DrawAlignR: An interactive tool for across run chromatogram alignment visualization

Justin Sing, Shubham Gupta, Arshia Mahmoodi, and Hannes Röst

Contents

Data Preparation

DrawAlignR requires three types of files that are generated through an OpenSWATH Workflow using OpenMS tools.

For more information on OpenMS, please visit https://www.openms.de/

For more informaiton on the OpenSWATH workflow, please visit http://openswath.org/en/latest/

Before running an OpenSwathWorkflow, you will need to ensure you have the correct input formats. You can convert all raw MS data to standard mzML/mzXML format using ProteoWizard-MSConvert.

Library / Assay Generation

Generating the Target Assay Library

To generate the Targeted Assay library you need to run two OpenMS tools. `OpenSwathAssayGenerator` and `OpenSwathDecoyGenerator`. The former will generate assays for each target pepetide in the dda spectral library, the latter will append decoys to the targeted assay library from FDR estimation.

OpenSwathAssayGenerator

This tool has several parameters that you can adjust to tweak your target library. To understand what the parameters are, you can visit `https://abibuilder.informatik.uni-tuebingen.de/archive/openms/Documentation/release/latest/html/OpenMS_tutorial.html. This includes the documentation for the rest of OpenMS tools.

```
$ ./build/openms_build/bin/OpenSwathAssayGenerator -in
./dda_spectral_library/psgs_standard_consensus_Phospho_pep.TraML -out
./psgs_phospho_optimized.TraML
```

OpenSwathDecoyGenerator

To estimate true signal (true peptides at feature x), we need to differentiate/separate true positives from false positives. In order to do this, we use the decoy model that puts in false signal/peptides. This tool also has several different flags that you can set, such as different decoy methods: `shuffling`, `reverse`, `pseudo-reverse`, etc. and switchKR which switches lysine/argine ends. You can also adjust the ratio of decoy:targets.

```
$ ./build/openms_build/bin/OpenSwathDecoyGenerator -in
./psgs_phospho_optimized.TraML -out ./psgs_phospho_optimized_decoys.TraML
```

TargetedFileConverter

Newer versions of OpenMS have moved towards sqlite data structures, due to more efficient data storage useage. You should convert the TraML target assay library generated in the previous step to the pqp sqlite format.

```
$ ./build/openms_build/bin/TargetedFileConverter -in
psgs_phospho_optimized_decoys.TraML -out psgs_phospho_optimized_decoys.pqp
```

OpenSwathWorkflow

Now that you have generated the targeted assay library, you are ready to run OpenSwathWorkflow.

Default OpenSwathWorkflow

File names below are used as an example, make the output names more understandable.

NOTE: You need to run the OpenSwathWorkflow for each raw MS run file you have.

```
$ ./build/openms_build/bin/OpenSwathWorkflow \
        -in ./run1.mzXML \ ## Raw mzXML data file
        -tr ./lib/psgs/pqp \ ## Targeted assay library file
        -tr_irt ./IRT.TraML \ ## IRT peptides library file
        -out_osw ./run1_MSDATA_RESULTS.osw \ ## name of output file, output
can be .tsv file or .osw (sqlite) file.
        -out_chrom run1.chrom.mzML  \ ## if you want to generate
chromatogram output file. Can be .mzMl or sqMass (sqlite).
        -batchSize 1000 \ ## Size in which to split the data up into for
extraction and scoring
        -readOptions cacheWorkingInMemory \ ## read the data on the fly, or
read from cached in memory
        -tempDirectory ~/scratch/ \ ## temporary directroy to store cached
files
        -threads 6 \ ## number of threads to use for computation
        -debug 10 ## debug level, debug 10 will spit out verbose
information, helpful for debugging
```

Pyprophet

After running OpenSwathWorkflow you want to score the results (extracted peak groups from OpenSwathWorkflow) and estimate the FDR distribution. To do that you can use pyprophet.

Pyprophet contains several steps of it's own.

Pyprophet Merge

You first want to merge each individual osw run file into one osw file, this make things easier for running the workflow and getting global estimates. You can also run it on each file indivdiually, but generally we run the workflow on a merged.osw file.

```
pyprophet merge --template=./lib/psgs_phospho_optimized_decoys.pqp --
out=merged.osw $(find ./results/ -maxdepth 2 -type f -name
*_MSDATA_RESULTS.osw)
```

The command above will call the pyprophet merge function, and merge all the osw files (that are found using the bash find function to return all the files matching "*_MSDATA_RESULTS.osw"), into one osw file using the library file as a template.

Pyprophet Scoring

After you have a merged osw file, you want to perform the semi-supervised scoring algorithm.

Generally you would only need to score on the MS2 level. However, you can also use an integrated scoring on the --level=`ms1ms2`, this will integrate the scores between MS1 features and MS2 features.

```
pyprophet score --in=./merged.osw --level=ms2 --ss_num_iter=10 --
xeval_num_iter=10 --threads=10
```

The command above will score on the merged_runs.osw file, using the MS2 features, using the semi-supervised learning on 10 iterations (this is the default) and performing 10 cross-validations (this is the default) using 10 threads.

Required Input Data for DrawAlignR

Directory Structure

To make things easier, and to have a contained experiment, DrawAlignR utilizes a structured directory as input.
The top level experiment directory should contain sub-directories:

- containing the merged pyprophet scored `osw` file obtained from the OpenSwathWorkflow and pyprophet scoring workflow,
- the extracted chromatogram files (`mzML` or `sqMass`) obtained from OpenSwathWorkflow, and
- the `pqp` library assay file obtained from OpenSwathAssayGenerator and OpenSwathDecoyGenerator.

DrawAlignR, can alternatively also allow for individual file input without having a structured working directory, however, it is suggested to have one.
Please see below for an example structured working directory.

An example Working Directory should be ideally structured and named as below:

```
/Project_Working_Directory
|__/mzml
|   |__/run0.chrom.mzML
|   |__/run1.chrom.mzML
|__/osw
|   |__/merged.osw
|__/pqp
|   |__/assay_library.pqp
|__/sqmass(Optional, either mzml or sqmass format is acceptable)
|   |__/run1.chrom.sqMass
|   |__/run1.chrom.sqMass
```

Data Size

Depending on how complex your experiments are and how many runs you may have, the extracted chromatogram files may be large files. The larger your files are, the longer it may take for DrawAlignR to cache all chromatographic data into memory, this is something to keep in mind.

Downloading Tutorial Dataset

We have example datasets hosted on PeptideAtalas PASS01520

This dataset contains two datasets, which both have a full dataset and a smaller data subset:

- Spyogenes
  - Spyogenes_Full
  - Spyogenes_Small_Subset
- Synthetic_Phosphopeptide_Dataset
  - Synthetic_Dilution_Phosphoproteomics_Full
  - Synthetic_Dilution_Phosphoproteomics_Small_Subset

NOTE: Using the smaller data subsets, will be quicker for tests, and alignment parameter checks, since loading full extracted chromatogram data may take longer to cache into memory.

The easiest and most straight forward way of downloading the data is to use your browser's FTP mode: ftp://PASS01520:HE7445u@ftp.peptideatlas.org/

Retrieving Data on MacOS

MacOS is capable of establishing a connection to a remote FTP server using the native Finder application.

- Open a **Finder Window**
- From the main menu, select **Go** -> **connect to Server**
- In the pop-up dialog box, type `ftp://PASS01520:HE7445u@ftp.peptideatlas.org/` in the *Server Address:* field
- Press **Connect**
- enter Username/Password
  - Username: PASS01520
  - Password: HE7445u

Alternatively, you can type the ftp address into a safari browser.

Retrieve Data on Windows

On Windows, you can use the Windows file manager/ File Explorer to connect to a remote FTP server.

- Open a **File Explorer**
- Right click on **This PC / Computer** ** Select **Add a network location**
- Go through pop-up wizard and select **Choose a custom network location**
- In the “Specify the location of your website” dialog, enter `ftp://PASS01520:HE7445u@ftp.peptideatlas.org/`
- enter Username/Password
  - Username: PASS01520
  - Password: HE7445u

Retrieve Data on Linux/ Ubuntu

On Ubuntu, you can use nautilus file manager to connect to a remote FTP server.

- Click on **Other Locations**
- In the bottom field to the right of *Connect to Server**, enter `ftp://PASS01520:HE7445u@ftp.peptideatlas.org/`
- enter Username/Password
    - Username: PASS01520
    - Password: HE7445u

Using/ Calling DrawAlignR

The visualization tool can be called via two methods, either via installing the repository using devtools github package installer or by calling shiny::runGitHub.

