Package ‘SwathXtend’

October 12, 2016

Type Package

Title SWATH extended library generation and satistical data analysis

Version 1.0.0

Date 2015-08-04

Author J WU and D Pascovici

Maintainer Jemma Wu <jwu@proteome.org.au>

Depends e1071, openxlsx, VennDiagram, lattice

Description It contains utility functions for integrating spectral libraries for SWATH and statistical data analysis for SWATH generated data.

biocViews Software

License GPL-2

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

applyttest 2
applyttestPep 3
buildSpectraLibPair 4
canonicalFormat 5
checkQuality 6
cleanLib 6
ionCorGS 7
medianNorm 8
mlr 9
mlrGroup 9
mlrrep 10
outputLib 11
plotAll 12
plotDensities 12
plotErrorBarsLines 13

plotRelativeDensities 14
plotRIICor 14
plotRTCor 15
plotRTResd 16
readLibFile 17

Index **18**

`applyttest` *Utility to apply a t-test to all rows of a matrix*

Description

Generate fold change and t-test p-value for all rows of a data matrix

Usage

```
applyttest(mat, Group, doLogs = TRUE, numerator = levels(Group)[1])
```

Arguments

`mat`	Matrix containing data, possibly with missing values
`Group`	Group with two levels of length equal to the number of matrix columns
`doLogs`	True/false, log data before applying test
`numerator`	The level of the group used as numerator for the fold change, by default the first one

Value

Data frame with two values, t-test p-value and fold change.

See Also

`applyttestPep`

Examples

```
mat = matrix(rnorm(600), nrow=100)
mat[1:20, 1:3] = 3+mat[1:20, 1:3] # create some differences
mat[30, 1:3] = NA # and some missing values
mat[100,] = NA


applyttest(mat, Group = rep(c("A", "B"), each=3), doLogs=FALSE)
applyttest(abs(mat), Group = rep(c("A", "B"), each=3), doLogs=TRUE)
```

applyttestPep *Function to apply t-test separately for all peptides of each protein*

Description

Generate fold changes and p-values for each protein (col 1) determined by a number of peptides (col 2).

Usage

```
applyttestPep(peptides, Group, doLogs = TRUE, numerator = levels(as.factor(Group))[1])
```

Arguments

`peptides`	Data frame with two descriptive columns: proteins, peptides, then data in the remaining ncol - 2 columns.
`Group`	Factor describing data membership. Must have two levels, and length = ncol(mat) - 2.
`doLogs`	TRUE/FALSE, log-transform data prior to analysis
`numerator`	The group level used as the numerator in the fold change.

Value

Data frame with rows Protein, fold change and p-value.

See Also

`applyttest`

Examples

```
# make random matrix with first 10 proteins differentially expressed
mat = exp(6+matrix(rnorm(6000), ncol=6))
Protein = sort(paste("P", sample(1:300, 1000, replace=TRUE)))
Peptide = paste("Pep", 1:1000)
for (j in 1:10) mat[Protein == unique(Protein)[j], 4:6] = 3*mat[Protein == unique(Protein)[j], 1:3]

res = applyttestPep(data.frame(Protein, Peptide, mat), rep(c("A", "B"), each=3), numerator="B")
# first 10 proteins should have fold change 3
plot(log(res$FC), -log(res$pval), col=rainbow(2)[1+ as.numeric(1:1000 > 10)])

# add some missing values
mat[5:20,4] = NA
res = applyttestPep(data.frame(Protein, Peptide, mat), rep(c("A", "B"), each=3), numerator="B")
# first 10 proteins should have fold change 3
plot(log(res$FC), -log(res$pval), col=rainbow(2)[1+ as.numeric(1:1000 > 10)])
```

buildSpectraLibPair *Build a spectra library by integrating a pair of spectrum libraries*

