Package 'Doscheda'

November 13, 2024

Type Package

Title A DownStream Chemo-Proteomics Analysis Pipeline

Version 1.28.0

Author Bruno Contrino, Piero Ricchiuto

Maintainer Bruno Contrino <br1contrino@yahoo.co.uk>

Description Doscheda focuses on quantitative chemoproteomics used to determine protein interaction profiles of small molecules from whole cell or tissue lysates using Mass Spectrometry data. The package provides a shiny application to run the pipeline, several visualisations and a downloadable report of an experiment.

License GPL-3

Depends R (>= 3.4)

Imports methods, drc, stats, httr, jsonlite, reshape2 , vsn, affy, limma, stringr, ggplot2, graphics, grDevices, calibrate, corrgram, gridExtra, DT, shiny, shinydashboard, readxl, prodlim, matrixStats

biocViews Proteomics, Normalization, Preprocessing, MassSpectrometry, QualityControl, DataImport, Regression

Suggests BiocStyle, knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

git_url https://git.bioconductor.org/packages/Doscheda

git_branch RELEASE_3_20

git_last_commit 2b7a2d2

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2024-11-12

Contents

`boxplot,ChemoProtSet-method`

Default boxplot for objects of class ChemoProtSet

Description

Description

Usage

```
## S4 method for signature 'ChemoProtSet'
boxplot(x, ...)
```

Arguments

x	object of class 'ChemoProtSet'
...	other plotting options

Value

boxplot for objects of class ChemoProtSet

ChemoProtSet-class *An S4 class to run the doscheda pipeline*

Description

An S4 class to run the doscheda pipeline

Slots

`input` A data.frame containing the input data

`normData` A data.frame containin a processed and standardised version of the input data

`finalData` A data.frame containing the final data produced by the pipline

`parameters` A list containing all the parameters required to make the pipeline run successfully

`datasets` A list containing other potentially useful datasets

corrPlot *Plot showing correlation between all channels across replicates*

Description

Plot of the correlation between all the channels in the data.

Usage

```
corrPlot(x, ...)

## S4 method for signature 'ChemoProtSet'
corrPlot(x, ...)
```

Arguments

`x`	object of class 'ChemoProtSet'
`...`	corrplot options

Value

correlation plot for objects of class ChemoProtSet

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
corrPlot(ex)
```

densityPlot *Density plot for objects of class ChemoProtSet*

Description

Description

Usage

```
densityPlot(x, rankProteins = FALSE, ...)

## S4 method for signature 'ChemoProtSet'
densityPlot(x, rankProteins = FALSE, ...)
```

Arguments

x	object of class 'ChemoProtSet'
rankProteins	plot a the set of ranked proteins or plot the density of the channels
...	other plot options

Value

density plot for objects of class ChemoProtSet

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
densityPlot(ex)
```

doscheda *Doscheda: A package for Down Stream Chemo-Proteomics Data Analysis*

Description

The Doscheda package provides three categories of important functions: foo, bar and baz.

Foo functions

The foo functions ...

doschedaApp *Run shiny application for DOSCHEDA*

Description

Run a version of the pipline with some extra features and a simple user experience. The application is documented in detail at here

Usage

```
doschedaApp()
```

Value

Launches shiny application

doschedaData *Peptide Intensity data set for Doscheda*

Description

A fabricated data set to run the Doscheda pipeline from peptide intensity.

Usage

```
data(doschedaData)
```

Format

An object of class `data.frame` with 21140 rows and 15 columns.