Calling tool via Installing DrawAlignR

```
### Install devtools if not already installed.
## install.packages("devtools")
## Install DrawAlignR from Roestlab Github Repository
devtools::install_github("Roestlab/DrawAlignR")
## To run the tool
DrawAlignR::runDrawAlignR()
```

OR

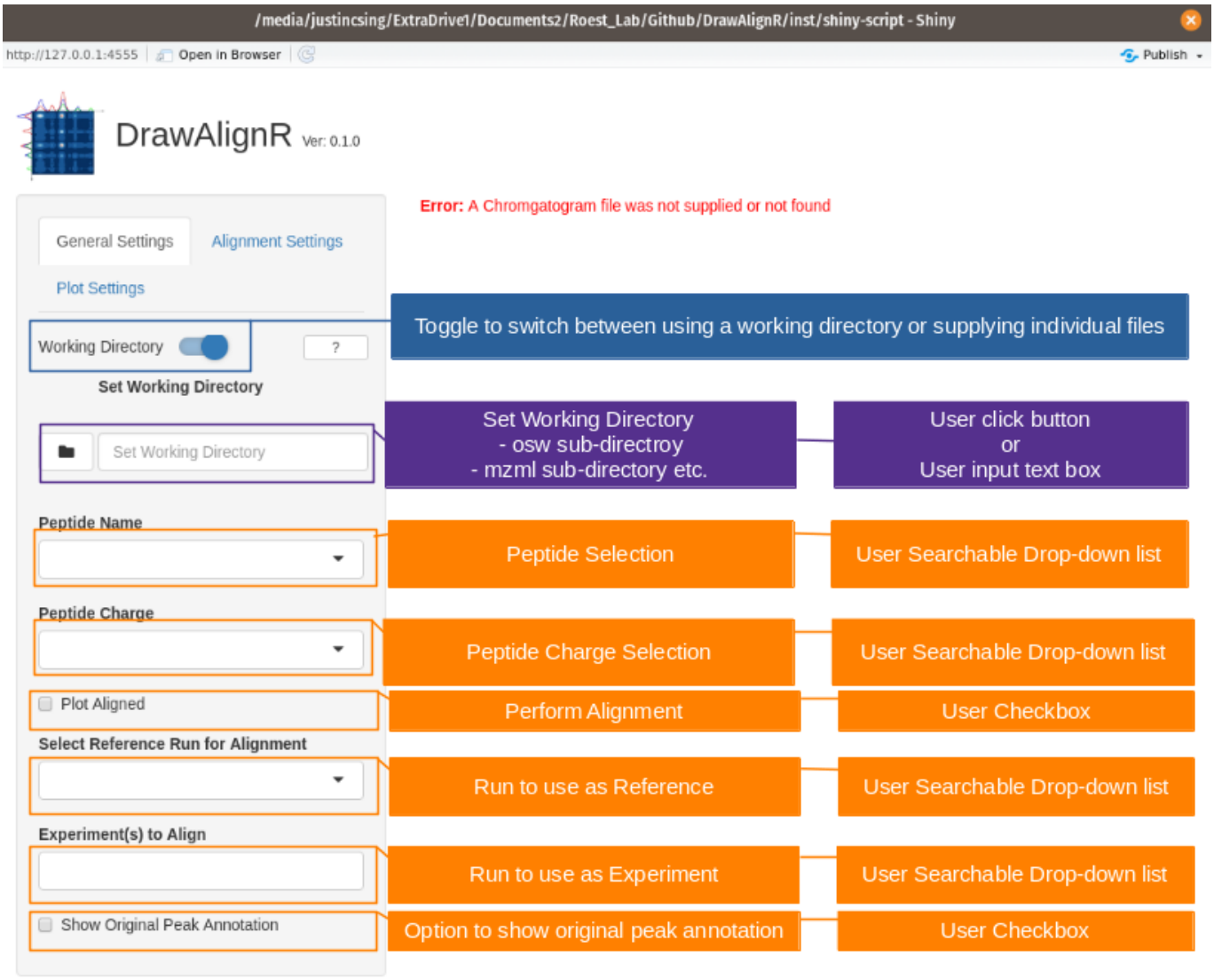
Calling tool via Github repo

```
### Install shiny
## install.packages("shiny")
shiny::runGitHub(repo = "Roestlab/DrawAlignR/", username = "Roestlab",
subdir = "inst/shiny-script")
```

User Interface

There are three major tabs in the left side pannel:

- **General Settings**
  - These are general settings for uploading chromatogram files (.mzML or .sqMass), library assay files (.pqp), OpenSwathWorkflow results file (.osw). The user can also set the working directory that contains sub-directroys for osw and mzml files.
  - The user selects the peptide and charge state to visualize.
  - The user selects how many plots to show for each chromatogram run file suppled.
  - The user can select the alignment option to perform an alignment for the selected peptide.
  - The user can visualize the reference plot, experiment plot and the experiment aligned plot.
- **Alignment Settings**
  - The user can change various alignment parameters
- **Plot Settings**
  - The user can change various plot visualization settings


/media/justincsing/ExtraDrive1/Documents2/Roest_Lab/Github/DrawAlignR/inst/shiny-script - Shiny
http://127.0.0.1:4555
Open in Browser
Publish
DrawAlignR
Ver: 0.1.0
Error: A Chromgatogram file was not supplied or not found
General Settings
Alignment Settings
Plot Settings
Working Directory
?
Set Working Directory
Set Working Directory
Peptide Name
Peptide Charge
Plot Aligned
Select Reference Run for Alignment
Experiment(s) to Align
Show Original Peak Annotation
Toggle to switch between using a working directory or supplying individual files
Set Working Directory
- osw sub-directroy
- mzml sub-directory etc.
User click button
or
User input text box
Peptide Selection
User Searchable Drop-down list
Peptide Charge Selection
User Searchable Drop-down list
Perform Alignment
User Checkbox
Run to use as Reference
User Searchable Drop-down list
Run to use as Experiment
User Searchable Drop-down list
Option to show original peak annotation
User Checkbox

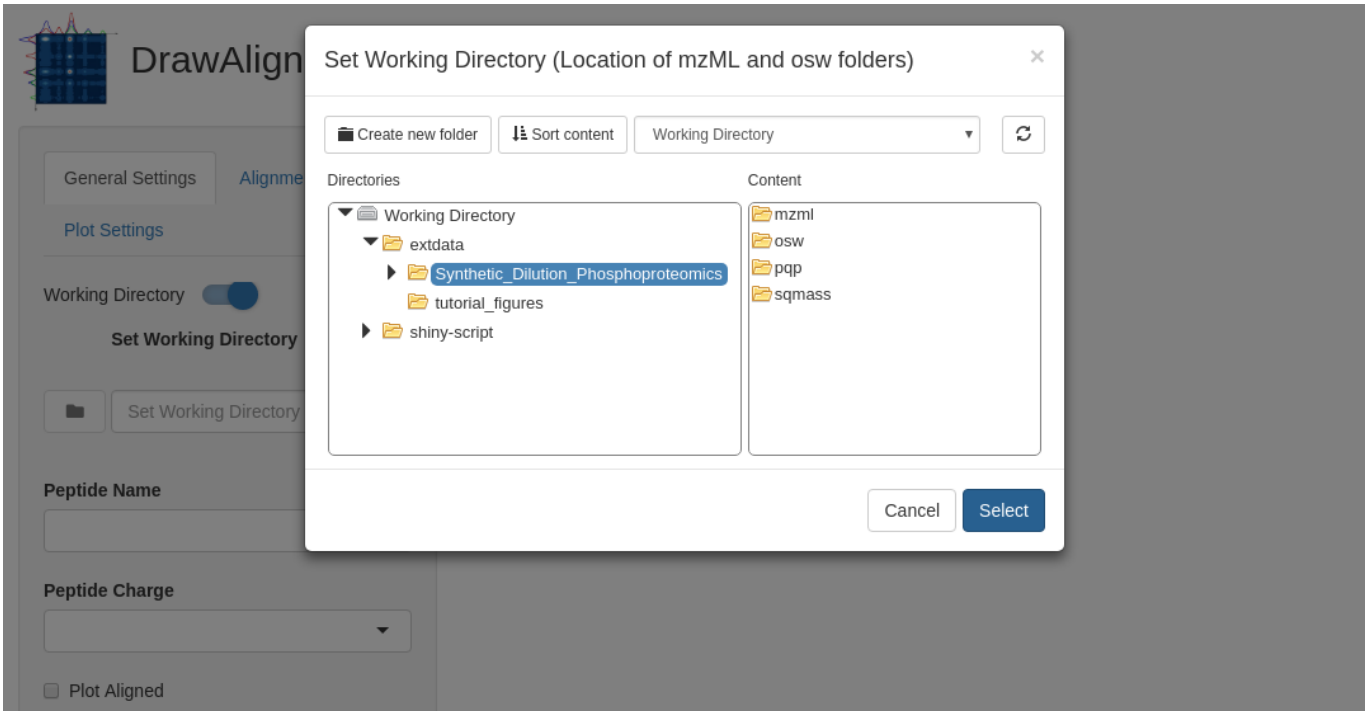
Tutorial for Performing Alignment

Set Working Directory

Use the Set Working Directory button to set the working directory that contains an mzml folder with .chrom.mzml files and an osw file with a merged.osw file, and optionally a library file in the .pqp format. Or you can directly enter the path to the working directory using the input textbox area. DrawAlignR will parse the working directory and look for `../osw`, `../mzml` (or `../sqmass`), and `../pqp` subfolders containing their corresponding file types. You can alternatively toggle the **Working Directory** button to enter chromatogram, library and osw files seprately. **Note:** For alignment it is preferred if you enter a working directory.