Description

Build a spectra library by integrating a pair of spectrum libraries

Usage

```
buildSpectraLibPair(baseLib, extLib, hydroIndex, method = c("time", "hydro",
  "hydrosequence"), includeLength = FALSE, labelBase = NA, labelAddon = NA,
  formatBase = c("peakview", "openswath"), formatExt = c("peakview",
  "openswath"), outputFormat = c("peakview", "openswath"),
  outputFile = "extendedLibrary.txt", plot = FALSE,
  clean = TRUE, merge = TRUE, parseAcc = TRUE, consolidateAccession = TRUE, ...)
```

Arguments

`baseLib`	a base library data frame or file
`extLib`	an external/addon library data frame or file
`hydroIndex`	a data frame or file containing peptide hydrophobicity index
`method`	a character string to specify the RT alignment method. One of "time" (default), "hydro" and "hydrosequence" can be selected.
`includeLength`	a logic value representing if include peptide length as a feature for predicting retention time. Only applicable when method is "hydro".
`labelBase`	a character string to specify the labels of proteins from the base library
`labelAddon`	a character string to specify the labels of proteins from the addon library
`formatBase`	a character string denoting the file format of base library file. One of "peakview" (default) and "opensswath"
`formatExt`	a character string denoting the file format of addon library file. One of "peakview" (default) and "opensswath"
`outputFormat`	a character string denoting the file format of the output integrated library. One of "peakview" (default) and "opensswath"
`outputFile`	A character string to specify the sepctra library created
`plot`	a logic value, representing if plots during processing will be plotted or not
`clean`	a logic value, representing if the input libraries will be cleaned before integration. Default value is True.
`merge`	a logic value, representing if the output will be the merged library (default) or the adjusted add-on library.
`parseAcc`	a logic value, repesenting if the protein accessions will be parsed for consolidation.
`consolidateAccession`	a logic value, representing if the protein accessions will be consolidated to the base libary in the integrated library. Default value is True.
`...`	Additional parameters to pass in.

Value

A data frame of the integrated spectrum library

Examples

```
libfiles <- paste(system.file("files",package="SwathXtend"),
c("Lib2.txt","Lib3.txt"),sep="/")
Lib2_3 <- buildSpectraLibPair(libfiles[1], libfiles[2],
outputFormat="peakview", clean=TRUE, nomod=TRUE, nomc=TRUE)
```

`canonicalFormat` *Standardise a sprectrum library data frame*

Description

Standardise a sprectrum library data frame

Usage

```
canonicalFormat(dat, format = c("peakview", "openswath"))
```

Arguments

`dat`	a data frame of a spectrum library
`format`	a character string, representing the format of the input spectrum library. One of "peakview" (default) and "openswath"

Value

a data frame of the library in canonical format

Examples

```
file <- paste(system.file("files",package="SwathXtend"),"Lib1.txt",sep="/")
dat <- read.delim2(file,sep="\t",stringsAsFactor = FALSE, header=TRUE)
dat <- try(canonicalFormat(dat, format = "peakview"))
```

checkQuality *Checking for the integration quality of two libraries*

Description

Checking for the integration quality of two libraries

Usage

```
checkQuality(datBaseLib, datExtLib, ...)
```

Arguments

`datBaseLib`	a data frame of the base library
`datExtLib`	a data frame of the add-on library
`...`	Additional parameters to pass in

Value

A list of quality indicators, including squared retention time (RT) correlation coefficient, root mean squared errors of RT residuals, and median of relative ion intensity correlation coefficient

Examples

```
libfiles <- paste(system.file("files",package="SwathXtend"),
   c("Lib2.txt","Lib3.txt"),sep="/")
datBaseLib <- readLibFile(libfiles[1])
datExtLib <- readLibFile(libfiles[2])
res <- checkQuality(datBaseLib, datExtLib)
```

cleanLib *Spectrum library cleanining*

Description

Spectrum library cleanining

Usage

```
cleanLib(datLib, clean = TRUE, intensity.cutoff = 5, conf.cutoff = 0.99,
  nomod = TRUE, nomc = FALSE, enz = c("trypsin", "gluc", "chymotrypsin"))
```

