Importing Data Back to Skyline](#_Toc403469541)

# Peak View

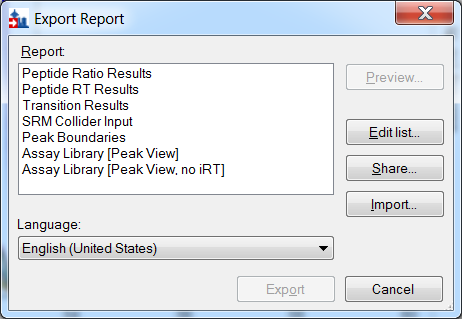
Peak View is SCIEX’s proprietary software, used to analyze .wiff files. Peak View requires a license to use. DIA data processing requires the “MS/MS(ALL) with SWATH Acquisition MicroApp” add-on from SCIEX. The instructions below will work with PeakView v2.1 and SWATH MicroApp v2.0; they may also work with earlier versions but this has not been tested.

*General Comments from Dario:* Peak View has roughly similar capabilities to Skyline in terms of analyzing DIA data, however it has the limitations that:  
(1) it can only be used to analyze SCIEX data  
(2) it uses a pre-trained scoring model for peak picking and does not provide flexibility in how the model is trained  
(3) I have had difficulty obtaining RT information from DDA runs directly as opposed to through assay libraries, though PeakView may support this in a way I don’t know about yet.   
Speed appears comparable to Skyline although I have not yet tested its memory limits on large (>100,000) numbers of transitions.

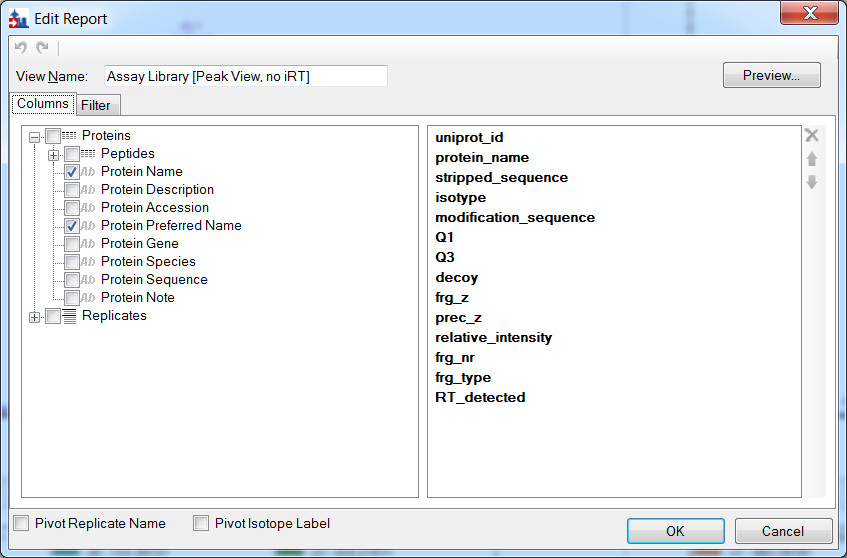
## Exporting Data from Skyline

To process data in Peak View you must provide both (1) raw mass spec data and (2) specification of the transitions to be extracted and their properties (e.g. spectral library, RT). The easiest way to provide (2) to Peak View is through a format known as an “ion library” (SCIEX terminology) or “assay library” (Aebersold lab terminology). Skyline can produce a PeakView-formatted assay library using a custom report which is included with the tutorial data set. To export the assay library from Skyline:

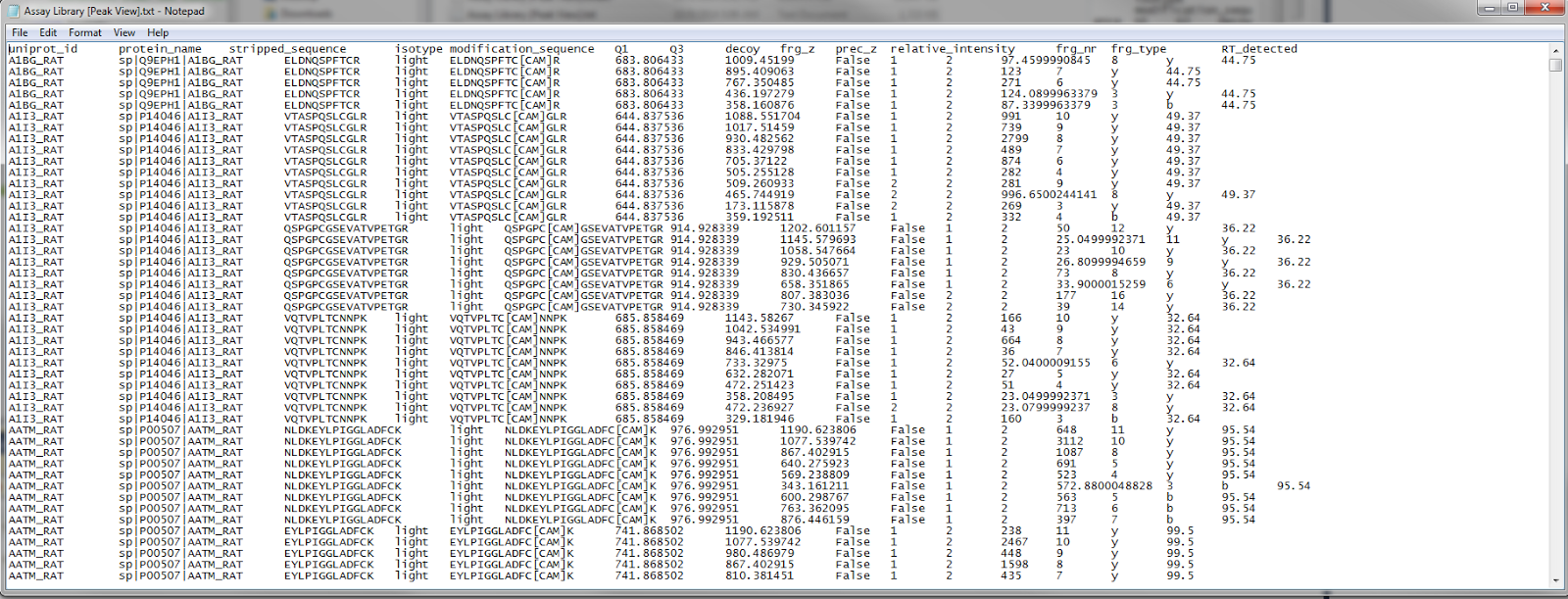
* Load a suitable DIA Skyline document like ExternalSoftwareTools\PeakView\HeldDIA.sky from the   
  provided ExternalSoftwareTools.zip  
  (note: associated WIFF files are at [TODO URL] /net/maccoss/vol1/home/brendanx/data/MacCoss/MProphet/20140609\_Held\_DIA\_processed )
* Click on **File -> Export -> Report**
* Click on the **Import** button and select the AssayLibraryPeakView.skyr file included in the tutorial data set.
* Click **OK** to import the custom reports contained in the .skyr file.
* You should see a series of new reports, including one titled Assay Library [Peak View, no iRT], as shown below.
* Click the **Export** button to export the report.
* Choose tab separated output.
* Choose a name for the report and click **OK.** Any name is OK but make sure to give it a **.txt** extension (e.g. ExportForPeakView.txt) as this is the only extension Peak View can read.