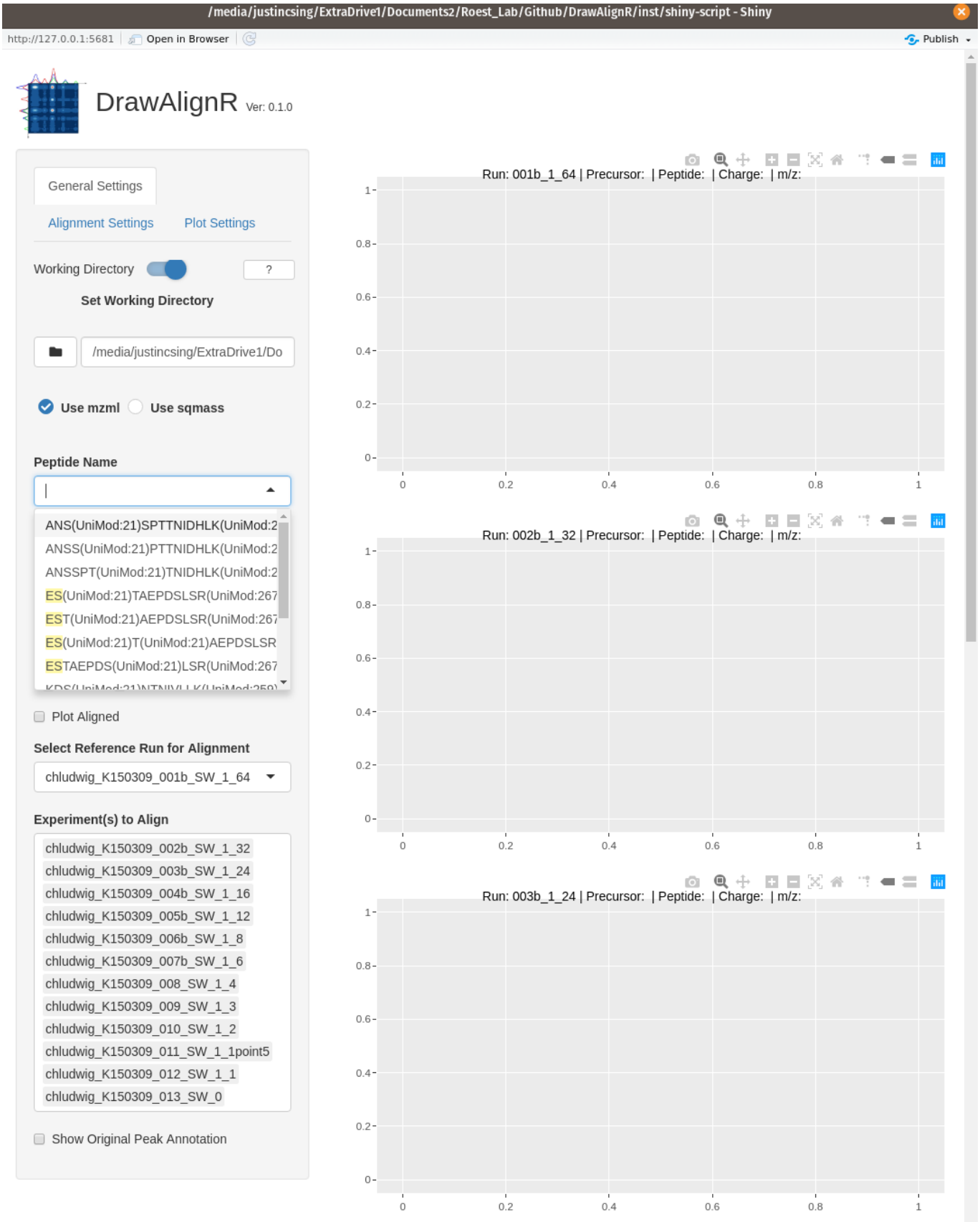
An example Working Directory should be ideally structured and named as below:

```
/Project_Working_Directory
|__/mzml
|   |__/run0.chrom.mzML
|   |__/run1.chrom.mzML
|__/osw
|   |__/merged.osw
|__/pqp
|   |__/assay_library.pqp
|__/sqmass(Optional, either mzml or sqmass format is acceptable)
|   |__/run1.chrom.sqMass
|   |__/run1.chrom.sqMass
```



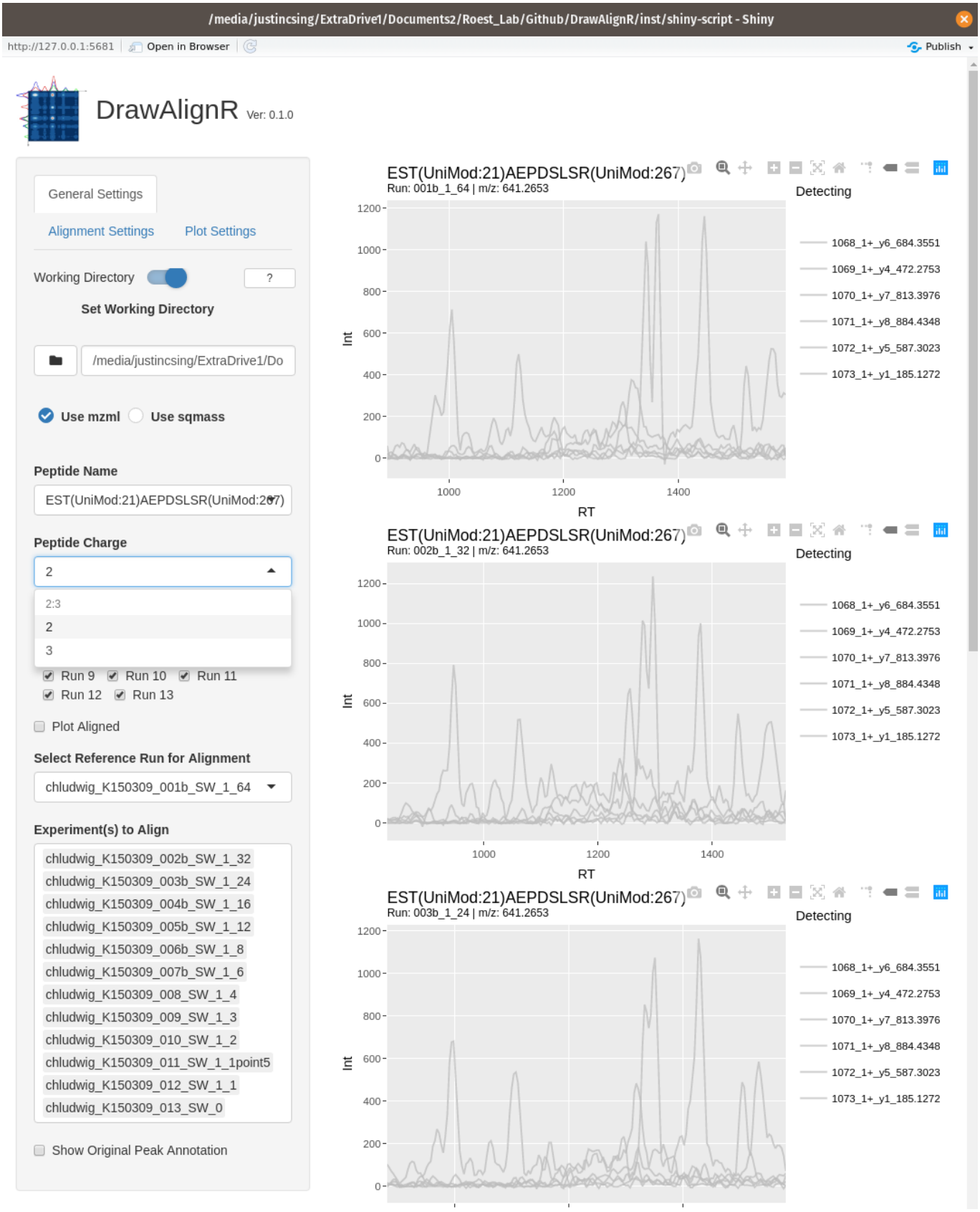
Select a Peptide to visualize Chromatogram alignment

Choose which peptide you want to visualze using the Peptide dropdown list. The dropdown list is searchable, so you can easily search for a specific peptide to visualize. The list of peptides is extracted from either the input library file if abailable, or an osw file if available.



