iSwathX
An R package for extending the SwathXtend functions and develop a web application

Zainab Noor
Macquarie University, Australia
zainab.noor@students.mq.edu.au

April 13, 2018

iSwathX is an R package which inherits basic functions from the 'SwathXtend' package and extends them to perform the novel functionalities.Functions from this package has been used to develop a web application named 'iSwathX' available on shiny server (https://biolinfo.shinyapps.io/iSwathX). It aimed to provide an automatic pipeline for extended assay library generation and visualization. This vignette describes how to use the funcitons in iSwathX.

Introduction

The first integrated DIA and quantitative analysis protocol, termed SWATH was shown to offer accurate, reproducible and robust proteomic quantification (Gillet et al 2012). An important concept in DIA analysis is use of a LC-retention time referenced spectral ion library to enable peptide identification from DIA generated multiplexed MS/MS spectra. iSwathX is an R based software package with a web application to facilitate the generation of extended assay libraries for SWATH data extraction, making it compatible to the most commonly used DIA data analysis software.

Package installation

To install the iSwathX package the following commands can be executed within R.

```
> install.packages("devtools")
> library(devtools)
> install_github("znoor/iSwathX", subdir = "package")
```

Typically the workspace is cleared and the iSwathX pakcage is loaded.

```
> rm(list=ls())
> library(iSwathX)
```

The example data, that is included in the package, consists of five assay libraries. The libraries can be loaded using *readLibFile*. Library format can be

“PeakView” (AB Sciex 2014), “OpenSWATH” (Rost et al. 2014) “Skyline” () or “Spectronaut” () format which is in a tab-delimitated .txt or comma-delimitated .csv file. The parameter *clean* in function *readLibFile* specifies if the library to be clearned, which will be describe later.

```
> filenames <- c("Lib2.txt", "Lib3.txt")
> libfiles <- paste(system.file("files",package="iSwathX"),
+         filenames,sep="/")
> Lib2 <- readLibFile(libfiles[1], clean=TRUE)
> Lib3 <- readLibFile(libfiles[2], clean=TRUE)
```

If the file format is “peakview”, it requires the following columns:

- Q1: Q1 m/z (precursor m/z)
- Q3: Q3 m/z (fragment m/z)
- RT_detected: retention time
- protein_name: protein name
- isotype: isotype type
- relative_intensity: fragment ion intensity
- stripped_sequence: peptide sequences without modifications
- modification_sequence: peptide sequences with modifications
- prec_z: peptide charge
- frg_type: fragment type (b or y ion)
- frg_z: fragment charge
- frg_nr: ion number
- iRT: calibrated retention time (the values might not be meaningful for experiments with no iRT peptides spiked in)
- uniprot_id: database accession number
- decoy: whether the peptide a decoy or not (TRUE or FALSE)
- confidence: the confidence of the identified peptide (a value between 0 and 1)
- shared: whether the peptide is shared by multiple proteins (TRUE or FALSE)
- N: a ranking number for the protein

Optional columns for PeakView format libaries include:

- score: score for peptide identification
- prec_y: the precursor ion intensity

- rank: ion intensity ranking
- mods: modification
- nterm: N terminal modification
- cterm: C terminal modification

If the file format is "openswath", it must contain the following columns:

- PrecursorMz: precursor m/z
- ProductMz: fragment m/z
- Tr_recalibrated: retention time
- ProteinName: protein name
- GroupLabel: isotype type
- LibraryIntensity: fragment ion intensity
- PeptideSequence: peptide sequences without modifications
- FullUniModPeptideName: peptide sequences with modifications
- UniprotID: database accession number
- decoy: whether the peptide a decoy or not
- PrecursorCharge: precursor charge
- FragmentType: fragment type
- FragmentCharge: fragment charge
- FragmentSeriesNumber: fragment ion number

If the file format is "skyline", it must contain the following columns:

- PrecursorMz: precursor m/z
- ProductMz: fragment m/z
- Tr_recalibrated: retention time
- ProteinName: protein name
- Isotype: isotype type
- LibraryIntensity: fragment ion intensity
- PeptideSequence: peptide sequences without modifications
- ModificationSequence: peptide sequences with modifications
- UniprotID: database accession number
- decoy: whether the peptide a decoy or not