Examples

```
data(doschedaData)
head(doschedaData)
```

fitModel *Method to fit a model to an object of class 'ChemoProtSet'*

Description

Method to fit a model to an object of class 'ChemoProtSet'

Usage

```
fitModel(x)

## S4 method for signature 'ChemoProtSet'
fitModel(x)
```

Arguments

x	object of class 'ChemoProtSet'

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
channelNames <- c('Abundance..F1..126..Control..REP_1',
'Abundance..F1..127..Sample..REP_1',  'Abundance..F1..128..Sample..REP_1',
 'Abundance..F1..129..Sample..REP_1',  'Abundance..F1..130..Sample..REP_1',
'Abundance..F1..131..Sample..REP_1',  'Abundance..F2..126..Control..REP_2',
'Abundance..F2..127..Sample..REP_2', 'Abundance..F2..128..Sample..REP_2',
'Abundance..F2..129..Sample..REP_2',  'Abundance..F2..130..Sample..REP_2',
'Abundance..F2..131..Sample..REP_2')
ex <- new('ChemoProtSet')
ex<- setParameters(x = ex,chansVal = 6, repsVal = 2,dataTypeStr = 'intensity',
 modelTypeStr = 'linear',PDBool = FALSE,removePepsBool = FALSE,
 incPDofPDBool = FALSE,incGeneFileBool = FALSE,organismStr = 'H.sapiens', pearsonThrshVal = 0.4)
ex<- setData(x = ex, dataFrame = doschedaData, dataChannels = channelNames,
accessionChannel = 'Master.Protein.Accessions',
              sequenceChannel = 'Sequence', qualityChannel = 'Qvality.PEP' )
ex <- removePeptides(ex,removePeps = FALSE)
ex <- runNormalisation(ex)
ex <- fitModel(ex)
ex

ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)

ex
```

getDatasets *Accessor function for the datasets slot.*

Description

Accessor function for the datasets slot of a ChemoProtSet object.

Usage

```
getDatasets(x)

## S4 method for signature 'ChemoProtSet'
getDatasets(x)
```

Arguments

x	object of class ChemoProtSet

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- new('ChemoProtSet')
getDatasets(ex)
```

getFinal *Accessor function for the finalData slot.*

Description

Accessor function for the finalData slot of a ChemoProtSet object.

Usage

```
getFinal(x)

## S4 method for signature 'ChemoProtSet'
getFinal(x)
```

Arguments

x	object of class ChemoProtSet

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- new('ChemoProtSet')
getParameters(ex)
```

---

getInput *Accessor function for the Input*

Description

Accessor function for the Input slot of a ChemoProtSet object.

Usage

```
getInput(x)

## S4 method for signature 'ChemoProtSet'
getInput(x)
```

Arguments

x	object of class ChemoProtSet

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- new('ChemoProtSet')
getInput(ex)
```

getNorm *Accessor function for the normData*

Description

Accessor function for the normData slot of a ChemoProtSet object.

Usage

```
getNorm(x)

## S4 method for signature 'ChemoProtSet'
getNorm(x)
```

Arguments

x	object of class ChemoProtSet

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- new('ChemoProtSet')
getNorm(ex)
```

getParameters *Accessor function for the parameters slot.*

Description

Accessor function for the parameters slot of a ChemoProtSet object.

Usage

```
getParameters(x)

## S4 method for signature 'ChemoProtSet'
getParameters(x)
```

Arguments

x	object of class ChemoProtSet

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- new('ChemoProtSet')
getParameters(ex)
```

---

makeReport *Create report from 'ChemProtSet' object*

Description

Genereate a report that includes several plots and descriptions for an experiment that has been anaylsed using Doscheda

Usage

```
makeReport(x)
```

Arguments

x	Object of class 'ChemoProtSet'

Value

html report of processed 'ChemoProtSet' object

Examples

```
## Not run:
ex<- new('ChemoProtSet')
makeReport(ex)

## End(Not run)
```

meanSdPlot *MeanSd plot for objects of class ChemoProtSet*

Description

Shows the ranked means with a running median calculated with a window size of 10

Usage

```
meanSdPlot(x, ...)

## S4 method for signature 'ChemoProtSet'
meanSdPlot(x, ...)
```

Arguments

x	object of class 'ChemoProtSet'
...	other plot options

Value

meanSd plot for objects of class ChemoProtSet

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
meanSdPlot(ex)
```

pcaPlot *PCA of the main data sets contained in a object of class ChemoProtSet*

Description

Plot of Principal Component Analysis for the first two principal components of the experimental data.

Usage

```
pcaPlot(x, ...)