The report template you will export looks like this:



and the report itself, after some editing (see below), should look like this:



The columns generated by the report are the following:

|  |  |
| --- | --- |
| **Name** | **Meaning** |
| uniprot\_id | Abbreviated protein name |
| protein\_name | Full protein name |
| stripped\_sequence | Peptide sequence without modifications |
| isotype | Precursor isotope, e.g. “light” or “heavy” |
| modification\_sequence | Peptide sequence with modifications |
| Q1 | Precursor Mass |
| Q3 | Product Mass |
| frg\_z | Product charge |
| prec\_z | Precursor mass |
| relative\_intensity | Spectral library intensity of that transition |
| frg\_nr | Ion series number, e.g. the 8 in y8 |
| frg\_type | Type of fragment ion, e.g. b, y, precursor |
| iRT | Predicted iRT value for this transition |
| RT\_detected | Measured retention time value in the Skyline document |

## Preparing the Exported Skyline Report for Peak View

The custom report exported from Skyline can *almost* be directly read by Peak View, but there are a few changes in format that need to be made first. Currently these changes need to be made manually but hopefully can be automated in the future:

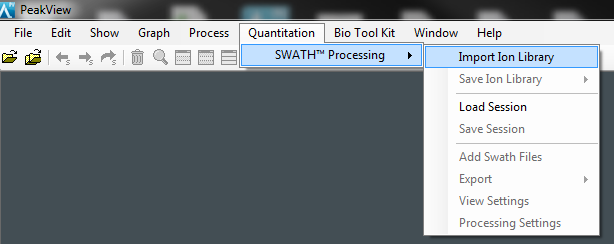
1. Format of modified peptide sequences in **modification\_sequence** column must be edited. Peak View can only read modifications in the “short-name” format, for example C[CAM] for carbamidomethylation of carbon. Skyline can *read* short-name format, but it currently always *outputs* in mass difference format, e.g. C[+57.0]. Thus, you must manually change the mass difference format to the short-name format in the **modification-sequence** column, before it can be read by Peak View. To find the short name for the modification, you can first find the long name in the Skyline settings (which should contain the mod), and then (if you have access to Skyline source code) you can look up the short name in the UniModData.cs file in the Skyline project, which can be found at <https://svn.code.sf.net/p/proteowizard/code/trunk/pwiz/pwiz_tools/Skyline/Model/DocSettings/UniModData.cs>.
2. Remove “#N/A” in **relative\_intensity** column. When a particular transition does not have a corresponding intensity in the spectral library, Skyline will output a #N/A value in its report. Peak View cannot read #N/A and these values should instead be set to 0. Other columns will occasionally also contain #N/A, and need to be changed to an appropriate default value. Sometimes, when your experiment lacks a certain type of data (e.g. no spectral libraries), the whole column is #N/A. In that case, you can delete the whole column and Peak View will still be able to read the file.
3. Adjust **iRT** and/or **RT\_detected** depending on the setup. There are two ways of using retention time information in Skyline: there may be iRT values that are used to predict the retention time, or the RT may be inferred approximately from accompanying DDA runs. The Skyline report exports *both* formats (in the **iRT** and **RT\_detected** columns, respectively), but you only need one and in fact Skyline will get confused if both are present. If you have iRT values, then you should set every entry in the **RT\_detected** column to -1. If you do not have iRT values and wish to use the detected RT times instead, then simply delete the **iRT** column or use the “no iRT” version of the report as provided. An example of each format is included with this tutorial.
4. The file must be tab-separated, not comma separated. If this isn’t already the case, you can simply replace commas with tabs in a text editor.

If, after making these changes, Peak View still cannot read the file, the problem is usually some format or value-type error (e.g. Peak View doesn’t like a particular modification notation). Unfortunately, Peak View gives rather non-specific errors (usually it gives a generic format error with no indication of which line it occurred on), so tracking down the source of the problem can be difficult. You may need to import the file part by part to isolate the error.

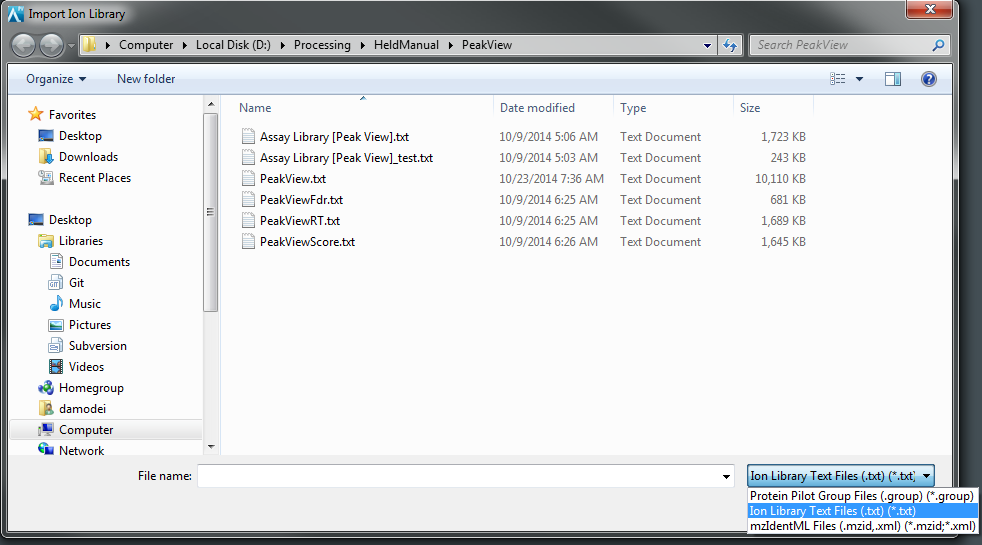
## Importing To Peak View

The Skyline report should now be in a format that Peak View can read as an assay library/ ion library. To import to Peak View:

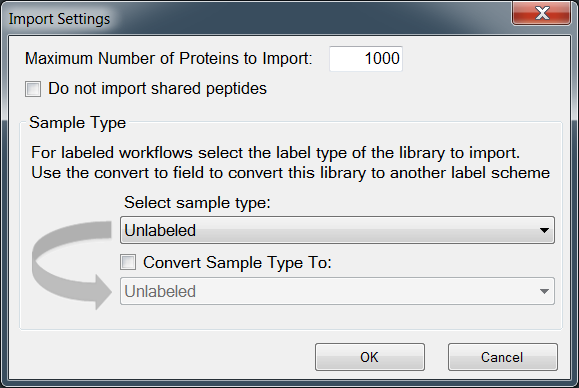
* Open Peak View. Make sure you have the MS/MS(All) SWATH Micro App installed.
* Under the **Quantitation -> SWATH Processing** menu, click on **Import Ion Library.**



* A menu should show up allowing you to choose a file, as shown below. Select **Ion Library Text Files** in the bottom right, then select the Skyline report you exported/modified, and click **Open**

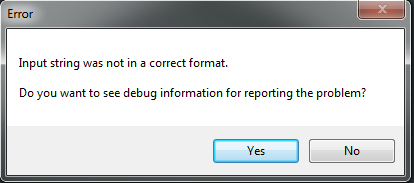


Next you should see the following menu:



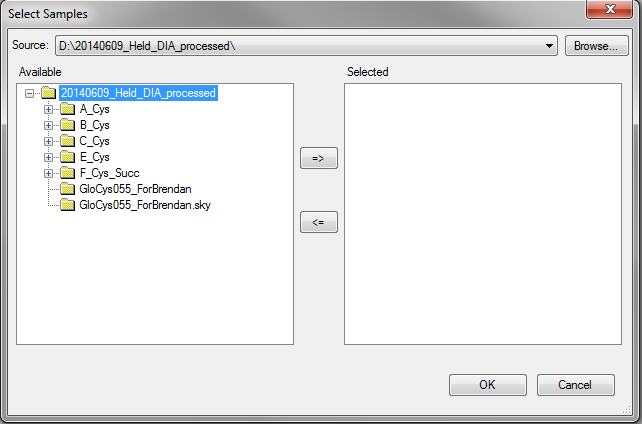
* Enter **1000** for maximum number of proteins to import (or a higher number, if your sample has more proteins). It is not important exactly what number you enter, it just has to be larger than the number of proteins in your sample.
* Uncheck **Do not import shared peptides**
* If your Skyline document contains all unlabeled peptides, select **Unlabeled** for sample type. If it contains heavy/light pairs or another labeling scheme, select the appropriate labeling scheme. The data set we’re using here is Unlabeled.
* If your Skyline document has a labeling scheme, but you want Peak View to process each precursor separately, pretending that all the isotopes (e.g. heavy and light) of a given peptide are simply separate peptides, then check **Convert Sample Type To** and make sure **Unlabeled** is selected (this workflow can be useful for certain types of software comparisons). Otherwise, don’t check this box.
* Click **OK**

If there is a problem in your ion library / assay library, you will get an error message, usually this one:



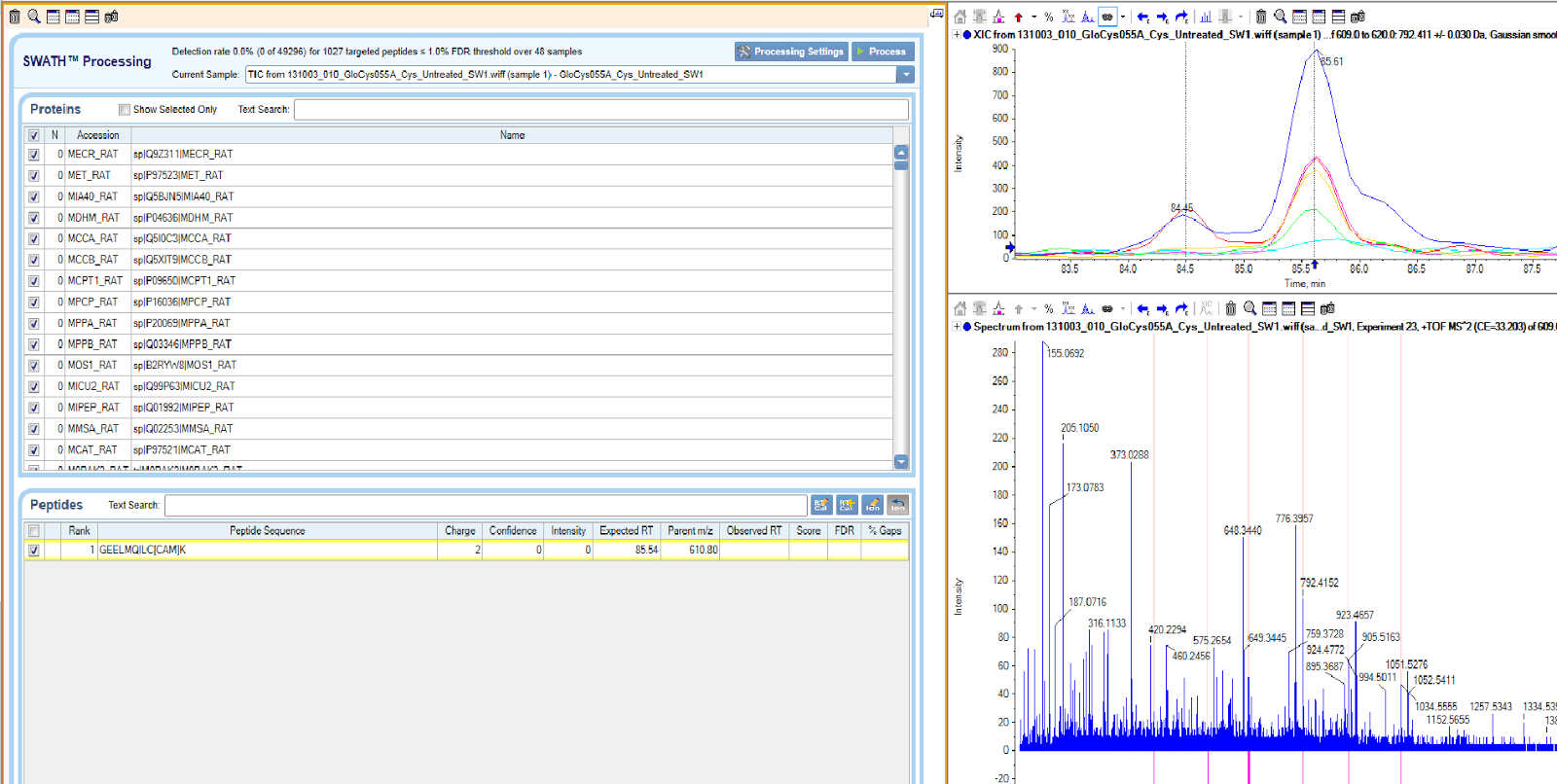
Typically it indicates a format problem (e.g. modified sequence is unrecognized, or #N/A occurs in a column). You may also get another error message saying the required columns were not recognized, this usually means you’re using tabs instead of spaces, some column names are incorrect or missing, or the tab-formatting is incorrect. These errors can be hard to debug and it is often necessary to rely on trial and error (or ask SCIEX for help).

In any case, once you’ve solved any errors in your ion library, you should get something like the following screen:



* Select any files and folders containing the mass spec (.wiff) files you want to analyze. You can get a suitable set from [TODO URL] /net/maccoss/vol1/home/brendanx/data/MacCoss/MProphet/20140609\_Held\_DIA\_processed/C\_Cys\_subset.zip
* Click on the **=>** button to add these files to the experiment, or use the **Browse** button to find more files
* Click **OK**

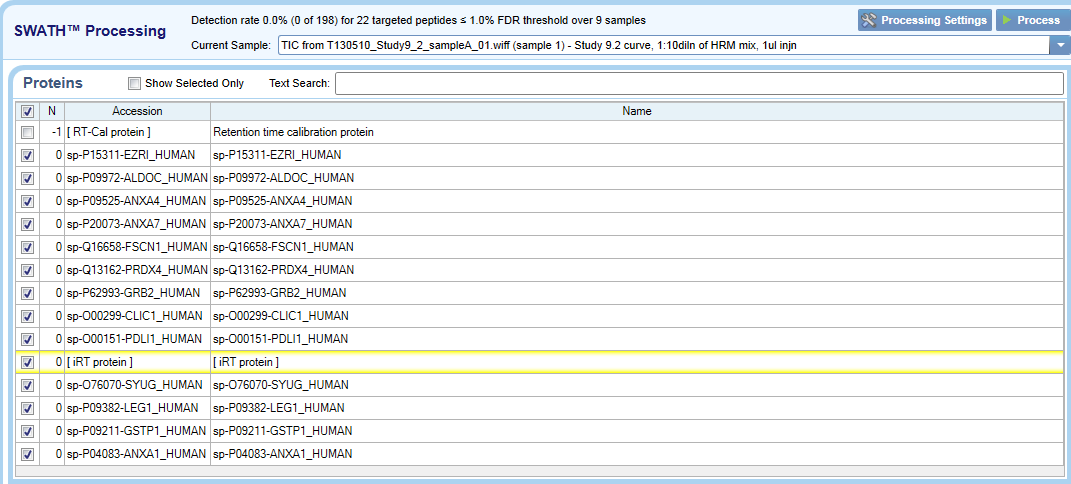
After a few minutes, the transitions in your assay library should be extracted for the files you indicated. You’ll then get several Peak View panels showing a list of transition, chromatograms, and spectra, roughly like this:



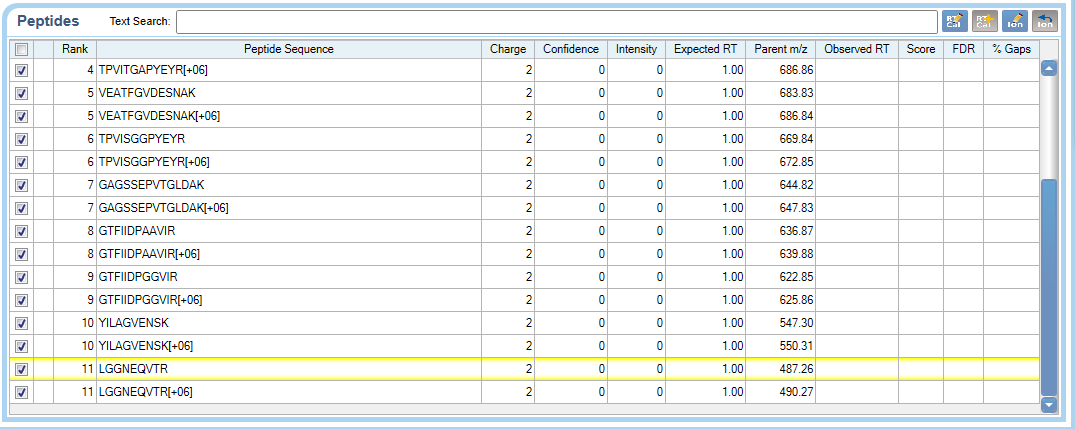
## RT Calibration in Peak View (iRT-Based Workflows Only)

If you chose to import iRT values rather than detected RT times (see point 3 under “Preparing the Exported Skyline Report for Peak View”), then you need to calibrate those iRT values so that Peak View can convert them into predicted retention times. (If you used detected RT times, no calibration is necessary and you can skip this section). Your Skyline document should have contained retention time standards, and these retention time standards should have been imported into Peak View and should now be visible as peptides. To calibrate your iRT values based on the RT standards, follow these steps:

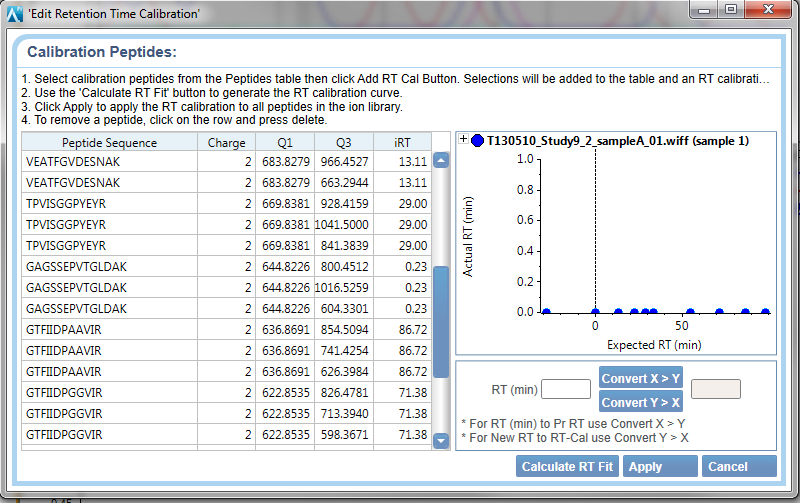
* In the list of proteins, click on the protein (or, depending on how you set up your assay library, proteins), corresponding to the RT standards:



* In the list of peptides for the RT protein, click on one of the peptides (make sure it is a non-heavy peptide if you are using a labeled workflow) and then click the **RT-Cal** button with a plus next to it (not the one with a pencil next to it).
* Select another peptide, click the **RT-Cal** (with plus) button again, and repeat this process for each of the calibration peptides. (Note: it does *not* work to select all the peptides and then click the **RT-Cal** button – you need to do it one at a time).