- PrecursorCharge: precursor charge
- FragmentType: fragment type
- FragmentCharge: fragment charge
- FragmentSeriesNumber: fragment ion number

Optional columns include:

- rank: ion intensity ranking
- nterm: N terminal modification
- cterm: C terminal modification

If the file format is “spectronaut”, it must contain the following columns:

- PrecursorMz: precursor m/z
- FragmentMz: fragment m/z
- RetentionTime: retention time
- iRT: indexed retention time
- ProteinGroups: protein name
- RelativeIntensity: fragment ion intensity
- StrippedPeptide: peptide sequences without modifications
- ModifiedPeptide: peptide sequences with modifications
- UniProtIds: database accession number
- PrecursorCharge: precursor charge
- FragmentType: fragment type
- FragmentCharge: fragment charge
- FragmentNumber: fragment ion number

Optional columns include:

- rank: ion intensity ranking
- decoy: whether the peptide a decoy or not
- nterm: N terminal modification
- cterm: C terminal modification

Building extended assay library

To build an extended library using iSwathX, one seed library and one add-on library are needed. The seed library is usually a local assay library which was generated with SWATH data using the same instrument and the same chromatography condition. The add-on library can be a local archived assay library or an external library downloaded from public data repositories such as SWATHAtlas(Biology IfS 2014).

Library cleaning

All candidate assay libraries were first subject to a cleaning process which removes low confident peptides and low intensity ions by user-defined thresholds. The default values for these two thresholds are 99% for peptide confidence and 5 for ion intensity. The clearning process can also opt to remove peptides with modifications for miss cleavages. The clearning process can be done separately using function *cleanLib* or as part of the library reading process as shown above.

```
> Lib2 <- cleanLib(Lib2, intensity.cutoff = 5, conf.cutoff = 0.99,
+                         nomod = FALSE, nomc = FALSE)
> Lib3 <- cleanLib(Lib3, intensity.cutoff = 5, conf.cutoff = 0.99,
+                         nomod = FALSE, nomc = FALSE)
```

Library summary

The function *libSummary* generates the library summary. It displays information related to number of proteins, peptides (precursor ions) and fragment ions in assay libraries.

```
> libSummary(Lib2)

$proteins
[1] 20

$peptides
[1] 346

$transitions
[1] 1745

> libSummary(Lib3)

$proteins
[1] 20

$peptides
[1] 581

$transitions
[1] 2940
```

Library format conversions

All candidate assay libraries can be converted into different file formats.

PeakView format will be returned using function *peakviewFormat*.

```
> peakviewFormat(Lib2)
```

OpenSwath format will be returned using function *OswathFormat*.

```
> OswathFormat(Lib2)
```

Skyline format will be returned using function *skylineFormat*.

```
> skylineFormat(Lib3)
```

Spectronaut format will be returned using function *spectronautFormat*.

```
> spectronautFormat(Lib3)
```

Matching quality checking

It is very important to check the matching quality between the seed and add-on libraries before building the extended library. Function *checkQuality* can be used to perform the library matching quality check based on the retention time and the relative ion intensity.

Retention time correlation and predicted average error of RT plots by retention times will be returned using function *computeRTTime*.

```
> list.timecor <- computeRTTime(Lib2, Lib3)
> list.timecor[[1]]
```

```
[1] 0.9778811
```

Retention time correlation and predicted average error of RT plots by hydrophobicity indexes will be returned using function *computeRTHydro*.

```
> hydroFile <- paste(system.file("files", package = "iSwathX"),
+                                         "hydroIndex.txt", sep = "/")
> hydro <- readLibFile(hydroFile, type = "hydro")
> list.hydrocor <- computeRTHydro(Lib2, Lib3, hydro)
> list.hydrocor[[1]]
```

```
[1] 0.8968572
```

Relative ion intensity correlation (median spearman) of common fragment ions between two libraries will be returned using function *computeRTTime*.

```
> Lib2 <- normalise(Lib2)
> Lib3 <- normalise(Lib3)
> list.intensitycor <- computeIntensityCor(Lib2, Lib3)
> list.intensitycor[[1]]
```

We recommend if RT correlation R^2 is greater than 0.8, the RMSE less than 2 and intensity correlation is greater than 0.6, the two libraries have good matching quality. We suggest the integration of libraries should be performed only when the retention time (RT) and relative intensity (RI) matching quality are good.