Arguments

`datLib`	a data frame for a spectrum library
`clean`	a logic value indicating if the library will be cleaned. Default value is TRUE.
`intensity.cutoff`	A number value to specify cut off for relative intensity of fragment ions. Only ions with intensity higher than the cut off value (default as 5) will be kept.
`conf.cutoff`	A number value to specify cut off for precursor confidence. Only ions with confidence higher than the cut off value (default as 0.99) will be kept.
`nomod`	a logic value, representing if the modified peptides and its fragment ions will be removed. True (default) means will be removed.
`nomc`	a logic value, representing if peptides with miss cleavages are removed. Default vaue is False (not to remove).
`enz`	A character string representing the enzyme which can be one of "trypsin" (defalut), "gluc", or "chymotrypsin"

Value

a data frame of a cleaned spectrum library by the specified criteria

Examples

```
file <- paste(system.file("files",package="SwathXtend"),"Lib1.txt",sep="/")
dat <- read.delim2(file,sep="\t",header=TRUE,stringsAsFactors=FALSE)
dat <- canonicalFormat(dat)
dat <- cleanLib(dat)
```

`ionCorGS` *Gold standard relative ion intensity correlation (spearman)*

Description

This data set gives the relative ion intensity spearman correlation for 2023 peptides as the gold standard for benchmarking the matching quality of two peptide assay libraries.

Usage

```
data(ionCorGS)
```

Format

A vector containing spearman correlation coefficient for 2023 peptides.

Value

a numeric vector

Source

APAF

References

APAF

`medianNorm` *Utility to median normalize a matrix by columns*

Description

Divide appropriately to make all column medians equal to the max median

Usage

```
medianNorm(mat)
```

Arguments

`mat`	Data matrix to normalize; matrix assumed positive

Value

Matrix of same dimensions.

Examples

```
mat = 100+matrix(rnorm(1000), ncol=10)
mat[,10] = mat[,10] + 2
layout(matrix(1:2, nrow=1))
boxplot(mat)
boxplot(medianNorm(mat))

# note: issues when medians close to 0.
```

mlr *Function to implement mlr normalization*

Description

Calculate normalization factor, histogram peak and width at half peak for a vector

Usage

```
mlr(ratio, doplot)
```

Arguments

`ratio`	Vector, typically of log ratios
`doplot`	A logic value, wheter to plot the ratio histograms (FALSE as default)

Value

`nf`	Normalization factor
`peak`	Histogram peak
`wdt`	Width at half peak

References

Find mlr reference.

Examples

```
mlr(rnorm(1000))
# with shift
mlr(0.5 + rnorm(10000))
```

mlrGroup *Function to do mlr normalization for a matrix group*

Description

Do mlr normalization separately for each set of replicates first, then normalize the resulting matrix

Usage

```
mlrGroup(mat, Group)
```

Arguments

`mat`	Data matrix with replicates as columns
`Group`	Factor of length ncol(mat)

Value

Resulting normalized matrix of the same size as the initial one

References

Find reference to mlr paper

See Also

`mlrrep`, `mlr`

Examples

```
res = mlrGroup(iris[,-5], Group=as.factor(c("Sepal", "Sepal", "Petal", "Petal")))

layout(matrix(1:3, nrow=1))
boxplot(log(iris[,-5]), main="Log only")
boxplot(log(medianNorm(iris[,-5])), main="Median")
boxplot(log(res[[1]]), main="MLR")
```

`mlrrep` *Function to do mlr normalizatiopn on a matrix of replicates*

Description

Calculate all pairwise ratios, log-transform them, find the least variable replicate.