* Click on the other **RT-Cal** button (the one with the pencil) – you should see the following screen:



* Click on **Calculate RT Fit**. Peak View will conduct an RT calibration and this may take up to 30 minutes in some cases.
* When the calibration is done, inspect the regression graph on the right. If there are any severe outliers, consider removing them and re-running the calibration. Otherwise, click **Apply.**

The iRT predictions should now be calibrated; you can move on to processing the data.

## Processing the Data in Peak View

After RT calibration (if necessary), the next step is to process the data, which includes extracting transitions from the .wiff files and picking the best peak. This could take significant time depending on the number of files in the sample.

If the **Process** button is greyed out, it’s because you don’t have any peptides selected. Try this:

* Under the **Quantitation -> SWATH Processing** menu, click on **Processing Settings.**
* Check the **Fix Rank** box, then click **OK**.
* The **Processing** button should now be active.

Now you can proceed:

* To process the data, click on the **Process** button
* A save file dialog will open. Choose a name for the Peak View document which will be created from the processing, and click **Save.**
* Results and picked peaks will be available for manual browsing in the chromatogram pane on the right.

Use the protein and peptide boxes to select which protein and peptide you are viewing, and the **Current Sample** drop down menu to select the file. This allows you to browse through all the chromatograms as you would in Skyline. You can also see the spectrum at a given point by clicking on it in the chromatogram window.

## Exporting Data From Peak View

The next step is to export the results of the peak picking so that they can be compared to Skyline’s peak picking. There are three pieces of information to be exported from each chromatogram: the retention time of the picked peak, the score of the peak, and the q-value associated to that score. Because of the way Peak View works, you must export this information in 3 separate files:

* Click on **Quantitation -> SWATH Processing -> Export -> Score**
* In the Save File dialog, choose the name “PeakViewScore.txt” and click **Save**
* Click on **Quantitation -> SWATH Processing -> Export -> FDR**
* In the Save File dialog, choose the name “PeakViewFdr.txt” and click **Save**
* Click on **Quantitation -> SWATH Processing -> Export -> Observed Retention Time**
* In the Save File dialog, choose the name “PeakViewRt.txt” and click **Save**

This exports all the relevant data, although it requires another format conversion before it can be imported back to Skyline. This is covered in the next section.

## Converting Data to Skyline Format

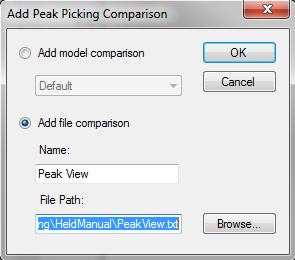
You can download a program “PeakViewConvert.exe” ([TODO URL]) that combines the score, FDR and retention time files into a single file for importing to Skyline. PeakViewConvert is a command line utility, just run it without any arguments for instructions.

The file output by this conversion is usually suitable for import to Skyline as is, but sometimes manual correction is required. In particular, Peak View outputs its own decoys along with the processed peptides, and these decoys sometimes fail to conform to the settings of the Skyline document. For example, the peptide T[Acetyl]NICVT might be in the original assay, with the setting that Acetyl must be N-terminal, and Peak View might generate a decoy NIT[Acetyl]CVT, which violates this rule. The new peptide will then give an error when imported by Skyline. Removing the [Acetyl] in the decoy will solve the problem; Skyline will then ignore the decoy instead of throwing an error.

## Importing Data Back to Skyline

To import the peak scores, q-values, and retention times back into Skyline, you can use the PeakScoringComparison framework, as described in the tutorial for that feature. Open the original Skyline document (the one used to export the initial report). Then:

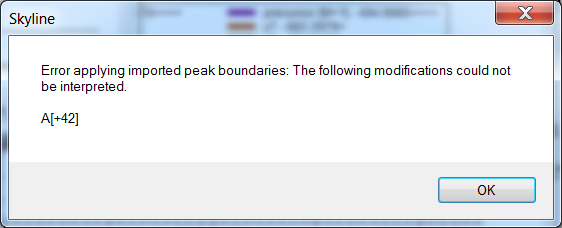
* Click on **Edit -> Refine -> Compare Peak Scoring**
* Click on **Add**; you should see this dialog:



* Select **Add File Comparison**
* Enter a name (e.g. “Peak View”)
* Click on **Browse**, and load the file that you created in the last section

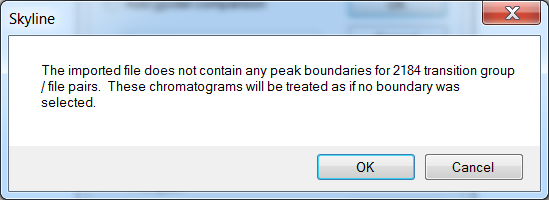
Click **OK**, and wait for Skyline to evaluate the performance of Peak View against the manually curated Peaks. You can also compare against Skyline’s automated Peak Picking, or against other tools as well.

You may encounter something like this:



This is a result of PeakView’s decoy behavior as described [above](#_Converting_Data_to). You’ll just have to edit the file and do the import again (in this case, just change all “[+42]” to “”).

You may also encounter something like this:



And a couple of other warning dialogs as well – all basically saying that there are things in the import that aren’t in the Skyline document. This is normal.

Alternatively, you can simply import the peak boundaries directly into Skyline for viewing and visualization. To do this, click on **File -> Import -> Peak Boundaries**, and open the file that you created in the last section. The peak boundaries determined by peak view should then be imported directly into Skyline. You can then inspect the peak boundaries directly through the chromatogram view.

# Spectronaut

Spectronaut is software produced by the Swiss company Biognosys and is specially designed to process and analyze DIA data. Relative to Skyline, which is very broad in its capabilities, Spectronaut focuses narrowly on processing high throughput DIA data. Like Skyline and Peak View, it accepts transitions, retention time, and spectral library information through an assay-library like format, and then separately imports mass spec files for which to extract those transitions.

*General Comments from Dario*: Spectronaut really does only one thing - extraction and statistical analysis of DIA transitions. All samples must contain the Biognosys HRM calibration kit; the software simply will not process the sample without it. This limits the applicability of Spectronaut on data acquired by others and not acquired with HRM.

## Load Some Data into Skyline

* Load some suitable data like ExternalSoftwareTools\Spectronaut\ReiterSPRG.sky from ExternalSoftwareTools.zip  
  (note: associated RAW files are at [TODO URL] /net/maccoss/vol1/home/brendanx/data/MacCoss/MProphet/20131031\_Reiter\_ABRF\_sPRG\_study )