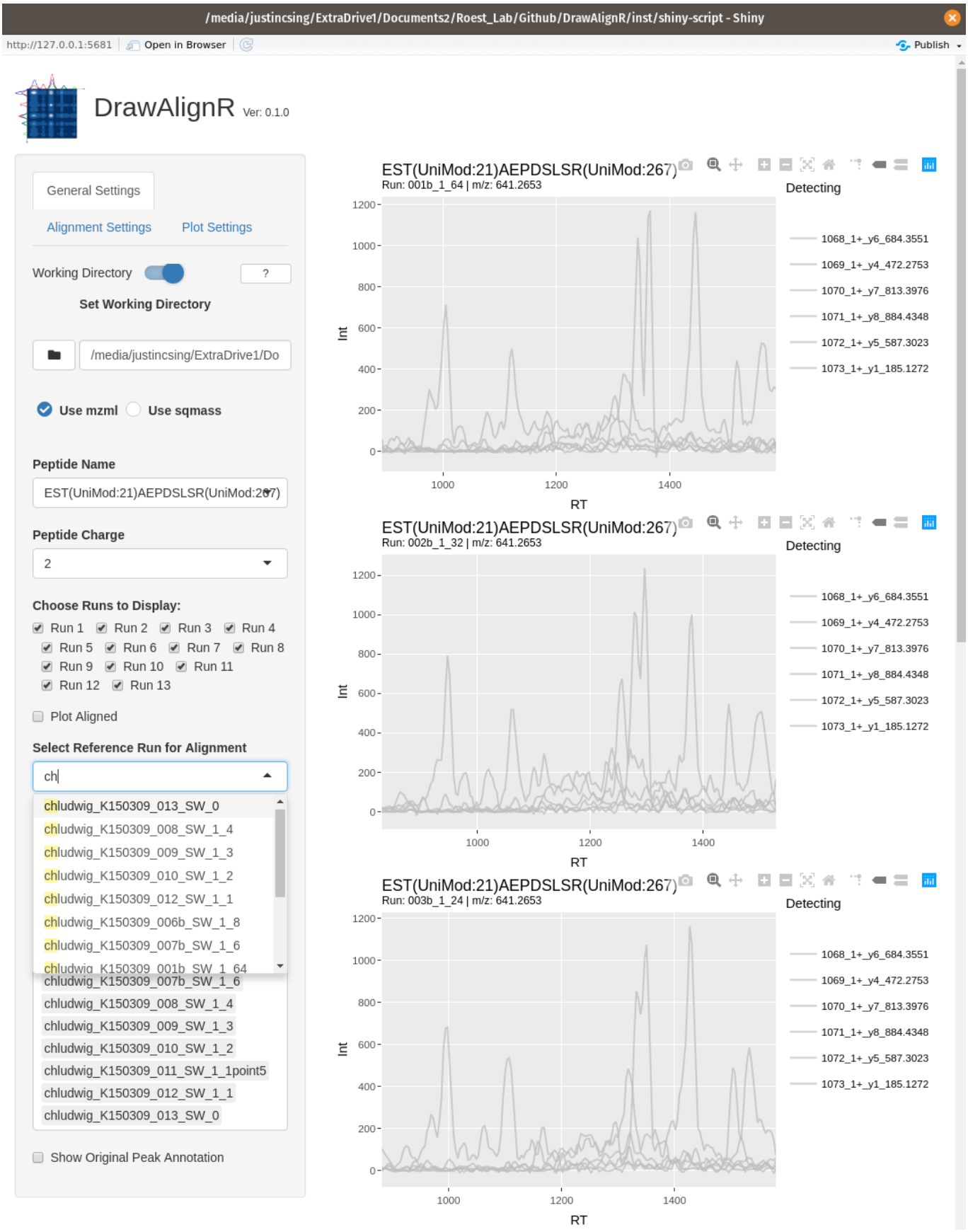
Select a Charge State for Selected Peptide

Choose which charge state to visualize for the selcted peptide.



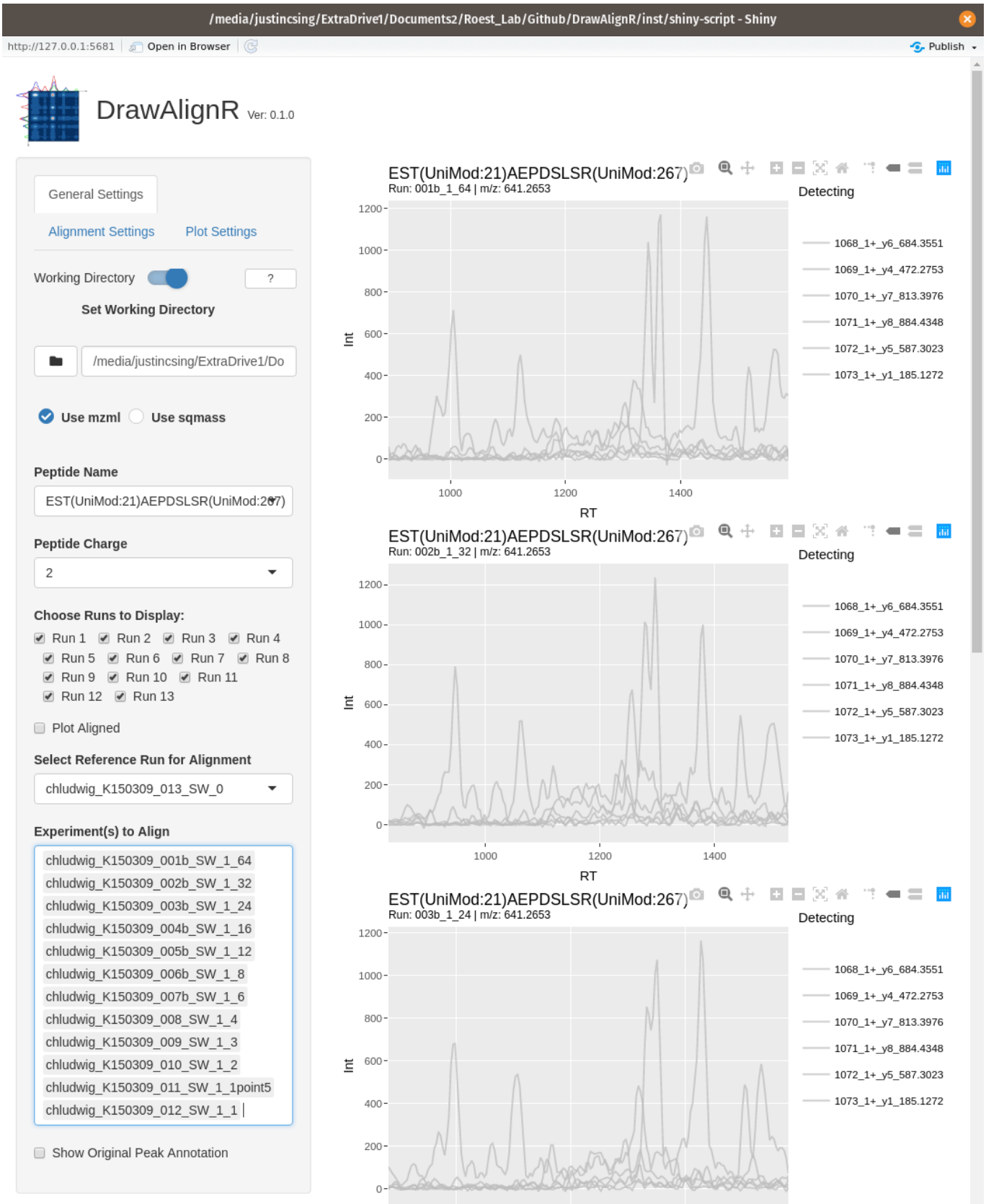
Select Reference Run

Choose which chromatogram file to use as the reference run. These are set through the searchable dropdown lists, which extracts the filenames from the supplied chromatogram files without the .chrom.mzml extension



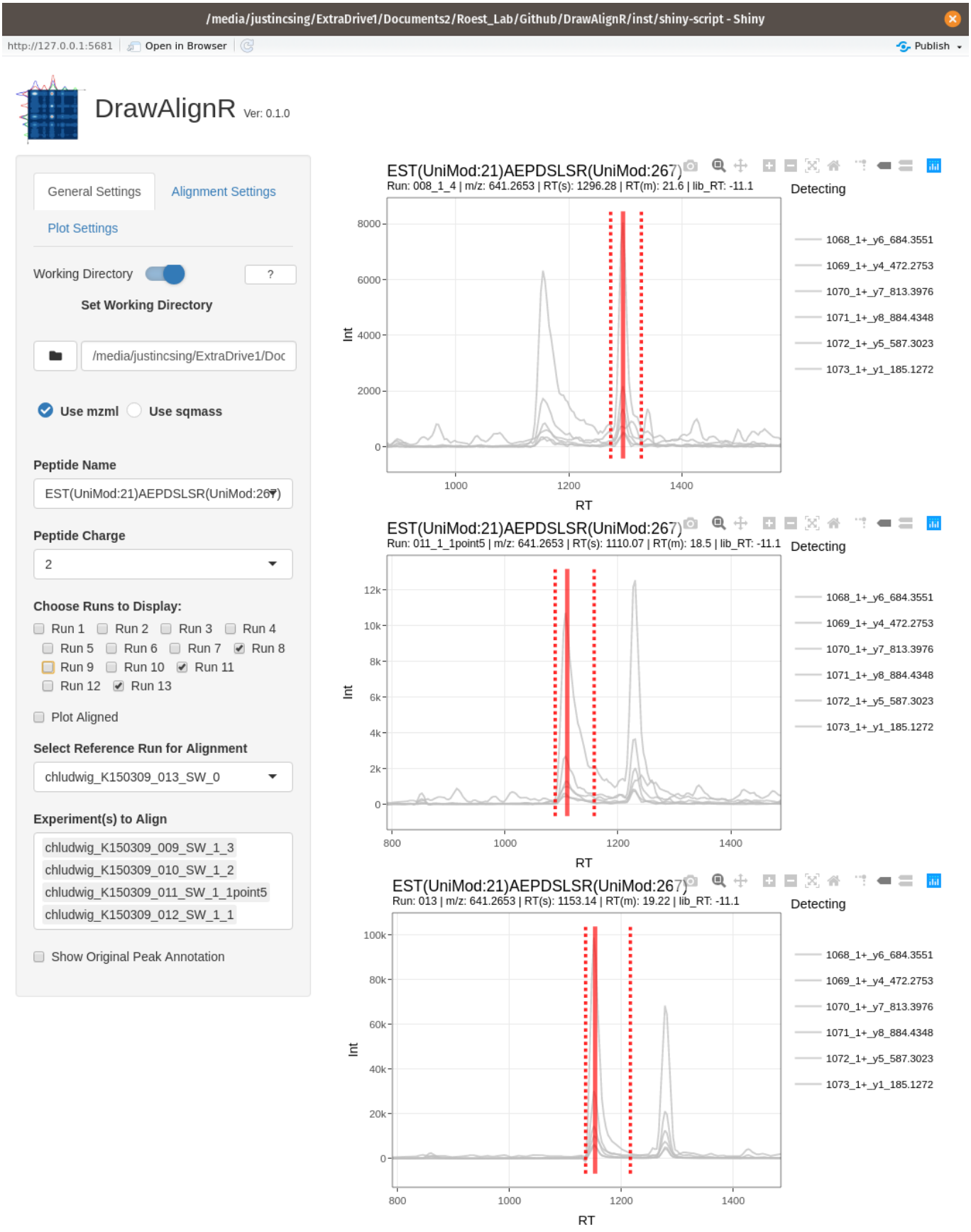
Select Experiment Run(s)