Various statics about the two libraries can be plotted and exported into a multi-tab spreedsheet using *plotStats* function. These include barplots of the number of proteins and peptides of the seed library, add-on library and their relationship (including overlapping proteins, peptides, retention time scatter plots and spearman correlation coefficient boxplots)

```
> list.statplots <- plotStats(Lib2, Lib2)
> list.statplots[["ppnum"]]
> list.statplots[["pdens"]]
> list.statplots[["phist"]]
```

Build the extended library

If the seed and add-on libraries have good matching quality, we can generate an extended library by integrating them using function *buildSpectraLibPair*.

```
> Lib2_3 <- buildSpectraLibPair(Lib2, Lib3, clean=T,plot=F,
+                                  outputFormat = "peakview",
+                                  outputFile = "Lib2_3.txt")
```

iSwathX, based on SwathXtend, provides two methods of retention time alignment: time-based and hydrophobicity-based. If the retention time correlation between the seed and addon libraries are good (e.g., $R^2 > 0.8$), time-based method is recommended. Otherwise, hydrophobicity-based method can be tried. The hydrophobicity index for peptides can be calculated using SSRCalc(Krokhin 2006). The format of a hydrophobicity index file should include three columns, Sequence, Length and Hydrophobicity. An example of the hydrophobicity index file is included this package. The peptides in this file are all the peptides appearing in the three single assay libraries, i.e., *Lib1.txt*, *Lib2.txt* and *Lib3.txt*.

```
> hydroFile <- paste(system.file("files",package="iSwathX"),
+                                 "hydroIndex.txt",sep="/")
> hydro <- readLibFile(hydroFile, type="hydro")
> head(hydro)
```

```
                 Sequence Length Hydrophobicity
1               AGIQLSPK      8          18.66
2            DASAGIQLSPK     11          22.84
3               DLVEHVAK      8          15.74
4 DPANLPWGSSNVDIAIDSTGVFK     23          47.65
5              FVMGVNEEK      9           24.4
6     GIEEGLMTTVHSLTATQK     18          34.76
```

To build extended libraries using hydrophobicity-based retention time alignment, we can use the following command. The "method" can also be "hydrosequence" which will the combination of hydrophobicity index and the peptide sequence when building the model.

```
> Lib2_3.hydro <- buildSpectraLibPair(libfiles[1], libfiles[2], hydro,
+                                     clean=T,
+                                     nomc=T, nomod=T, plot=F,
+                                     method="hydro",
+                                     outputFormat = "peakview",
+                                     outputFile = "Lib2_3.txt")
```

Export the library

The output of the library format can be “PeakView”, “OpenSwath”, “Skyline”, “Spectronaut”

```
> outputLib(Lib2_3, filename="Lib2_3.txt", format="peakview")
```

Library checking

Function *reliabilityCheckLibrary* compares the extended library with the seed library and checks the peptide and protein coverage. The input is the seed library and extended library files, and the output is a plot displaying the number of peptides and proteins in each library.

```
> reliabilityCheckLibrary(Lib2, Lib2_3)
```

References

Biology IfS (2014) SWATHAtlas.
Gillet LC et al. “Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis”. Molecular and Cellular Proteomics 11. 2012
Krokhin OV. “Sequence-specific retention calculator. Algorithm for peptide retention prediction in ion-pair RP-HPLC”. Analytical chemistry 78:7785-7795. 2006
Wu JX et al. “SWATH mass spectrometry performance using extended peptide MS/MS assay libraries”. Molecular and Cellular Proteomics 15.7 (2016): 2501-2514
Wu JX et al. “Improving protein detection confidence using SWATH mass spectrometry with large peptide reference libraries”. Proteomics. (Under review 2017)
AB Sciex. “MS/MS with Swath Acquisition MicroApp 2.0 User Guide”. 2014
Rost H L., et al. “OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data.” Nature biotechnology 32.3 (2014): 219-223.