## Exporting Data from Skyline

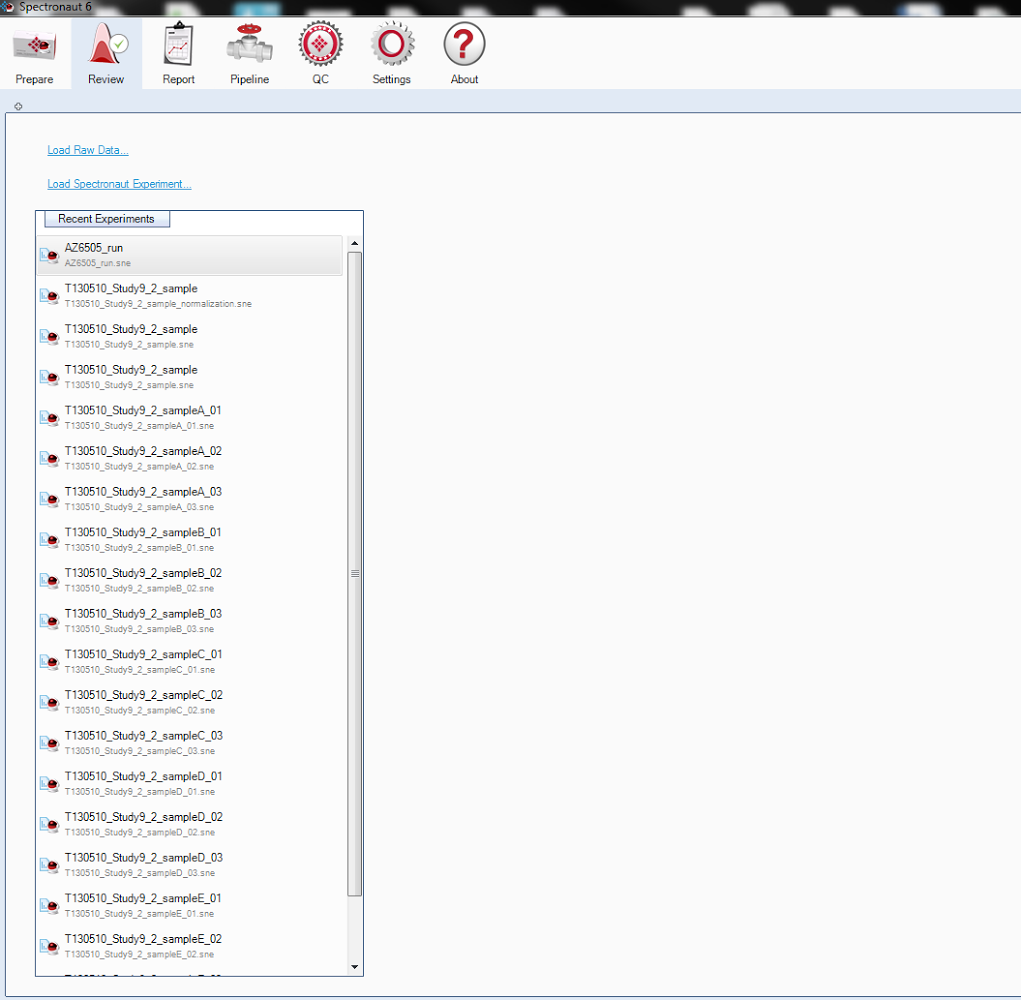
Exporting data from Skyline for use in Spectronaut can be done in exactly the same manner as exporting for use in Peak View. Spectronaut is highly flexible about the names and format of columns, so you can even use the same skyline report and column titles as you used for Peak View. See the subsection entitled “[Exporting Data from Skyline](#_Exporting_Data_from_1)” under the Peak View section in this document.

## Preparing the Exported Skyline Report for Spectronaut

Because Spectronaut is so flexible with input, you shouldn’t need to do much to prepare the data for import. Any “#N/A” entries should be replaced with an appropriate default (e.g. 0 in the case of spectral libraries), and iRT standard peptides should be removed (Spectronaut does not need to be given calibration peptides directly and instead simply assumes the HRM kit is in the sample).

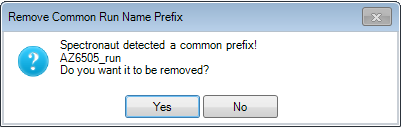
## Importing and Processing in Spectronaut

You are now ready to import your assay library to Spectronaut. Open up Spectronaut; you should see a screen like this:



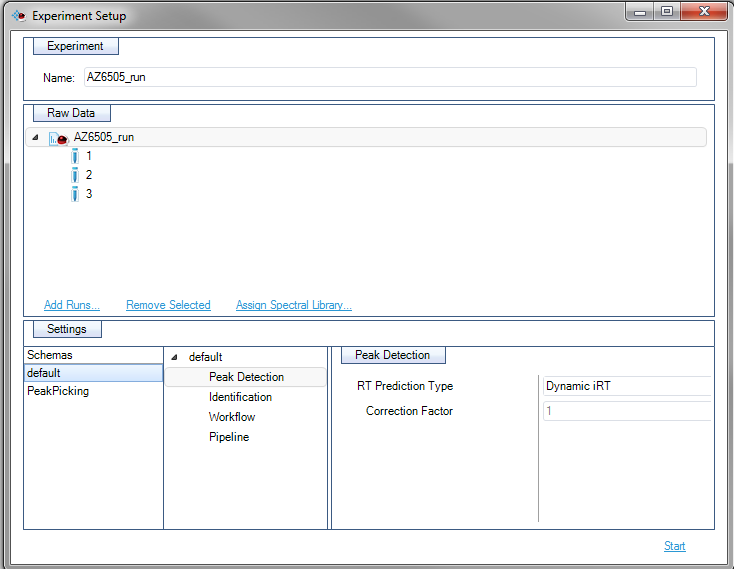
What you see you may vary slightly depending on what if any experiments you’ve done recently, but the screenshot above is basically what you should see. To import your assay library:

* Click on the **Review** button on the top panel.
* Click on the **Load Raw Data** text link near the upper left corner
* A file open menu should appear; select all the mass spec files you want to import, in the case of this tutorial, the 3 .raw files that were in our Skyline document (AZ6505\_run1.raw, AZ6505\_run2.raw, AZ6505\_run3.raw).
* Click **Open** to add these files
* You should see the following screen, asking if you want to remove common prefixes:

****

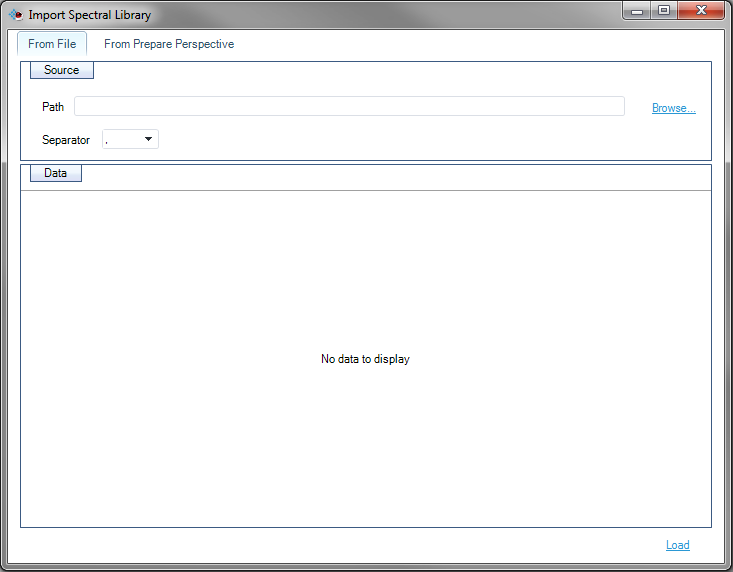
* Click **No** to retain the current prefix

You should now see the following screen, which allows you to set up the assay library you will use to extract transitions from these files, as well as to choose various options for the data processing.

****

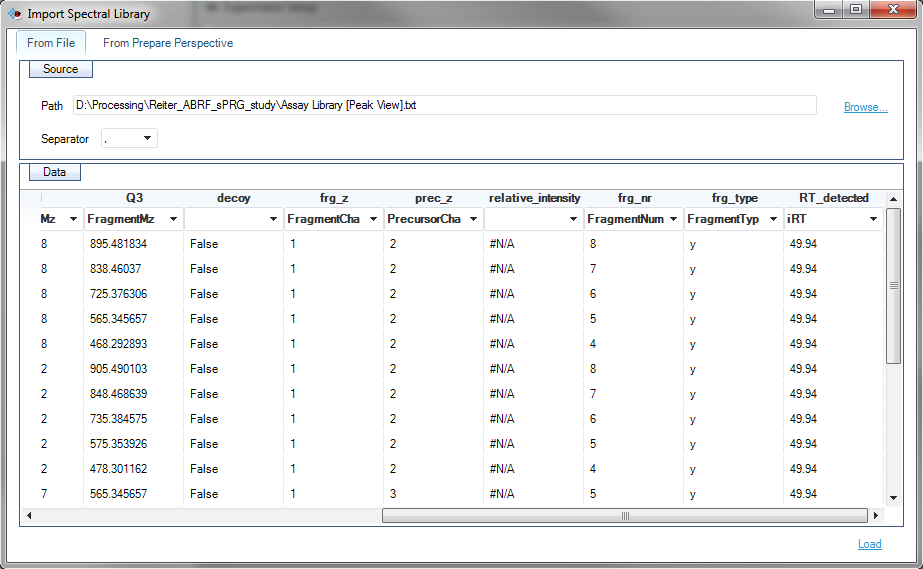
We’ll start by assigning the spectral library:

* Make sure the top-level run folder “AZ6505\_run” is selected in the **Raw Data** pane, rather than just one of the three files under it
* Click the **Assign Spectral Library** text link in the **Raw Data** pane
* You should see the following screen:



* Click **Browse** in the top right corner to import an assay library
* An open file dialog box should appear. Select the assay library you want to import (in our case, the file “Assay Library [Peak View].txt” that you’ve generated from Skyline) and click **Open**

The first few lines of the file should now appear in the **Import Spectral Library** dialog, as shown below:



As you can see near the middle of the screenshot, there are two rows of column titles shown. The upper row (the one containing “Q3”, “decoy”, “frg\_z”, and so on) are the names that you’ve given to the column in the text file itself. The lower row (the one containing “FragmentMz”, “FragmentCharge”, and so on) are the data fields Spectronaut is looking for (e.g. the column entitled “FragmentMz” should be matched to a list of product Mz values).