## S4 method for signature 'ChemoProtSet'
pcaPlot(x, ...)
```

Arguments

x	object of class 'ChemoProtSet'
...	other plot options

Value

PCA plot for objects of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
pcaPlot(ex)
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
pcaPlot(ex)
```

---

plot.ChemoProtSet *Default plot for objects of class ChemoProtSet*

Description

Description

Usage

```
## S3 method for class 'ChemoProtSet'
plot(x, sigmoidCoef = "rb50", ...)
```

Arguments

x	object of class 'ChemoProtSet'
sigmoidCoef	the sigmoidal coeffcient, one of ('difference', 'slope', 'rb50'). Obselete if modelType is 'linear'
...	other plotting options

Value

plot for objects of class ChemoProtSet

processedExample — *Processed Peptide Intensity data set for Doscheda*

Description

A processed fabricated data set to run the Doscheda pipeline from peptide intensity.

Usage

```
data(processedExample)
```

Format

An object of class `ChemoProtSet` of length 1.

Examples

```
data(processedExample)
str(processedExample)
```

removePeptides — *Method to remove peptides from input data of an object of class 'ChemoProtSet'*

Description

Method to remove peptides from input data of an object of class 'ChemoProtSet'

Usage

```
removePeptides(x, changePearson = NA, removePeps = TRUE)

## S4 method for signature 'ChemoProtSet'
removePeptides(x, changePearson = NA,
  removePeps = TRUE)
```

Arguments

`x`	object of class 'ChemoProtSet'
`changePearson`	option to change the peasrson threshold cut-off parameter
`removePeps`	boolean value indicating whether peptide removal should take place

Value

object of class ChemoProtSet

See Also

`DoschedaSet`

Examples

```
## Not run:
channelNames <- c('Abundance..F1..126..Control..REP_1',
'Abundance..F1..127..Sample..REP_1',  'Abundance..F1..128..Sample..REP_1',
 'Abundance..F1..129..Sample..REP_1',  'Abundance..F1..130..Sample..REP_1',
'Abundance..F1..131..Sample..REP_1',  'Abundance..F2..126..Control..REP_2',
'Abundance..F2..127..Sample..REP_2', 'Abundance..F2..128..Sample..REP_2',
'Abundance..F2..129..Sample..REP_2',  'Abundance..F2..130..Sample..REP_2',
'Abundance..F2..131..Sample..REP_2')
ex <- new('ChemoProtSet')
ex<- setParameters(x = ex,chansVal = 6, repsVal = 2,
dataTypeStr = 'intensity', modelTypeStr = 'linear',
 PDBool = FALSE,removePepsBool = FALSE,incPDofPDBool = FALSE,
  incGeneFileBool = FALSE,organismStr = 'H.sapiens',
   pearsonThrshVal = 0.4)

ex<- setData(x = ex, dataFrame = doschedaData,
 dataChannels = channelNames,
 accessionChannel = 'Master.Protein.Accessions',
  sequenceChannel = 'Sequence',
  qualityChannel = 'Qvality.PEP' )
ex <- removePeptides(ex,removePeps = FALSE)
ex

## End(Not run)
```

`replicatePlot` *Plot replicates between concentrations*

Description

Plot of Fold Change between replicate i and replicate j at at a given concentration

Usage

```
replicatePlot(x, conc, repIndex1, repIndex2, ...)

## S4 method for signature 'ChemoProtSet'
replicatePlot(x, conc, repIndex1, repIndex2, ...)
```

Arguments

`x`	object of class 'ChemoProtSet'
`conc`	concentration of channel
`repIndex1`	index of replicate on x axis
`repIndex2`	index of replicate on y axis
`...`	options