Choose which chromatogram file to use as the experiment run. These are set through the searchable dropdown lists, which extracts the filenames from the supplied chromatogram files without the .chrom.mzml extension



Select Which Chromatogram Runs to Display

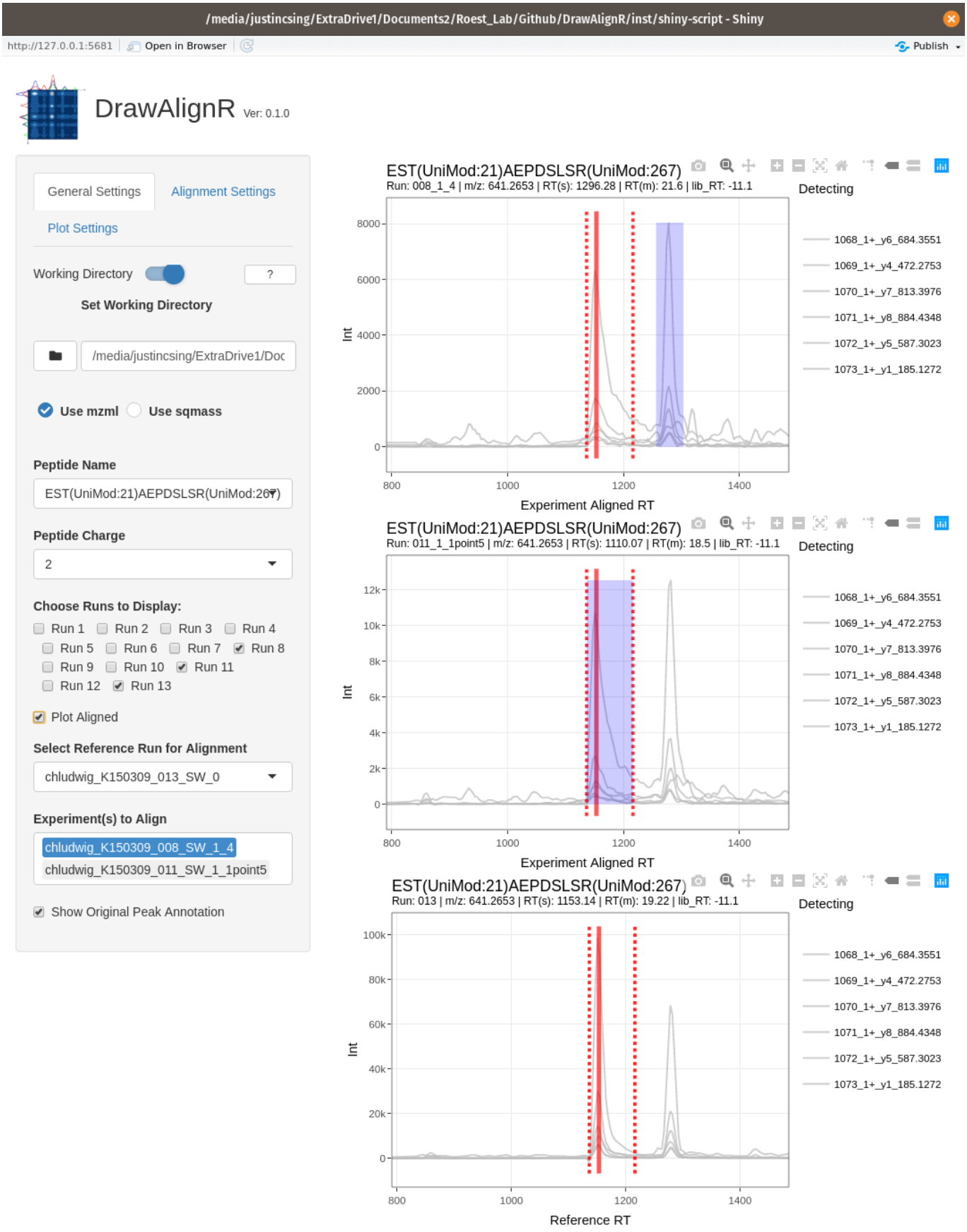
Choose which chromatogram runs to display by checking or unchecking the Run n checkbox.



Select Plot Align checkbox to perform alignment

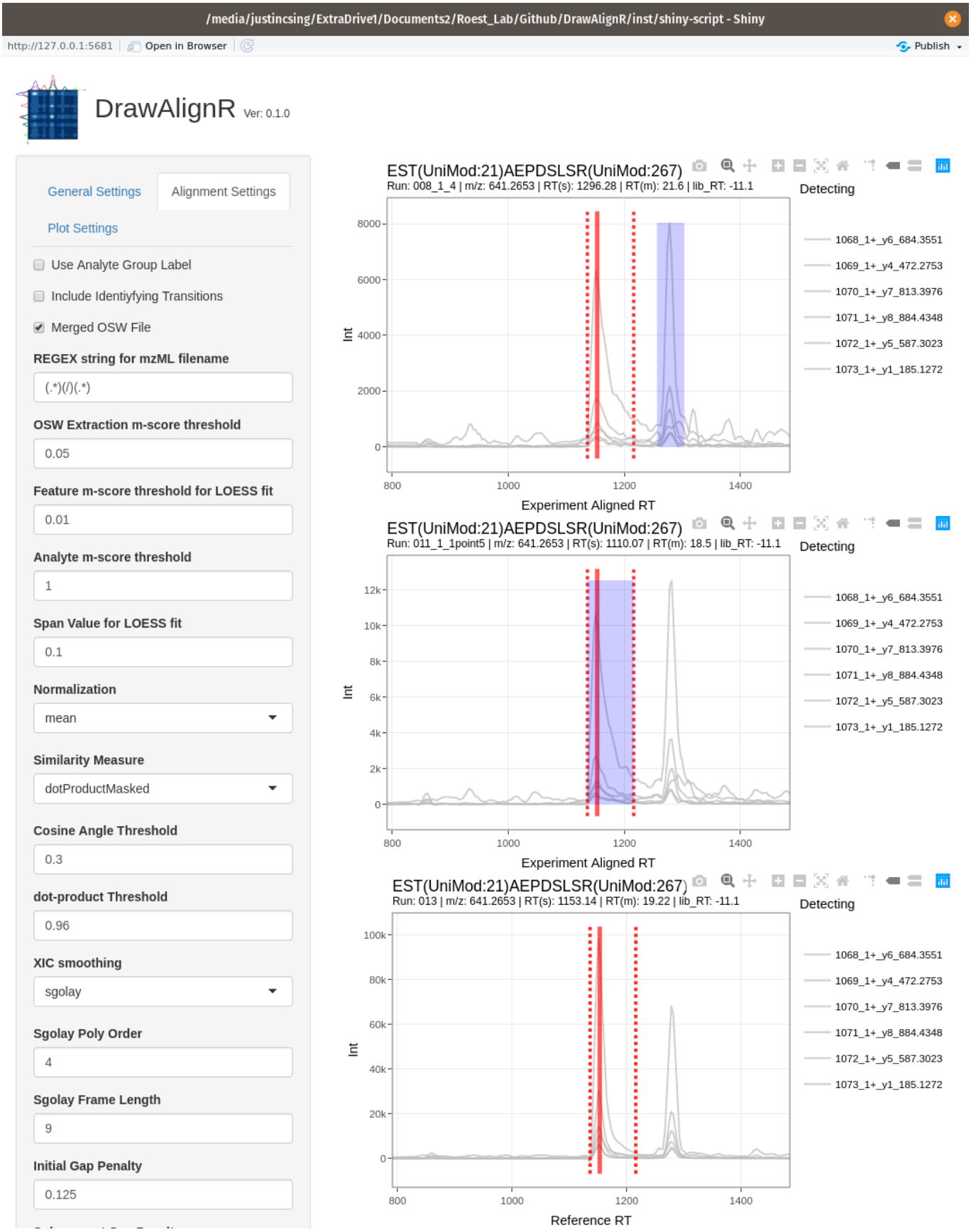
Check the Plot Aligned checkbox to perform the alignment of the two runs for the seleected peptide You can also use the

- Show Original Peak Annotation to show where the original peak was annotated before alginment.