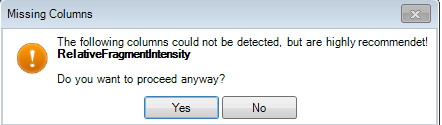
Spectronaut has attempted to guess which columns are which, and tends to be mostly correct, but you will often need to correct one or more errors. To manually override Spectronaut’s column matching, you can click on one of the lower row of column names (e.g. “FragmentMz” and choose from a dropdown list of possible names to assign instead). A column title can also be blank to indicate that Spectronaut should not use it at all.

Note how the user has set RT\_detected to mean iRT in Spectronaut in the figure above. Do that! Also make sure the modification sequence is mapped.

How many columns you have to adjust will depend on what you’ve imported before – Spectronaut remembers column names that it’s seen before, and so its matching gets better over time. My copy of Spectronaut gets every column correct, but since the “relative\_intensity” column contains mostly “#N/A” (there are no spectral libraries for many of the transitions), I’ve told Spectronaut to ignore that column entirely by setting the title of that column to a blank.

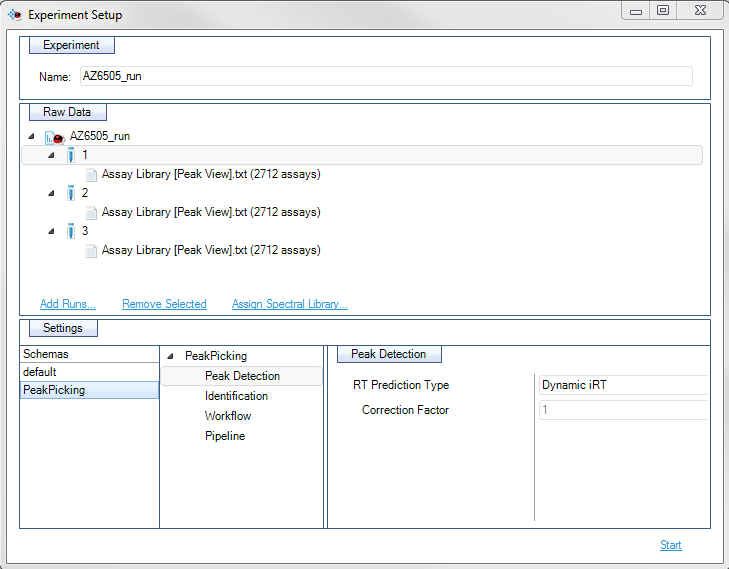
When you are done matching columns to names, you are ready to execute the import:

* Click the **Load** text link in the bottom right
* If any important columns are missing, you should see a dialog asking you to confirm proceeding without them, as shown below. In our case, the “RelativeFragmentIntensity” column is missing because we don’t have spectral library information for many peptides; this is OK and we’ll proceed without it.



* Click **Yes** to proceed

You should now be returned to the **Experiment Setup** menu, which should now look like this:



Check that every run in the **Raw Data** pane has a set of assays attached to it, as shown above. If only some runs have assays, you need to repeat the **Assign Spectral Library** steps, making sure that the directory “AZ6506\_run”, rather than just one of the files, has been selected.

Next we need to adjust the **Settings** pane. This pane determines the settings under which Spectronaut will process the data, and can make a big difference in the type and quality of data processing. I recommend the following settings:

* Under **Identification,** make sure **Dynamic Score Refinement** is checked
* Under **Identification,** make sure **Include MS1 scoring** is checked
* Under **Workflow,** make sure **Interference Correction** is checked
* Under **Workflow,** check the **Cross Run Normalization** checkbox

There is also a list of **Schemas** at the leftmost part of the **Settings** pane. Spectronaut has a set of default settings represented by the **default** schema; you ordinarily have to change these every time in order to get the settings you want. However, it’s possible to define and save a new schema (I’ve defined one called “PeakPicking” as shown above) that has exactly the settings you want. Defining new schemas is covered in the “Customizing Spectronaut Settings” subsection below. In this particular workflow, we won’t define a schema.

You’re now ready to begin data processing:

* Click the **Start** text link in the lower right
* You should return to the main Spectronaut screen and a progress bar should appear in the lower left showing the progress of the data processing:

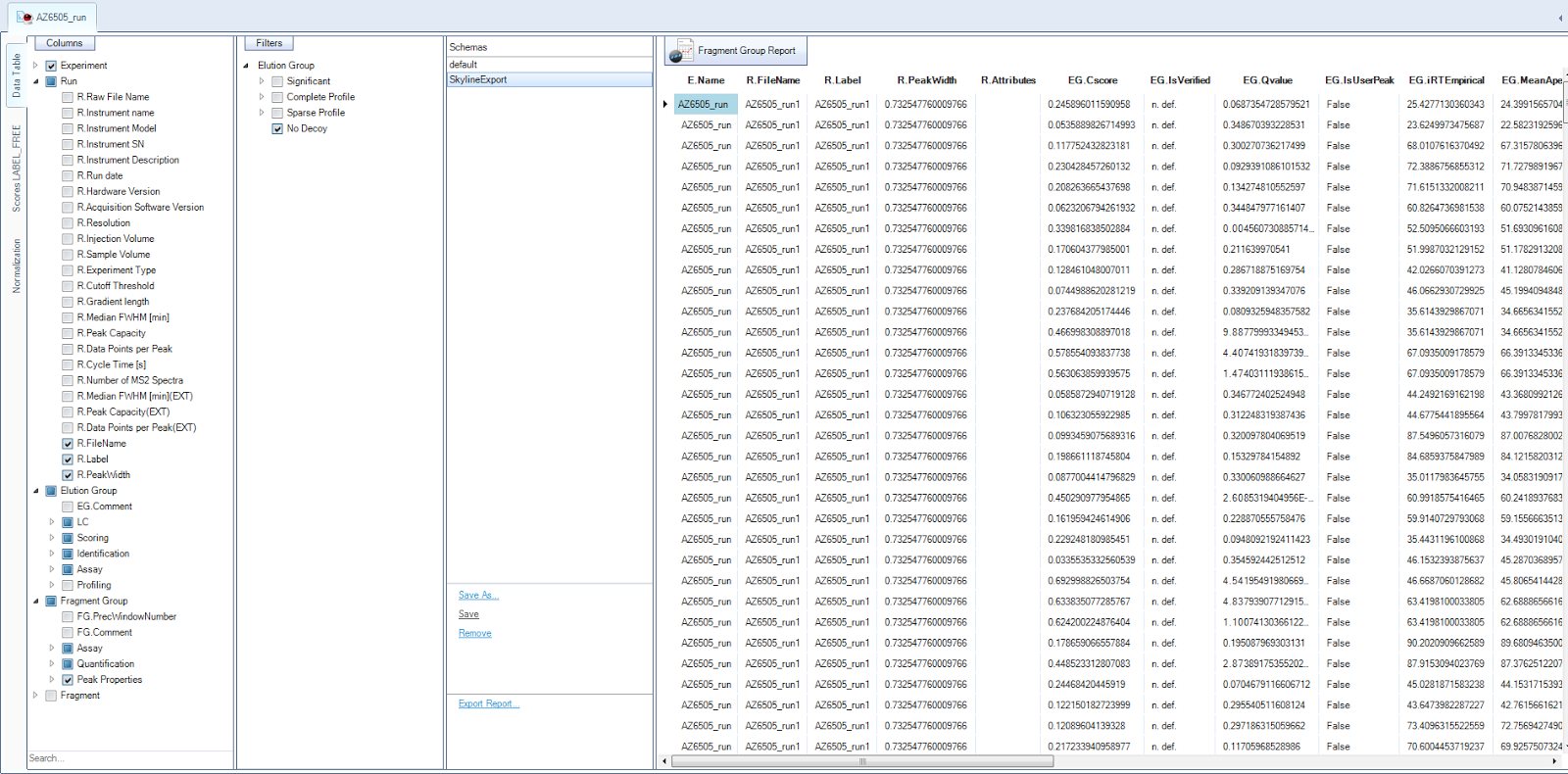


For the 3 files and ~3000 transitions in the example files, Spectronaut should take 5-15 minutes to process the data.