Value

Replicate plot for objects of class ChemoProtSet

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
replicatePlot(ex,0,1,2)
```

---

runDoscheda *Wrapper Function to run the entire Doscheda pipeline*

Description

A wrapper for the whole Doscheda pipeline, if users want to avoid using the separate steps.

Usage

```
runDoscheda(dataFrame, dataChannels, accessionChannel, chansVal, repsVal,
  dataTypeStr, modelTypeStr, PDBool = TRUE, removePepsBool = NA,
  incPDofPDBool = FALSE, PDofPDname = NA, incGeneFileBool = FALSE,
  organismStr = "h.sapiens", sigmoidConc = NA, pearsonThrshVal = 0.4,
  uniquePeps = NA, sequenceChannel = NA, qualityChannel = NA,
  pdofpdChannel = NA, incGeneID = FALSE, geneIDFile = NA,
  normType = "loess")
```

Arguments

dataFrame	data.frame of the input data set
dataChannels	column names of dataFrame that correspond to data channels. These should be ordered in the format: rep1_concentration_0, ..., rep1_concentration_n, rep2_concentration_0, ...
accessionChannel	string that is the same as the column name for the protein accessions in dataFrame
chansVal	number of channels / concentrations in experiment
repsVal	number of replicates in experiment
dataTypeStr	string describing the data type of input data set. This can be 'LFC' for log fold-changes, 'FC' for fold-changes and 'intensity' for peptide intensities
modelTypeStr	string describing the type of model applied. This can be 'linear' for a linear model or 'sigmoid' for a sigmoidal model
PDBool	boolean value indicating if the input data is from Proteome Discoverer 2.1 or not
removePepsBool	boolean value indicating if peptide removal will take place. Only valid if input data is peptide intensities
incPDofPDBool	boolean value indicating if the input data contais a pull-down of pull-down colum
PDofPDname	string with the same name as colulmn containing pull-down of pull-down data. NA if this is not applicable
incGeneFileBool	boolean value indicating if the data requires a protein accession to gene ID conversion file

organismStr	string giving the name of organism. the options are: 'H.sapiens', 'D. melanogaster', 'C. elegans', 'R. norvegicus', 'M. musculus'. This is only needed if PDbool is FALSE
sigmoidConc	vector of numerical values for concentrations of channels in the case of a sigmoidal fit
pearsonThrshVal	numerial value between -1 and 1 which determines the cut-off used to discard peptides during peptide removal
uniquePeps	string that is the same as the column name for the number of unique peptides in dataFrame
sequenceChannel	string that is the same as the column name for the peptide sequences in dataFrame
qualityChannel	string that is the same as the column name for the peptide quality score in dataFrame
pdofpdChannel	string that is the same as the column name for the pull-down of pull-down data in dataFrame
incGeneID	boolean value indicating if a protein accession to gene ID file is supplied
geneIDFile	data.frame containing a protein accession to gene ID conversion file
normType	string indicating the type of normalisation that should take place ('loess', 'median', 'none')

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
channelNames <- c('Abundance..F1..126..Control..REP_1',
'Abundance..F1..127..Sample..REP_1',  'Abundance..F1..128..Sample..REP_1',
'Abundance..F1..129..Sample..REP_1',  'Abundance..F1..130..Sample..REP_1',
'Abundance..F1..131..Sample..REP_1',  'Abundance..F2..126..Control..REP_2',
'Abundance..F2..127..Sample..REP_2', 'Abundance..F2..128..Sample..REP_2',
'Abundance..F2..129..Sample..REP_2',  'Abundance..F2..130..Sample..REP_2',
'Abundance..F2..131..Sample..REP_2')

ex <- runDoscheda(dataFrame = doschedaData, dataChannels = channelNames,
chansVal = 6, repsVal = 2,dataTypeStr = 'intensity',
modelTypeStr = 'linear',PDBool = FALSE,removePepsBool = FALSE,
accessionChannel = 'Master.Protein.Accessions',
sequenceChannel = 'Sequence',qualityChannel = 'Qvality.PEP',
incPDofPDBool = FALSE, incGeneFileBool = FALSE,
 organismStr = 'H.sapiens', pearsonThrshVal = 0.4)
```