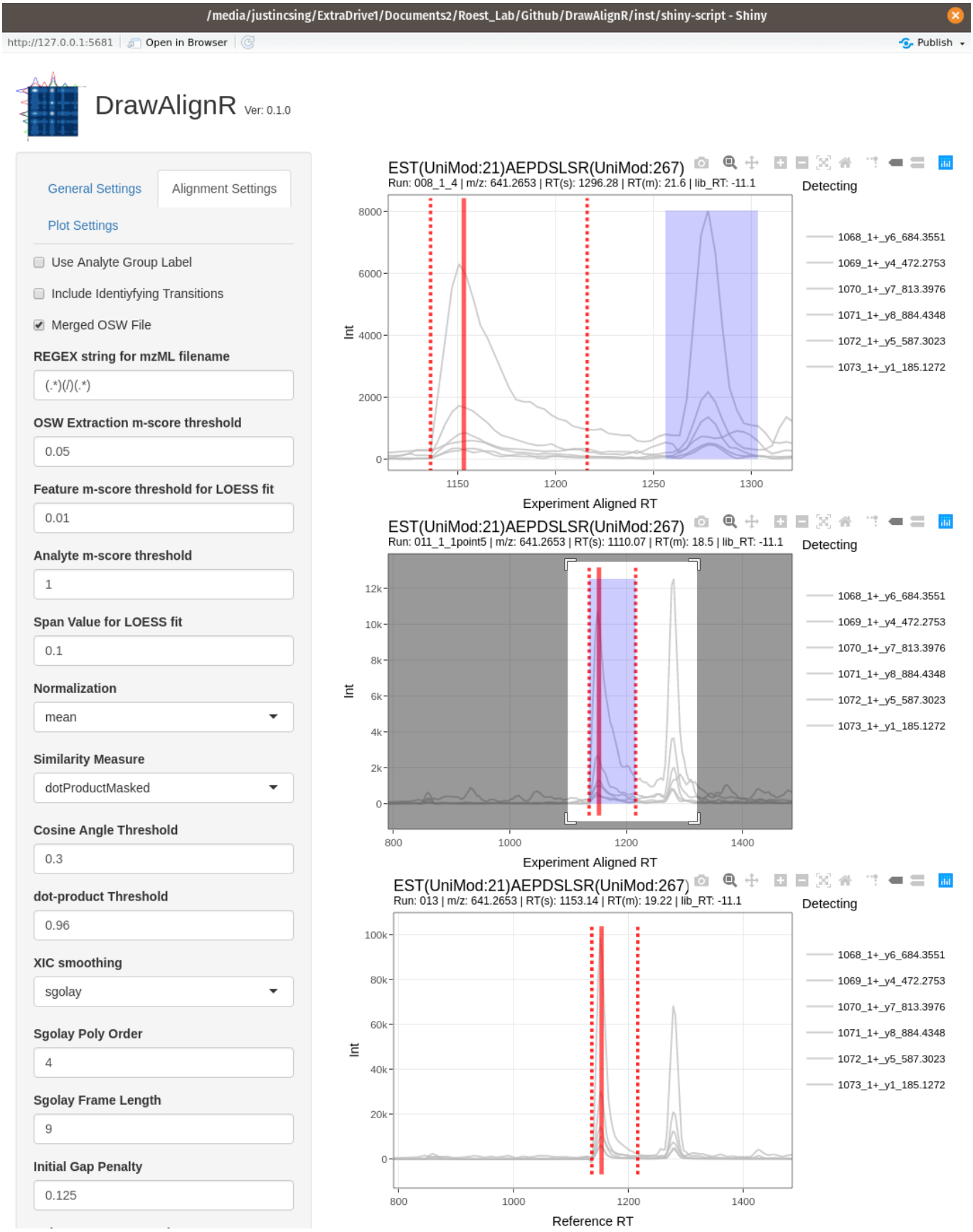


Change Alignment Parameters

Select the Alignment settings tab to change various alignment settings

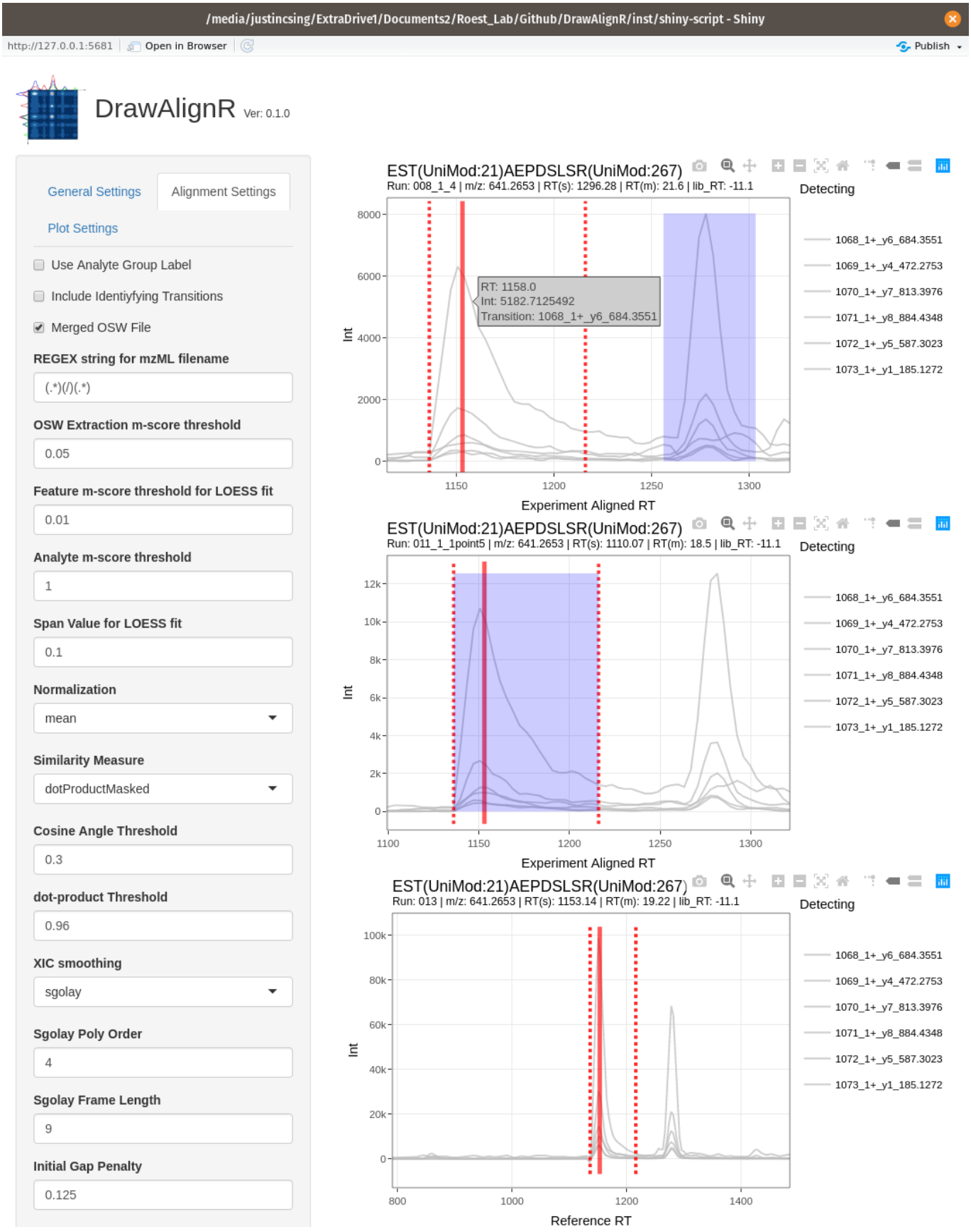


Zoom into each chromatogram for further inspection



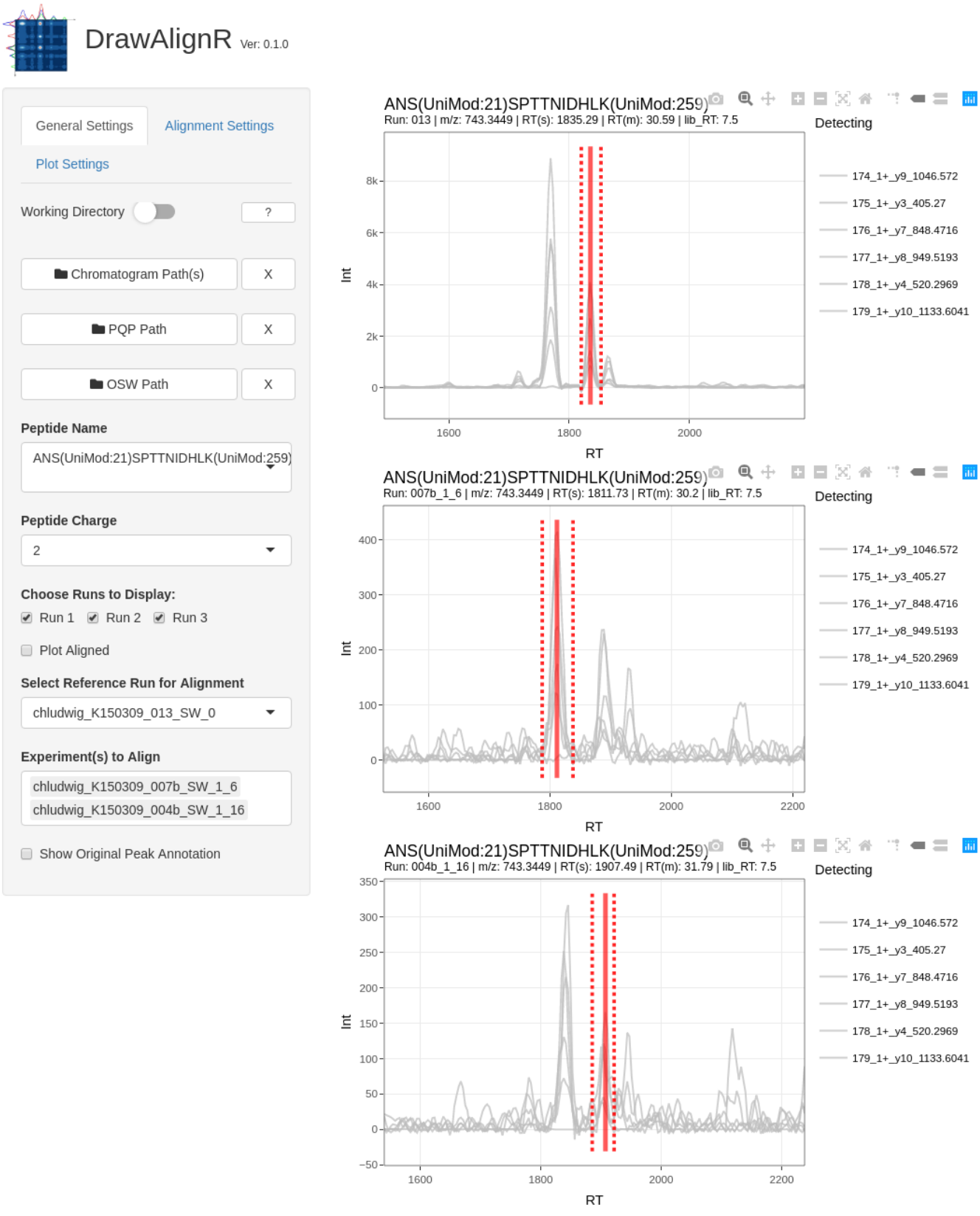
Use the Hover-tooltip

You can hover over the chromatogram traces to see information such as Retention time and Intensity



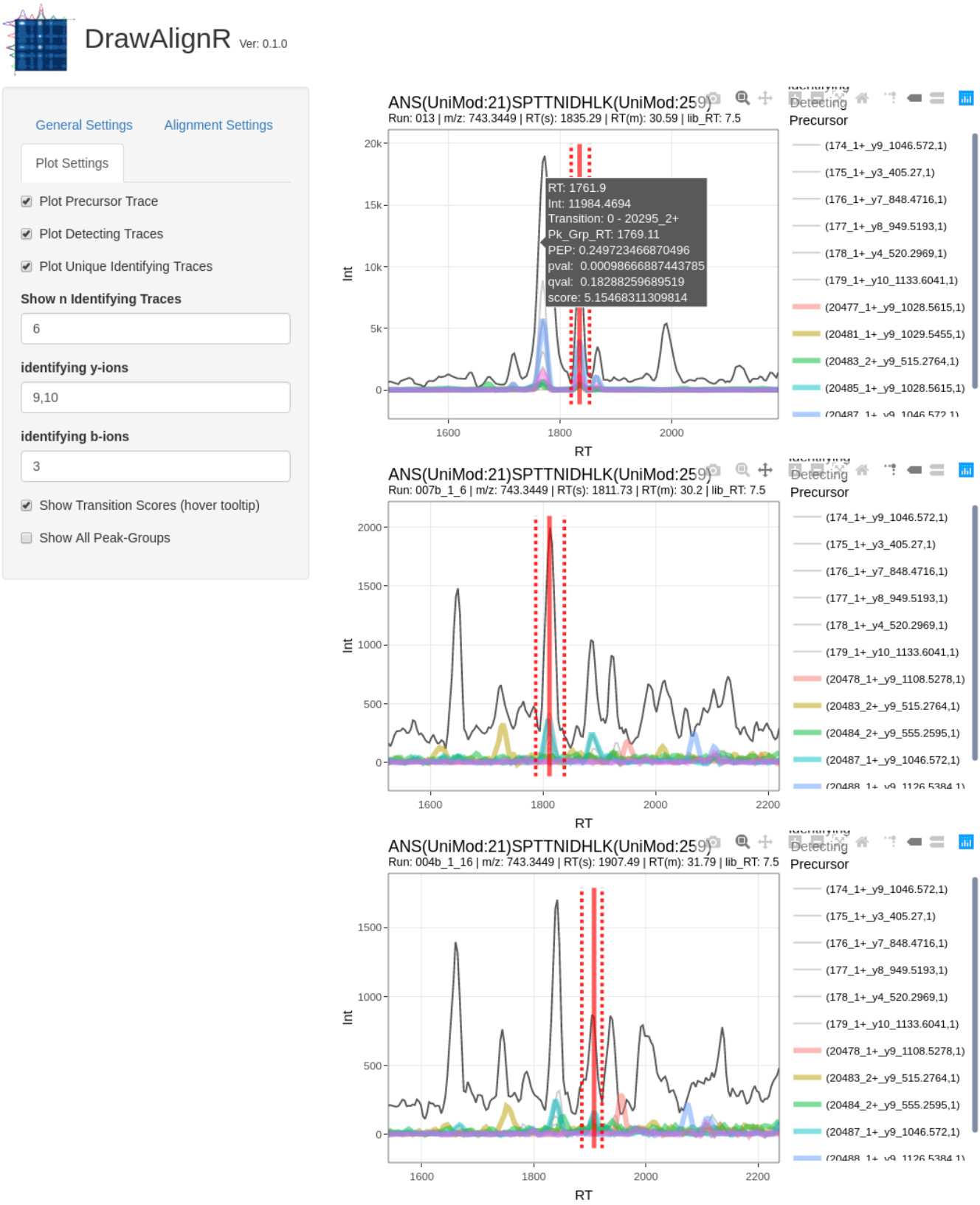
Visualizing Chromatograms