## Exporting Data from Spectronaut

After Spectronaut finishes processing the data, the next step is to examine the results and export them in tabular format. For the purposes of this tutorial we are interested in the peak positions and q-values produced by Spectronaut. To see the results:

* Click on the **Report** button on the top of the Spectronaut window
* You should see the following screen:



The first few lines of results are shown on the right side of the screen, along with a bunch of options to display various columns, filter the results, or export the results. You will want to make a report suitable for reading by Skyline, which can be done as follows:

* Under the **Filter** pane, check **No Decoy** to avoid exporting Spectronaut’s decoys
* Under the **Columns** pane, check the following boxes:
  + **Run -> R.FileName**
  + **Elution Group -> LC -> MeanApexRT**
  + **Elution Group -> Scoring -> EG.CScore**
  + **Elution Group -> Identification -> EG.QValue**
  + **Elution Group -> Assay -> EG.Modified Sequence**

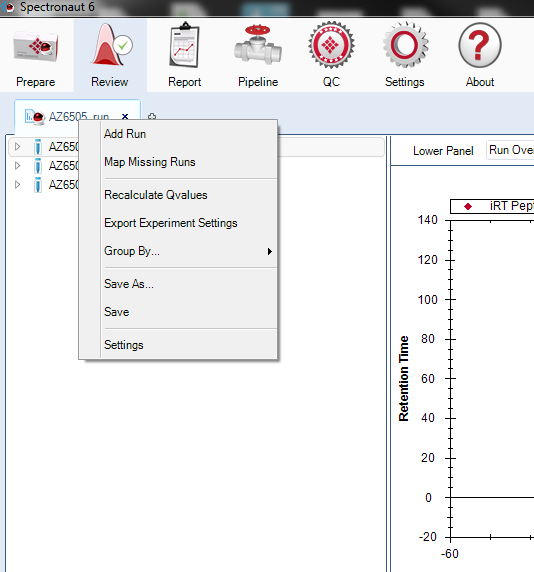
This will cause the file name, chosen peak apex, modified sequence, precursor charge, and Spectronaut’s q-value to appear as columns in the exported table. Now we are ready to export the table:

* Click the **Export Report** text link in the bottom middle of the screen.
* A save file dialog should show up, choose a file name (e.g. “PeakViewExport”) and make sure to choose .tsv or .csv, not .xls, as the extension.
* Click the **Save** button

Several files will be written out by Spectronaut to the directory you specified. For purposes of this tutorial and for peak picking comparisons, we are only interested in the .tsv file.

You are now done processing and exporting the data in Spectronaut. You can save the Spectronaut results as a Spectronaut file (somewhat analogous to a Skyline document) using the following steps:

* Click on the **Review** button
* Right click on the tab near the top left showing the name of the experiment (in this case “AZ6505\_run”). You should see something like this:



* Click the **Save As…** menu item
* A save file dialog should appear. Choose a name for your Spectronaut experiment and click the **Save** button.

You can now safely close your Spectronaut experiment/document and re-open it at any later time.

## Converting Data Back to Skyline Format

The file exported by Spectronaut is very close to a format that Skyline can read, but a couple manual changes are required:

* Open the .tsv file that you wrote in the last section
* Change the name of the “EG.QValue” column to “annotation\_QValue”
* Change the name of the “EG.MeanApexRT” column to “Apex”
* Change the name of the “EG.CScore” column to “annotation\_Score”

With these changes, Skyline should be able to read the file exported by Spectronaut.

## Importing Data Back to Skyline

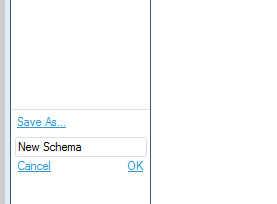
After performing the conversion above, you should be able to import your data back to Skyline in the same manner as described in the “[Importing Data Back to Skyline](#_Importing_Data_Back)” subsection of the “Peak View” section. You can either compare Spectronaut’s chosen boundaries to manually curated ground truth, or you can directly import Spectronaut’s chosen boundaries in order to visualize/inspect them in Skyline. As with PeakView, you will probably encounter a warning or two about things in the input file that aren’t in the Skyline document, you can just click through those.

## Customizing Spectronaut Settings

In the above instructions you need to change both the settings from processing data, and the report format for exporting data from Spectronaut. These changes to the default settings do not automatically persist across documents; if you want them to persist, you need to create named schemas and then choose these schemas every time you create a new Spectronaut experiment.

To create a schema for processing settings:

* Click on the **Settings** button on the top panel in the main Spectronaut window
* Select the “default” schema and make the settings changes you want to re-use
* In the botton left of the screen, click the **Save As** button, enter the name of a schema, and click **OK,** as shown here:



The new processing settings schema will then become available.

To create a new report schema:

* Click on the **Report** button on the top panel in the main Spectronaut window
* Select the “default” schema and make the report changes you want to re-use (e.g. add the columns and filters indicated in the tutorial above)
* In the botton middle of the screen, click the **Save As** button, enter the name of a schema, and click **OK,** as shown here:

The new schema will then appear as one of the available schemas and will include the columns and filters you’ve added.

The new schemas will remain available the next time you open a fresh Spectronaut experiment.

# Open Swath

Open Swath is a command-line based tool for picking peaks in DIA data. It was the first such tool for DIA analysis created (before Peak View, Spectronaut, or Skyline’s advanced DIA peak picking). It was developed in the Aebersold lab, and is used internally by that lab to process most of their DIA data. It is algorithmically innovative but difficult to use as software, and still under development. To install and use it I have found it useful to take advice and direction from its creator, Hannes Rost. Open Swath is integrated into a widely-used C++ proteomics suite called OpenMS. The latest version of OpenMS (as of writing) is OpenMS 1.11, however this does not contain the latest version of Open Swath. Instead I recommend checking out the latest source code version of OpenMS and building from that, using any patches as necessary (ask Hannes about this).

The workflow for running OpenSwath can be greatly simplified if you use the latest version; if you use the OpenMS 1.11 you’ll need to use additional steps not described in this document. Regardless of which workflow you use, it may be useful to also refer to the OpenSwath user guide, which can be found here (<http://www.openswath.org/example_OpenSWATH.zip>).

OpenSwath must be run in combination with mProphet, another command-line tool produced by the Aebersold lab. The instructions below describe how to export data from Skyline for use by OpenSwath, how to convert the results to mProphet format, how to run mProphet, and finally how to convert the output of mProphet back to a format that can be imported by Skyline and compared to Skyline’s performance.

In general, OpenSwath takes about an hour per .mzML file to run. mProphet usually takes a few minutes when concatenated on all the files. OpenSwath can technically run on Windows, but runs much better on Linux (I have never successfully gotten it working on Windows). I have found it easiest to run OpenSwath on the proteome.gs.washington.edu cluster – this allows several jobs to run in parallel. I generally convert Skyline output to OpenSwath format, then convert back to Skyline format often running through OpenSwath and mProphet

## Exporting Data from Skyline

Exporting (transition, retention time, and spectral library) data from Skyline to be used in OpenSwath is similar to the process of exporting it for use in PeakView (see “Exporting Data from Skyline” under the PeakView section). You can mostly follow the instructions in that section. However, instead of using the report “AssayLibrary [Peak View]”, you should instead use the report “AssayLibrary [OpenSwath]” (another custom report which I’ve included with this tutorial). Also, you should ensure that the exported file is comma-separated, not tab-separated. In addition, OpenSwath does not handle extraneous columns well, and the columns need to be in a specific order. All of these conditions should be taken care of by the report format. For reference, the names of the necessary columns are shown below. You can compare this to the PeakView table shown in the “Exporting Data from Skyline” subsection of the “Peak View” section above.

|  |  |
| --- | --- |
| **Name** | **Meaning** |
| ProteinName | Full protein name |
| ModifiedSequence | Peptide sequence with modifications |
| PeptideSequence | Peptide sequence without modifications |
| PrecursorCharge | Precursor charge |
| PrecursorMz | Mass to charge of precursor ion |
| Tr\_recalibrated | Irt value |
| LibraryIntensity | Spectral library intensity of that transition |
| Decoy | Is this transition a decoy? |
| FragmentType | Type of fragment ion, e.g. b, y, precursor |
| FragmentNumber | Ion series number, e.g. the 8 in y8 |
| FragmentCharge | Charge on the fragment ion |
| ProductMz | Mass to charge of fragment ion |
| labelType | Type of label on this transition |
| RawRT | Raw retention time where the peak was detected |