`runNormalisation` *Method to remove peptides from input data of an object of class 'ChemoProtSet'*

Description

Method to remove peptides from input data of an object of class 'ChemoProtSet'

Usage

```
runNormalisation(x, normalise = "loess")

## S4 method for signature 'ChemoProtSet'
runNormalisation(x, normalise = "loess")
```

Arguments

`x`	object of class 'ChemoProtSet'
`normalise`	string indicating the type of normalisation that should take place ('loess', 'median', 'none')

Value

object of class ChemoProtSet

See Also

`DoschedaSet`

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex
```

`setData` *Method for attaching and standardising data for objects of class 'ChemoProtSet'*

Description

This method will subset the orginal data set into the required columns, standardising column names in the process.

Usage

```
setData(x, dataFrame, dataChannels, accessionChannel, uniquePeps = NA,
  sequenceChannel = NA, qualityChannel = NA, pdofpdChannel = NA,
  incGeneID = FALSE, geneIDFile = NA)

## S4 method for signature 'ChemoProtSet'
setData(x, dataFrame, dataChannels, accessionChannel,
  uniquePeps = NA, sequenceChannel = NA, qualityChannel = NA,
  pdofpdChannel = NA, incGeneID = FALSE, geneIDFile = NA)
```

Arguments

x	object of class 'ChemoProtSet'
dataFrame	data.frame of the input data set
dataChannels	column names of dataFrame that correspond to data channels. These should be ordered in the format: rep1_concentration_0, ..., rep1_concentration_n, rep2_concentration_0, ...
accessionChannel	string that is the same as the column name for the protein accessions in dataFrame
uniquePeps	string that is the same as the column name for the number of unique peptides in dataFrame
sequenceChannel	string that is the same as the column name for the peptide sequences in dataFrame
qualityChannel	string that is the same as the column name for the peptide quality score in dataFrame
pdofpdChannel	string that is the same as the column name for the pull-down of pull-down data in dataFrame
incGeneID	boolean value indicating if a protein accession to gene ID file is supplied
geneIDFile	data.frame containing a protein accession to gene ID conversion file

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
channelNames <- c('Abundance..F1..126..Control..REP_1',
'Abundance..F1..127..Sample..REP_1',  'Abundance..F1..128..Sample..REP_1',
 'Abundance..F1..129..Sample..REP_1',  'Abundance..F1..130..Sample..REP_1',
'Abundance..F1..131..Sample..REP_1',  'Abundance..F2..126..Control..REP_2',
'Abundance..F2..127..Sample..REP_2', 'Abundance..F2..128..Sample..REP_2',
'Abundance..F2..129..Sample..REP_2',  'Abundance..F2..130..Sample..REP_2',
'Abundance..F2..131..Sample..REP_2')

ex <- new('ChemoProtSet')
ex<- setParameters(x = ex,chansVal = 6, repsVal = 2,dataTypeStr = 'intensity',
modelTypeStr = 'linear',PDBool = FALSE,removePepsBool = FALSE,
incPDofPDBool = FALSE,incGeneFileBool = FALSE,organismStr = 'H.sapiens', pearsonThrshVal = 0.4)
```

```
ex<- setData(x = ex, dataFrame = doschedaData, dataChannels = channelNames,
accessionChannel = 'Master.Protein.Accessions',
sequenceChannel = 'Sequence',qualityChannel = 'Qvality.PEP')

ex
```

`setParameters` *Method to set parameters for a ChemoProtSet*

Description

Give the ChemoProtSet object the correct parameters for a given experiment in order to successfully run the pipelin