The user can also just visualize the chromatograms alone without performing alignment to visually inspect each trace. If the user has an IPF dataset, they can visualize site-determining ions (unique identifying transitions) of the modified peptide.



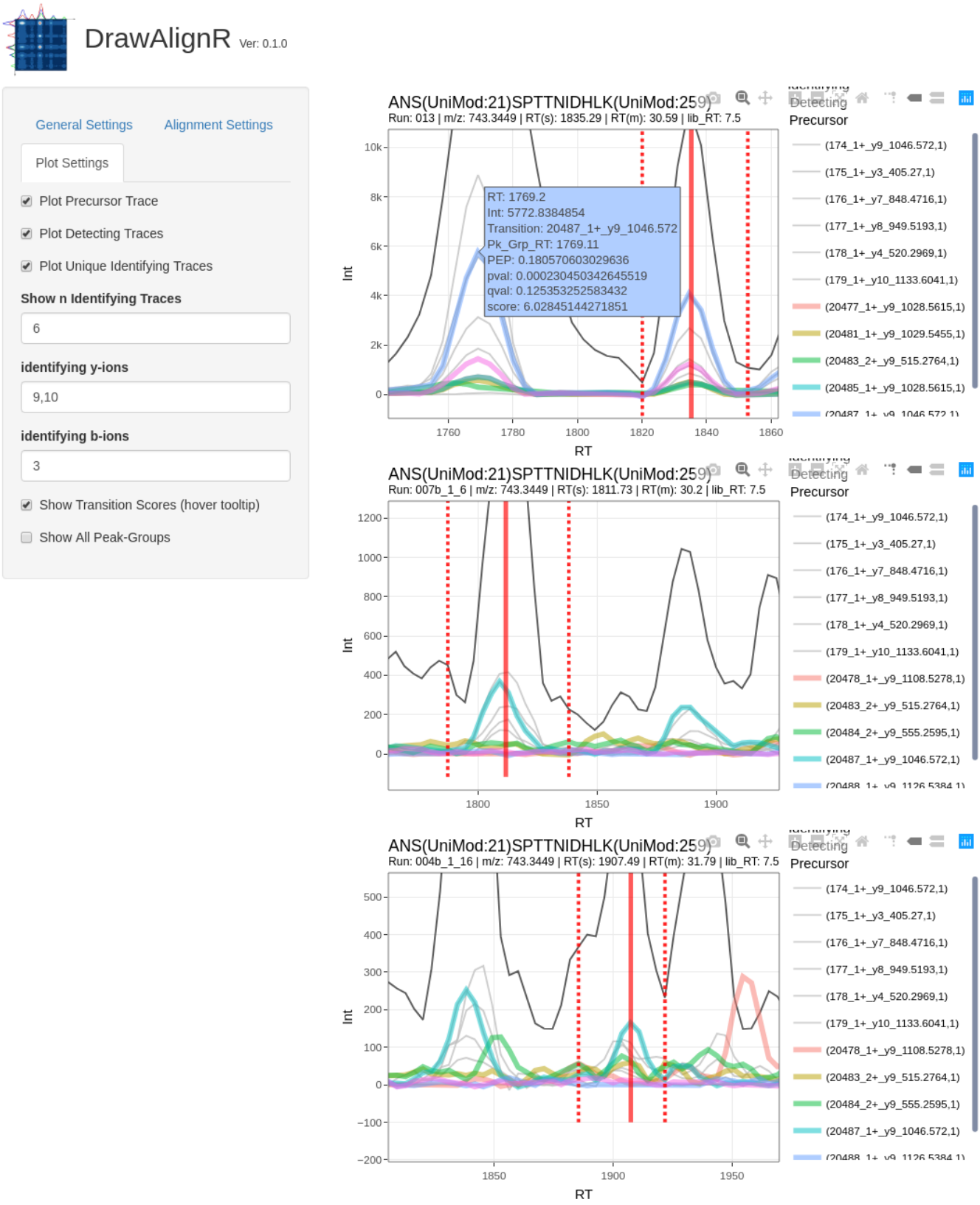
Precursor, Detecting, Identifying

The user can choose to display the precursor tace, or the 6 detecting traces, or the unique identifying traces. The precursor trace is displayed in `black`, the detecting traces are displayed in a `light gray` and the unique identifying transitions are `colored`