Usage

```
mlrrep(mat)
```

Arguments

`mat`	Data matrix with replicates as columns

Value

`mat.norm`	Normalized data matrix; matrix assumed positive
`wdmat`	Square matrix of half peak widths for each ratio of replicates of size ncol(mat)
`nfmat`	Square matrix of normalization factors for each ratio of replicates of size ncol(mat)
`idx`	Index of replicate to be used as denominator yielding smallest widths

See Also

mlr, mlrGroup

Examples

```
# Example using the iris data
mlrrep(iris[,-5])

# random data
mat = exp(matrix(rnorm(1000),ncol=4))
res = mlrrep(mat)
layout(matrix(1:2, nrow=1))
boxplot(log(res$mat.norm))
boxplot(log(mat))
```

---

outputLib *output a spectrum library into a PeakView format file*

Description

output a spectrum library into a PeakView format file

Usage

```
outputLib(dat, filename = "NewLib.txt", format = c("peakview", "openswath"),
  nodup = TRUE)
```

Arguments

dat	A data frame of a spectrum library
filename	A character string for the name of the output.
format	A character string representing the output format. One of "peakview" (default) and "openswath".
nodup	A logic value, indicating if remove duplicated sprectrum (default)

Value

a file with the specified file name (lib.txt as default) will be saved under the current working directory

Examples

```
file <- paste(system.file("files",package="SwathXtend"),"Lib1.txt",sep="/")
dat <- readLibFile(file)
outputLib(dat)
```

`plotAll` *Plot statistical plots for two libraries*

Description

Plot statistical plots for two libraries

Usage

```
plotAll(datBaseLib, datExtLib, file = "allplots.xlsx", ...)
```

Arguments

`datBaseLib`	a data frame for a base spectrum library
`datExtLib`	a data frame for a external spectrum library
`file`	a character string for the output file
`...`	Additional parameters to pass in

Value

a list of two data frames

Examples

```
libfiles <- paste(system.file("files",package="SwathXtend"),
c("Lib2.txt","Lib3.txt"),sep="/")
datBaseLib <- readLibFile(libfiles[1])
datExtLib <- readLibFile(libfiles[2])
res <- plotAll(datBaseLib, datExtLib)
```

`plotDensities` *Utility to do side by side density plots*

Description

Side by side density plots

Usage

```
plotDensities(data, group = rownames(data), xlab = "Log Abundance")
```

Arguments

`data`	Data with samples as columns.
`group`	Group of the same length as the number of columns of data
`xlab`	Label to be printed

Value

No value returned, plotting only

Examples

```
plotDensities(iris[,-5], rep(c("A", "B"), each=2))
```

---

plotErrorBarsLines   *Utility for clustering plots to plot lines and an overall trend*

Description

Prints faint lines for each profile, and a mean/error bars

Usage

```
plotErrorBarsLines(v, barSizes, lines, labels = NULL, col = "blue", ylim, ...)
```

Arguments

`v`	Overall trend, to be printed solid, length n
`barSizes`	Size of the error bars, length n
`lines`	Matrix of n columns, and as many rows as lines
`labels`	Labels to be printed on the x axis, length n
`col`	Colour for main trend line
`ylim`	Can specfy limits so several graphs are on the same scale
`...`	Additional parameters to pass in

Value

No returned value; plot only.

See Also

`help`, ~~~

Examples

```
mat = matrix(rnorm(100), 10)
plotErrorBarsLines(apply(mat,1,FUN=mean), apply(mat,1,FUN=sd),
lines=mat, col="red", main="A random plot", xlab="Some label")
```

`plotRelativeDensities` *Plotting utility to overlay all relative densities*

Description

Overlay all relative densities

Usage

```
plotRelativeDensities(mat, Group = NULL, idx = NULL, main = "Densities")
```

Arguments

`mat`	Matrix with positive entries, samples as columns
`Group`	The factor showing the sample membership, of length ncol(mat)
`idx`	Number between 1:ncol(mat); which sample to use as denominator, first one by default
`main`	Title; optional

Value

Plotting only

Examples

```
mat = matrix(abs(rnorm(50000)), ncol=5)
mat[,5] = mat[,5] + 2

plotRelativeDensities(mat, Group=c(rep("A",4),"B"), idx=1)
```

`plotRIICor` *Plot relative ion intensity correlation of two libraries*

Description

Plot relative ion intensity correlation of two libraries

Usage

```
plotRIICor(dat1, dat2, nomod = FALSE)
```

Arguments

`dat1`	A data frame containing the first spectrum library
`dat2`	A data frame containing the second spectrum library
`nomod`	a logic value, representing if the modified peptides and its fragment ions will be removed. FALSE (default) means not removing.

