Package ‘specL’

April 10, 2015

Type Package

Title specL - Prepare Peptide Spectrum Matches for Use in Targeted Proteomics

Version 1.0.0

Author Christian Trachsel <christian.trachsel@fgcz.uzh.ch>, Christian Panse <cp@fgcz.ethz.ch>, Jonas Grossmann <jg@fgcz.ethz.ch>

Maintainer Christian Panse <cp@fgcz.ethz.ch>

Depends R (>= 3.0.2), methods, DBI, RSQLite, seqinr, protViz (>= 0.2.5)

Suggests RUnit, BiocGenerics, BiocStyle, BiocParallel

Description specL provides a function for generating spectra libraries which can be used for MRM SRM MS workflows in proteomics. The package provides a BiblioSpec reader, a function which can add the protein information using a FASTA formatted amino acid file, and an export method for using the created library in the Spectronaut software.

License GPL-3

URL http://www.bioconductor.org/packages/devel/bioc/html/specL.html

Collate read.bibliospec.R genSwathIonLib.R annotateProteinID.R AllGenerics.R specL.R specLSet.R

biocViews MassSpectrometry, Proteomics

LazyData true

R topics documented:

`annotateProteinID` *annotateProteinID*

Description

This function assigns the protein identifier for a list of tandem mass specs having a peptide sequence assigned.

Usage

```
annotateProteinID(data, file = NULL, fasta = read.fasta(file = file,
        as.string = TRUE, seqtype = "AA"), digestPattern = "(([RK])|(^)|(^M))")
```

Arguments

`data`	list of records containing mZ and peptide sequences.
`file`	file name of a FASTA file.
`fasta`	a fasta object as reaturned by the `seqinr::read.fasta(...)` methode.
`digestPattern`	a regex pattern which can be used by the `grep` command. the default regex pattern assumes a tryptic digest.

Details

The protein sequences a read by the `read.fasta` function of the `seqinr` package. The protein identifier is written to the protein `proteinInformation` variable.

If the function is called on a multi core architecture it uses `mclapply`.

It is recommended to load the FASTA file prior to running `annotateProteinID` using `myFASTA <- read.fasta(file = file` instead of providing the FASTA file name to the function.

Value

it returns a list object.

Author(s)

Jonas Grossmann and Christian Panse, 2014

See Also

`?read.fasta` of the `seqinr` package.

http://www.uniprot.org/help/fasta-headers

Examples

```
# annotateProteinID

# our Fasta sequence
  irtFASTAseq <- paste(">zz|ZZ_FGCZCont0260|",
  "iRT_Protein_with_AAAAK_spacers concatenated Biognosys\n",
  "LGGNEQVTRAAAAKGAGSSEPVTGLDAKAAAAKVEATFGVDESNAKAAAAKYILAGVENS",
  "KAAAAKTPVISGGPYEYRAAAAKTPVITGAPYEYRAAAAKDGLDAASYYAPVRAAAAKAD",
  "VTPADFSEWSKAAAAKGTFIIDPGGVIRAAAAKGTFIIDPAAVIRAAAAKLFLQFGAQGS",
  "PFLK\n")

# be realistic, do it from file
  Tfile <- file();  cat(irtFASTAseq, file = Tfile);

#use read.fasta from seqinr
  fasta.irtFASTAseq <-read.fasta(Tfile, as.string=TRUE, seqtype="AA")
  close(Tfile)

#annotate with proteinID
# -> here we find all psms from the one proteinID above
  peptideStd <- specL::annotateProteinID(peptideStd,
  fasta=fasta.irtFASTAseq)

#show indices for all PSMs where we have a proteinInformation
 which(unlist(lapply(peptideStd,
  function(x){nchar(x$proteinInformation)>0})))
```

---

genSwathIonLib *Spectrum library generator for SWATH analysis*

Description

This function generates an ion library for SWATH analysis. It takes a R data object which contains a peaklist. The R data object can be generated using the `read.bibliospec` function.

Usage

```
genSwathIonLib(data,
    data.fit = data,
    mascotIonScoreCutOFF=20,
    proteinIDPattern=,
    max.mZ.Da.error = 0.1,
    ignoreMascotIonScore = TRUE,
    topN = 10,
    fragmentIonMzRange = c(200, 2000),
    fragmentIonRange = c(2,100),
    fragmentIonFUN = .defaultSwathFragmentIon,
    iRT = specL::iRTpeptides,
    AminoAcids = protViz::AA,
    file = NULL)
```

Arguments

`data` data set containing mZ and peptide sequence.