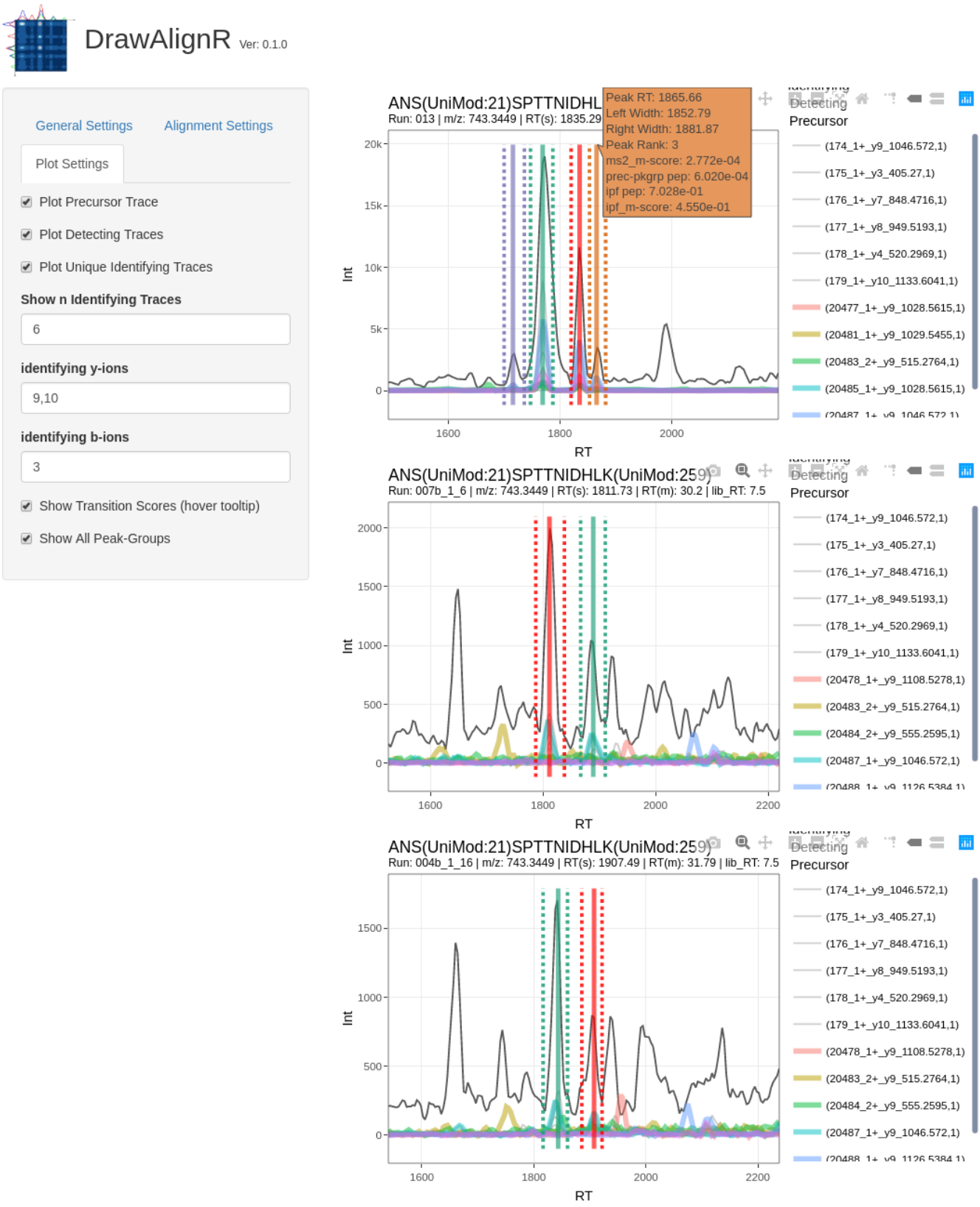
Displaying Transition Scores

The user can hover of the traces to display the transition scores such as the transitions posterior error probability, q-value and score.



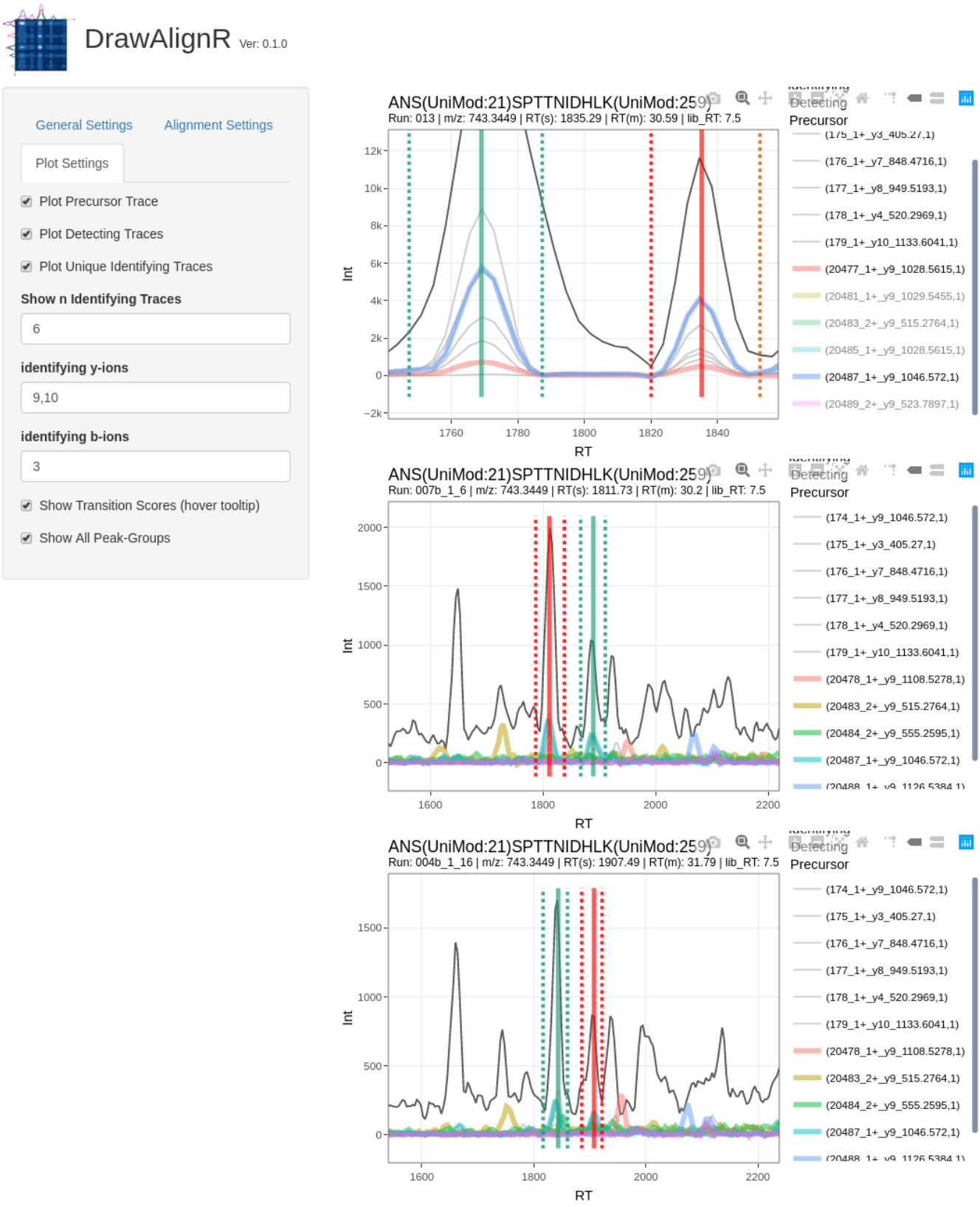
Displaying all Peak-Group Ranks

The user can choose to display the other potential peak-group ranks found by OpenSWATH



Unselecting a Few Transitions to Display

The user can click on the legend to hide transitions they don't want to display



Displaying a Single Transition

The user can choose to display a single transition by double clicking on the transition legend they wish to dispaly