## Preparing the Exported Data for OpenSwath

As with PeakView, the exported Skyline report needs to be modified slightly before it is in a format to be accepted by OpenSwath. The column names and most of the entries should be correct, but you will need to make the following changes:

1. You will need to create another column called “transition\_group\_id”. This column needs to contain a unique ID string for every precursor. Different transitions of the same precursor must have the same ID string, and different precursors must have a different ID string. I have been generating this string using the unmodified peptide sequence + PrecursorMz + labelType.
2. As with PeakView, you will need to handle columns with #N/A. For spectral libraries you can replace #N/A with 0, for other columns you can use an appropriate default.
3. You will need to modify the columns Tr\_recalibrated and RawRT depending on whether or not the experiment has iRT’s. If it does, delete the RawRT column. If it does not, delete the Tr\_recalibrated column, and rename the RawRT column as Tr\_recalibrated.
4. OpenSwath can only read UniMod format names (e.g. (UniMod:16)), for modifications, whereas Skyline outputs modifications in mass format. You’ll need to do this replacement manually, as Skyline cannot yet output UniMod format modifications (it can read these modifications, but can’t output them). The Skyline source file UniModData.cs contains a lookup for converting names of modifications to UniMod format.
5. If your file contains iRT peptides, you will need to move these peptides to a separate file, since OpenSwath can only accept input with the iRT peptides in a separate file from the peptides to be measured. Simply copy the title row, and the rows containing the iRT peptides, to a new file. If there are no iRT peptides, simply skip this step.

## Running Open Swath

Before you do anything else, you should convert your raw mass spec files (whether they are .raw, .wiff, or some other vendor format) to .mzML. The latest version of OpenSwath (the specially installed, patched version, not the standard OpenMS1.11) can read .mzML. The standard (older) version of OpenSwath can read .mzXML, but involves a lot of manual processing that you can avoid with the newer version (the .mzXML files have to be split up into 32 parts and manually processed – I do not recommend this workflow and suggest avoiding the .mzXML format for OpenSwath)

OpenSwath only accepts transitions, spectral library information, and retention time information in TraML format. OpenSwath provides a standalone command line tool to convert .csv files to .TraML, which can be run as follows:

ConvertTSVToTraML -in AssayLibrary.csv -out AssayLibrary.TraML

Where “AssayLibrary.csv” is the Skyline report you exported, and “AssayLibrary.TraML” is the TraML file to be produced. You should also convert the list of iRT peptides to .TraML format (point 5 of the last section describes how to make a separate file with just the iRT peptides):

ConvertTSVToTraML -in AssayLibraryIrt.csv -out AssayLibraryIrt.TraML

OpenSwath requires a set of decoy peptides in order for its peak picking algorithm to run, and has its own utility for generating these decoys. To run this utility, use the following command:

OpenSwathDecoyGenerator -in AssayLibrary.TraML -out AssayLibraryDecoys.TraML -method shuffle -append -exclude\_similar

Where “AssayLibrary.TraML” is the assay library in .TraML format that you just produced above. This command will output a file “AssayLibraryDecoys.TraML” that contains the target transitions with appropriate decoys appended. Several methods are available for producing the decoys, and are described in the OpenSwath documentation. The method I have found to work the best is “shuffle”, as shown in the command above. “-append –exclude\_similar” means to append the decoys to the original target list (making a combined list that OpenSwath can process) and to remove any decoys that are too similar to the targets (this is usually the option you want).

Now you are ready to run OpenSwath itself. OpenSwath is run with the following command:

OpenSwathWorkflow -in /path/to/file.mzml –tr ./AssayLibraryDecoys.TraML -tr\_irt ./AssayLibraryIrt.TraML -out\_tsv ./OutputFile.tsv -readOptions cache -tempDirectory /dir/for/temp/files -swath\_windows\_file ./swath\_windows.txt -debug 10

Here “file.mzml” is the raw mass spec data to be processed, “AssayLibraryDecoys.TraML” is the decoy-appended .TraML file that was just produced above, “AssayLibraryIrt.TraML” is the set of iRT peptides that you converted to TraML format, “swath\_windows.txt” is the file specifying the pattern of SWATH isolation windows, and “OutputFile.tsv” is the file where OpenSwath should output its picked peak results.

Note: if your experiment did not contain any iRT peptides, then you need to leave out the –tr\_irt option, and probably also invoke some additional options, but this is outside the scope of this document.

Because you’ll want to run OpenSwath on large batches of files, you’ll of course want to script these commands, and also run them on the cluster using pbs or some other batch management system. Typically an experiment will involve ~10-50 mzML files to be run through OpenSwath.

Scripts for doing this are provided at:

[TODO URL – these are in source control at pwiz\_tools\Skyline\Executables\PeakComparison\OpenSwathLinuxScripts]

At the lowest level the basic commands are in:

./prepare.ini

./parameters.ini [filename]

The first file does the decoy generation and conversion to .TraML format, and the second runs OpenSwath on the specified [filename]. I’ve set up to submit these jobs through pbs via:

./runjob.single

which itself is invoked by:

./OpenSwath.py

which is used to specify a list of .mzML files to process in batch.

In running OpenSwath in batch on the cluster, it’s important to properly handle disk access issues. The .mzML files are very large, and the majority of the OpenSwath processing time is spent reading them. Thus, to the extent possible, the input .mzML files, as well as the temporary files created during processing, should be put on as many separate machines as possible, so that cluster processing doesn’t get bottlenecked by all the CPU’s reading from the same disk. Ideally, the .mzML files should be distributed among the cluster CPU’s themselves, so that no remote disk access needs to occur.

## Running mProphet on the Output Data

OpenSwath does not actually output a set of picked peaks – it simply outputs a set of possible candidates for peaks. In order to actually get the picked peaks, you have to run mProphet, another piece of software produced by the Aebersold lab. mProphet is implemented as an R script. Typically, the outputs of the different OpenSwath runs need to all be combined together, and then the compound file needs to be run through mProphet. However, if the compound file is too long, mProphet will hang and not complete. The best solution is usually to combine the files together in groups of ~10 runs. This file combination (which involves simple concatenation plus some additional processing to rename some file names and columns to avoid ambiguity/duplication) can be performed using the “OpenSwathConvert.exe” file available at [TODO URL]. It will look something like this:

OpenSwathConvert.exe file1.tsv file2.tsv file3.tsv file4.tsv file5.tsv file6.tsv file7.tsv file8.tsv file9.tsv file10.tsv openswath\_mprophet.csv

Once you’ve combined a batch of ~10 files, you should run the combined file through mProphet. mProphet is an R-script and thus does not itself need to be installed, but simply downloaded and run. However it requires installing R and a series of perl modules. Installation is described at the following link:

http://www.mprophet.org/#Software\_installation

Once you have installed mProphet, it can be run as follows (adjusting for your R installation etc):

"C:\Program Files\R\R-3.0.2\bin\R.exe" --slave --args mquest=D:\Processing\HasmikQe\OpenSwath\OpenSwathMProphet3.csv workflow=LABEL\_FREE num\_xval=5 run\_log=FALSE write\_classifier=1 write\_all\_pg=0 < mProphet.R

Here, the “mquest=” option is a path to the file you created using OpenSwathConvert.exe. mProphet will output its files in the same directory as the “mquest=” file you specified, there are 4 such files. You want the one that ends in “\_peakgroups.xls”. Open up this file, and save it in a .tsv format. With a couple manual modifications, it can then be converted back to a format that Skyline can import.

## Converting Data Back to Skyline Format

The file produced by mProphet that you just opened is very close to a format that Skyline can read, but a couple manual changes are required:

* Open the .tsv file that you wrote in the last section
* Change the name of the “m\_score” column to “annotation\_QValue”
* Change the name of the “RT” column to “Apex”
* Change the name of the “d\_score” column to “annotation\_Score”

With these changes, Skyline should be able to read the file exported by mProphet.

## Importing Data Back to Skyline

After performing the conversion above, you should be able to import your data back to Skyline in the same manner as described in the “Importing Data Back to Skyline” subsection of the “Peak View” section. You can either compare OpenSwath’s chosen boundaries to manually curated ground truth, or you can directly import OpenSwath’s chosen boundaries in order to visualize/inspect them in Skyline.