`mascotIonScoreCutOFF`
a value for filtering the specs.

`proteinIDPattern`
a filter for protein.

`file` the output file name.

`max.mZ.Da.error`
the mZ error in Dalton.

`ignoreMascotIonScore`
boolean if mascot score is considered or not.

`topN` returns the most `N` intense fragment ion only.

`fragmentIonMzRange`
mZ range filter of framgent ion.

`fragmentIonRange`
range filter of the number of identified fragment ion set in `fragmentIonTyp`.

`fragmentIonFUN` `function (b, y)` which derives all requested fragment ion out a given tuble of `b` and `y` ion. If the parameter is not specified the method uses an internal function similar as the example below.

`iRT` optional table which contains iRT peptides. If an iRT table is provided (default) a `lm` is applied to normalize the rt in `data`. See also `?iRT`. A necessary condition is that `data` contains at least two iRT peptides.

`AminoAcids` a list containing of 1-letter code and monoisotopic mass of the amino acids. Default uses the `protViz::AA` data set.

`data.fit` data set containing mZ and peptide sequence which is used for normalizing rt using a lineat model `lm(formula = rt ~ aggregateInputRT * fileName, data)`. The rt aggregation for the model uses `median`.

Details

The function is the main contribution of the `specL` package. It generates the spectra library used in a SWATH analysis workflow out of a mass spectrometric measurement.

`genSwathIonLib` uses the core functions `protViz::findNN`, `protViz::fragmentIon`, and `protViz::aa2mass`.

The input is read by using `read.bibliospec` function of this package and passed by the `data` function parameter. If no BiblioSpec files are available also Mascot DAT files can be read using scripts contained in the protViz package exec folder.

The function first appear in the `protViz` 0.1.45 package. It has been removed in `protViz` 0.2.6 to avoid package dependencies.

Value

The output is a data structure containing a `specLSet` object. a list of `specL` objects, a vector of normalized retention times `rt`, and a vector of original retention times `rt.org`.

Author(s)

Christian Panse, Christian Trachsel, and Jonas Grossmann 2012, 2013, 2014

See Also

`vignette(specL)`

Examples

```
    myFragmentIon <- function (b, y) {
        Hydrogen <- 1.007825
        Oxygen <- 15.994915
        Nitrogen <- 14.003074

        b1_ <- (b )
        y1_ <- (y )

        b2_ <- (b + Hydrogen) / 2
        y2_ <- (y + Hydrogen) / 2

        b3_ <- (b + 2 * Hydrogen) / 3
        y3_ <- (y + 2 * Hydrogen) / 3

        return( cbind(b1_, y1_, b2_, y2_, b3_, y3_) )
    }

    L <- genSwathIonLib(data=peptideStd,
        data.fit=peptideStd.redundant,
        fragmentIonFUN=myFragmentIon)

    plot(L)
```

iRTpeptides *iRT peptides - independent retention time peptides*

Description

iRTpeptides data are used for genSwathIonLib rt normalization assuming.

iRTpeptides first appear in the protViz 0.1.45 package. It has been removed in protViz 0.2.10 to avoid package dependencies.

Format

contains a table

Author(s)

Jonas Grossmann and Christian Panse 2013

References

Using iRT, a normalized retention time for more targeted measurement of peptides. Escher C, Reiter L, MacLean B, Ossola R, Herzog F, Chilton J, MacCoss MJ, Rinner O. Source Proteomics. 2012 Apr;12(8):1111-21. doi: 10.1002/pmic.201100463.

Examples

```
 plot(sort(iRTpeptides$rt))

plot(pim<-protViz::parentIonMass(as.character(iRTpeptides$peptide)) ~ iRTpeptides$rt)
```

peptideStd *Peptide standard*

Description

This dataset is a list of a peptide spectrum matches (protein identification result) from two standard runs.

Format

contains a list of peptide spectrum assignments.

Details

These standard runs (LCMS experiments) are routinely run on well maintained instruments. In this case a standard run consits of a digest of the FETUIN_BOVINE protein (400 amol) and iRT peptides.

Author(s)

Christian Panse, Christian Trachsel and Jonas Grossmann 2014

Examples

```
peakplot(peptideStd[[40]]$peptideSequence, peptideStd[[40]])
```

---

plot-methods *Method for Function* `plot` *in Package* **specL**

Description

This methode has no additional arguments.

Value

The method plots on the current device.

Methods

`signature(x = "specL")` Plots the specL determined ions.

`signature(x = "specLSet")` Plots retention time versus retention time.