Value

a data frame of relative ion intensity correlations for all ions

Examples

```
libfiles <- paste(system.file("files",package="SwathXtend"),
   c("Lib2.txt","Lib3.txt"),sep="/")
datBaseLib <- readLibFile(libfiles[1])
datExtLib <- readLibFile(libfiles[2])
plotRIICor(datBaseLib, datExtLib)
```

`plotRTCor` *Plot for retention time correlation of two libraries*

Description

Plot for retention time correlation of two libraries

Usage

```
plotRTCor(dat1, dat2, label1, label2, nomod = FALSE)
```

Arguments

`dat1`	A data frame containing the first spectrum library
`dat2`	A data frame containing the second spectrum library
`label1`	a character string representing the x axis label for plotting
`label2`	a character string representing the y axis label for plotting
`nomod`	a logic value, representing if the modified peptides and its fragment ions will be removed. FALSE (default) means not removing.

Value

retentiont time correlation coefficient

Examples

```
libfiles <- paste(system.file("files",package="SwathXtend"),
   c("Lib2.txt","Lib3.txt"),sep="/")
datBaseLib <- readLibFile(libfiles[1])
datExtLib <- readLibFile(libfiles[2])
plotRTCor(datBaseLib, datExtLib, "Lib2", "Lib5")
```

---

plotRTResd *Plot residuals for retention time prediction of two libraries*

Description

Plot residuals for retention time prediction of two libraries

Usage

```
plotRTResd(dat1, dat2, nomod = FALSE)
```

Arguments

dat1	A data frame containing the first spectrum library
dat2	A data frame containing the second spectrum library
nomod	a logic value, representing if the modified peptides and its fragment ions will be removed. FALSE (default) means not removing.

Value

root mean square error of prediction residuals

Examples

```
libfiles <- paste(system.file("files",package="SwathXtend"),
   c("Lib2.txt","Lib3.txt"),sep="/")
datBaseLib <- readLibFile(libfiles[1])
datExtLib <- readLibFile(libfiles[2])
plotRTResd(datBaseLib, datExtLib)
```

readLibFile *Load a spectrum library into a data frame*

Description

Load a spectrum library into a data frame

Usage

```
readLibFile(file, format = c("peakview", "openswath"), type = c("spectrum",
  "hydro"), clean = TRUE, ...)
```

Arguments

`file`	A file of a spectrum library, in .txt or .csv format, can be .gz files.
`format`	A character string denoting the file format. One of "peakview" (default) and "openswath"
`type`	A character string denoting the file type. One of "spectrum" (default) and "hydro"
`clean`	A logic value, representing if the library will be cleaned.
`...`	Additional parameters to pass in

Value

a data frame of the library with cleaning process

Examples

```
file <- paste(system.file("files",package="SwathXtend"),"Lib1.txt",sep="/")
dat <- readLibFile(file)
```

Index

∗Topic **\textasciitildekwd1**
    applyttest, 2
    applyttestPep, 3
    medianNorm, 8
    mlr, 9
    plotErrorBarsLines, 13
∗Topic **\textasciitildekwd2**
    applyttest, 2
    applyttestPep, 3
    medianNorm, 8
    mlr, 9
    plotErrorBarsLines, 13
∗Topic **datasets**
    ionCorGS, 7

applyttest, 2, *3*
applyttestPep, *2*, 3

buildSpectraLibPair, 4

canonicalFormat, 5
checkQuality, 6
cleanLib, 6

help, *13*

ionCorGS, 7

medianNorm, 8
mlr, 9, *10, 11*
mlrGroup, 9, *11*
mlrrep, *10*, 10

outputLib, 11

plotAll, 12
plotDensities, 12
plotErrorBarsLines, 13
plotRelativeDensities, 14
plotRIICor, 14
plotRTCor, 15
plotRTResd, 16

readLibFile, 17