

# Package ‘Doscheda’

May 15, 2024

**Type** Package

**Title** A DownStream Chemo-Proteomics Analysis Pipeline

**Version** 1.27.0

**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** devel

**git\_last\_commit** 34b085c

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.20

**Date/Publication** 2024-05-15

## Contents

boxplot,ChemoProtSet-method . . . . .	2
ChemoProtSet-class . . . . .	3
corrPlot . . . . .	3
densityPlot . . . . .	4
doscheda . . . . .	5
doschedaApp . . . . .	5
doschedaData . . . . .	5
fitModel . . . . .	6
getDatasets . . . . .	7
getFinal . . . . .	8
getInput . . . . .	8
getNorm . . . . .	9
getParameters . . . . .	10
makeReport . . . . .	10
meanSdPlot . . . . .	11
pcaPlot . . . . .	12
plot.ChemoProtSet . . . . .	13
processedExample . . . . .	13
removePeptides . . . . .	14
replicatePlot . . . . .	15
runDoscheda . . . . .	16
runNormalisation . . . . .	18
setData . . . . .	18
setParameters . . . . .	20
volcanoPlot . . . . .	22
<b>Index</b>	<b>23</b>

---

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 pipeline  
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	<i>Density plot for objects of class ChemoProtSet</i>
-------------	---

---

## 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	<i>Doscheda: A package for Down Stream Chemo-Proteomics Data Analysis</i>
----------	---

---

**Description**

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

**Foo functions**

The foo functions ...

---

doschedaApp	<i>Run shiny application for DOSCHEDA</i>
-------------	---

---

**Description**

Run a version of the pipeline with some extra features and a simple user experience. The application is documented in detail at [here](#)

**Usage**

```
doschedaApp()
```

**Value**

Launches shiny application

---

doschedaData	<i>Peptide Intensity data set for Doscheda</i>
--------------	--

---

**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 = 'Quality.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)
```



**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 coefficient, one of ('difference', 'slope', 'rb50'). Obsolete 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	<i>Method to remove peptides from input data of an object of class 'ChemoProtSet'</i>
----------------	---

---

### 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 pearson 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 = 'Quality.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 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	<i>Wrapper Function to run the entire Doscheda pipeline</i>
-------------	---

---

### 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 contains a pull-down of pull-down column
PDofPDname	string with the same name as column 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	numerical 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 = 'Quality.PEP',
incPDofPDBool = FALSE, incGeneFileBool = FALSE,
organismStr = 'H.sapiens', pearsonThrshVal = 0.4)
```

---

runNormalisation	<i>Method to remove peptides from input data of an object of class 'ChemoProtSet'</i>
------------------	---

---

### 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	<i>Method for attaching and standardising data for objects of class 'ChemoProtSet'</i>
---------	--

---

### Description

This method will subset the original 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 = 'Quality.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 pipeline

### 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 contains a pull-down of pull-down column
PDofPDname	string with the same name as column 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	numerical 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**

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

- \* **datasets**
  - doschedaData, 5
  - processedExample, 13
- boxplot, ChemoProtSet-method, 2
- ChemoProtSet-class, 3
- corrPlot, 3
- corrPlot, ANY, ANY-method (corrPlot), 3
- corrPlot, ChemoProtSet-method (corrPlot), 3
- densityPlot, 4
- densityPlot, ANY, ANY-method (densityPlot), 4
- densityPlot, ChemoProtSet-method (densityPlot), 4
- doscheda, 5
- doscheda-package (doscheda), 5
- doschedaApp, 5
- doschedaData, 5
- DoschedaSet, 6–10, 12, 14, 17–19, 21, 22
- DoschedaSet (ChemoProtSet-class), 3
- fitModel, 6
- fitModel, ANY, ANY-method (fitModel), 6
- fitModel, ChemoProtSet-method (fitModel), 6
- getDatasets, 7
- getDatasets, ANY, ANY-method (getDatasets), 7
- getDatasets, ChemoProtSet-method (getDatasets), 7
- getFinal, 8
- getFinal, ANY, ANY-method (getFinal), 8
- getFinal, ChemoProtSet-method (getFinal), 8
- getInput, 8
- getInput, ANY, ANY-method (getInput), 8
- getInput, ChemoProtSet-method (getInput), 8
- getNorm, 9
- getNorm, ANY, ANY-method (getNorm), 9
- getNorm, ChemoProtSet-method (getNorm), 9
- getParameters, 10
- getParameters, ANY, ANY-method (getParameters), 10
- getParameters, ChemoProtSet-method (getParameters), 10
- makeReport, 10
- meanSdPlot, 11
- meanSdPlot, ANY, ANY-method (meanSdPlot), 11
- meanSdPlot, ChemoProtSet-method (meanSdPlot), 11
- pcaPlot, 12
- pcaPlot, ANY, ANY-method (pcaPlot), 12
- pcaPlot, ChemoProtSet-method (pcaPlot), 12
- plot.ChemoProtSet, 13
- processedExample, 13
- removePeptides, 14
- removePeptides, ANY, ANY-method (removePeptides), 14
- removePeptides, ChemoProtSet-method (removePeptides), 14
- replicatePlot, 15
- replicatePlot, ANY, ANY-method (replicatePlot), 15
- replicatePlot, ChemoProtSet-method (replicatePlot), 15
- runDoscheda, 16
- runNormalisation, 18
- runNormalisation, ANY, ANY-method (runNormalisation), 18

runNormalisation,ChemoProtSet-method  
(runNormalisation), [18](#)

setData, [18](#)

setData,ANY,ANY-method (setData), [18](#)

setData,ChemoProtSet-method (setData),  
[18](#)

setParameters, [20](#)

setParameters,ANY,ANY-method  
(setParameters), [20](#)

setParameters,ChemoProtSet-method  
(setParameters), [20](#)

volcanoPlot, [22](#)

volcanoPlot,ANY,ANY-method  
(volcanoPlot), [22](#)

volcanoPlot,ChemoProtSet-method  
(volcanoPlot), [22](#)