Usage

```
setParameters(x, chansVal, repsVal, dataTypeStr, modelTypeStr, PDBool = TRUE,
  removePepsBool = NA, incPDofPDBool = FALSE, PDofPDname = NA,
  incGeneFileBool = FALSE, organismStr = "h.sapiens", sigmoidConc = NA,
  pearsonThrshVal = 0.4)

## S4 method for signature 'ChemoProtSet'
setParameters(x, chansVal, repsVal, dataTypeStr,
  modelTypeStr, PDBool = TRUE, removePepsBool = NA, incPDofPDBool = FALSE,
  PDofPDname = NA, incGeneFileBool = FALSE, organismStr = "h.sapiens",
  sigmoidConc = NA, pearsonThrshVal = 0.4)
```

Arguments

`x`	object of class 'ChemoProtSet'
`chansVal`	number of channels / concentrations in experiment
`repsVal`	number of replicates in experiment
`dataTypeStr`	string describing the data type of input data set. This can be 'LFC' for log fold-changes, 'FC' for fold-changes and 'intensity' for peptide intensities
`modelTypeStr`	string describing the type of model applied. This can be 'linear' for a linear model or 'sigmoid' for a sigmoidal model
`PDBool`	boolean value indicating if the input data is from Proteome Discoverer 2.1 or not
`removePepsBool`	boolean value indicating if peptide removal will take place. Only valid if input data is peptide intensities
`incPDofPDBool`	boolean value indicating if the input data contais a pull-down of pull-down colum
`PDofPDname`	string with the same name as colulmn containing pull-down of pull-down data. NA if this is not applicable
`incGeneFileBool`	boolean value indicating if the data requires a protein accession to gene ID conversion file

organismStr — string giving the name of organism. the options are: 'H.sapiens', 'D. melanogaster', 'C. elegans', 'R. norvegicus', 'M. musculus'. This is only needed if PDbool is FALSE

sigmoidConc — vector of numerical values for concentrations of channels in the case of a sigmoidal fit

pearsonThrshVal — numerial value between -1 and 1 which determines the cut-off used to discard peptides during peptide removal

Value

object of class ChemoProtSet

See Also

DoschedaSet

Examples

```
channelNames <- c('Abundance..F1..126..Control..REP_1',
'Abundance..F1..127..Sample..REP_1',  'Abundance..F1..128..Sample..REP_1',
'Abundance..F1..129..Sample..REP_1',  'Abundance..F1..130..Sample..REP_1',
'Abundance..F1..131..Sample..REP_1',  'Abundance..F2..126..Control..REP_2',
'Abundance..F2..127..Sample..REP_2', 'Abundance..F2..128..Sample..REP_2',
'Abundance..F2..129..Sample..REP_2',  'Abundance..F2..130..Sample..REP_2',
'Abundance..F2..131..Sample..REP_2')

ex <- new('ChemoProtSet')
ex<- setParameters(x = ex,chansVal = 6, repsVal = 2,dataTypeStr = 'intensity',
modelTypeStr = 'linear',PDBool = FALSE, removePepsBool = FALSE,
incPDofPDBool = FALSE, incGeneFileBool = FALSE,
organismStr = 'H.sapiens', pearsonThrshVal = 0.4)

ex
```

---

volcanoPlot *Volcano plot for objects of class ChemoProtSet*

Description

Volcano plots designed to be run on objects of class 'ChemoProtSet' when a linear model has been applied.

Usage

```
volcanoPlot(x, coefficient = "slope", avExprs = 0.2, pVal = 0.05, ...)

## S4 method for signature 'ChemoProtSet'
volcanoPlot(x, coefficient = "slope",
  avExprs = 0.2, pVal = 0.05, ...)
```

Arguments

x	object of class 'ChemoProtSet'
coefficient	coefficient of linear model to be plotted ('slope','intercept','quadratic')
avExprs	average expression cutoff
pVal	p-value cut-off
...	other plotting options

Value

volcano plot for objects of class ChemoProtSet

See Also

DoschedaSet

Examples

```
ex <- processedExample
ex <- runNormalisation(ex)
ex <- fitModel(ex)
volcanoPlot(ex)
```

Index