---

read.bibliospec *BiblioSpec Reader*

Description

This function reads a BiblioSpec generated file and returns a list of tandem mass specs, peptide assignments, retention times, and modifications records.

Usage

```
read.bibliospec(file)
```

Arguments

`file`	the name of the BiblioSpec generated SQLite file.

Details

The function performs a SQL query on the SQLite files generated by bibliospec using the `RSQLite` package. The function is required for generating spec libraries used in a SWATH workflow.

BiblioSpec files are generated by using Skyline.

Value

It returns a list which can be read by the `genSwathIonLib` function and the `protViz::peakplot` function.

Author(s)

Christian Panse, 2014

See Also

`https://skyline.gs.washington.edu/labkey/project/home/software/Skyline/begin.view`

`https://skyline.gs.washington.edu/labkey/project/home/software/BiblioSpec/begin.view`

`http://www.sqlite.org/`

`?SQLite`

Examples

```
read.bibliospec
```

`show-methods` *Methods for Function* `show` *in Package* **specL** ~~

Description

Methods for function `show` in package **specL** ~~ writes specL or specLSet objects to a file or console.

Methods

`signature(x = "specL")` Prints specL object data to the console.

`signature(x = "specLSet")` Prints specL object data to the console.

specL-class *Class* "specL"

Description

This class is used to store, print, and plot the generated results of the package.

Objects from the Class

Objects can be created by calls of the form `new("specL", ...)`.

Slots

`group_id`**:** Object of class `"character"` just an id

`peptide_sequence`**:** Object of class `"character"` AA sequence

`proteinInformation`**:** Object of class `"character"` a string contains the protein identifier.

`q1`**:** Object of class `"numeric"` peptide weight m/Z as measured by the MS device

`q3`**:** Object of class `"numeric"` measured fragment ions.

`q3.in_silico`**:** Object of class `"numeric"` in-silico derived fragment ions.

`decoy`**:** Object of class `"character"` is this a decoy hit? 1 or 0.

`prec_z`**:** Object of class `"numeric"` pre-cursor charge.

`frg_type`**:** Object of class `"character"` fragmenbt ion type, e.g., b or y ion.

`frg_nr`**:** Object of class `"numeric"` fragment ion number

`frg_z`**:** Object of class `"numeric"` fragment ion charge.

`relativeFragmentIntensity`**:** Object of class `"numeric"` percentage base peaks of frament ions.

`irt`**:** Object of class `"numeric"` independent retention time in seconds.

`peptideModSeq`**:** Object of class `"numeric"` a vector contains the mass diff between AA and mod AA.

`mZ.error`**:** Object of class `"numeric"` a string contains the protein identifier.

`filename`**:** Object of class `"character"` a string contains the filename of the ions.

Methods

plot `signature(x = "specL")`: plots the fragment ions of `specL` object.

show `signature(x = "specL")`: shows the content of `specL` object.

write.Spectronaut `signature(x = "specL")`: writes the `specL` object to a ASCII file.

Note

No notes yet.

Author(s)

Christian Panse 2014

See Also

genSwathIonLib

Examples

```
showClass("specL")
```

specLSet-class *Class* "specLSet"

Description

This class is used to store, show, and write generated results of the package.

Objects from the Class

Objects can be created by calls of the form `new("specLSet", ...)`.

Slots

`ionlibrary`**:** A list of `"specL"` objects.

`rt.normalized`**:** A numeric vector of normalized retention time values.

`rt.input`**:** A numeric vector of retention time values.

Methods

show `signature(x = "specLSet")`: shows the object content.

plot `signature(x = "specLSet")`: plots normalized verus input rt.

write.Spectronaut `signature(x = "specLSet")`: writes object to ASCII file.

ionlibrary `signature(x = "specLSet")`: returns a list of `specL` objects.

rt.input `signature(x = "specLSet")`: returns a numeric vector of intput rt values.

rt.normalized `signature(x = "specLSet")`: returns a numeric vector of normalized rt values.

Note

No notes yet.

Author(s)

Christian Panse 2014

See Also

genSwathIonLib

Examples

```
showClass("specL")
showClass("specLSet")
```

---

write.Spectronaut-methods

Methods for Function `write.Spectronaut` *in Package* **specL**

Description

Methods for function `write.Spectronaut` in package **specL** ~~ writes specL objects to a file in a format which can be read by the 'Spectronaut' software. additional arguments are

file A file name. default is `file=spec.txt`.

Methods

`signature(x = "specL")` Prints specL object data to to a file.

`signature(x = "specLSet")` Prints specL object data to to a file.

Index