

# Package ‘SafeQuant’

March 13, 2017

**Type** Package

**Title** A Toolbox for the Analysis of Proteomics Data

**Version** 2.3.2

**Date** 2017-02-01

**Author** Erik Ahrne

**Maintainer** Erik Ahrne <erik.ahrne@unibas.ch>

**Description** Tools for the statistical analysis and visualization of (relative and absolute) quantitative (LFQ,TMT,HRM) Proteomics data.

**Imports** limma,

gplots,  
seqinr,  
corrplot,  
optparse,  
data.table,  
epiR,  
Biobase,  
ggplot2,  
magrittr,

**License** GPL-3

**RoxygenNote** 5.0.1

## R topics documented:

addIdQvalues . . . . .	3
addScaffoldPTMFAnnotations . . . . .	4
barplotMSSignal . . . . .	5
COLORS . . . . .	6
createExpDesign . . . . .	6
createExpressionDataset . . . . .	7
createPairedExpDesign . . . . .	8
cvBoxplot . . . . .	9
dotProduct . . . . .	10
expDesignTagToExpDesign . . . . .	10
export . . . . .	11
getAAProteinCoordinates . . . . .	12
getAllCV . . . . .	13
getAllDotProduct . . . . .	14

getAllEBayes . . . . .	14
getBaselineIntensity . . . . .	15
getCV . . . . .	16
getExpDesignProgenesisCsv . . . . .	17
getGlobalNormFactors . . . . .	18
getIBAQset . . . . .	18
getIdLevelQvals . . . . .	19
getImpuritiesMatrix . . . . .	20
getIntSumPerProtein . . . . .	21
getKinaseFreq . . . . .	22
getKinases . . . . .	22
getLOD . . . . .	23
getLoocvFoldError . . . . .	24
getMaxIndex . . . . .	24
getMeanCenteredRange . . . . .	25
getModifProteinCoordinates . . . . .	25
getMotifFreq . . . . .	26
getMotifX . . . . .	27
getNbDetectablePeptides . . . . .	28
getNbMisCleavages . . . . .	28
getNbPeptidesPerProtein . . . . .	29
getNbSpectraPerProtein . . . . .	30
getPeptides . . . . .	30
getRatios . . . . .	31
getRTNormFactors . . . . .	32
getScoreCutOff . . . . .	33
getSignalPerCondition . . . . .	33
getTopX . . . . .	34
getUserOptions . . . . .	35
ggDilutionCurve . . . . .	36
globalNormalize . . . . .	36
hClustHeatMap . . . . .	37
isCon . . . . .	38
isDecoy . . . . .	39
isStrippedACs . . . . .	39
kinaseMotif . . . . .	40
maPlotSQ . . . . .	41
missinValueBarplot . . . . .	41
option_list . . . . .	42
pairsAnnot . . . . .	43
parseMaxQuantProteinGroupTxt . . . . .	44
parseProgenesisFeatureCsv . . . . .	45
parseProgenesisPeptideMeasurementCsv . . . . .	46
parseProgenesisProteinCsv . . . . .	47
parseScaffoldPTMReport . . . . .	48
parseScaffoldRawFile . . . . .	48
perFeatureNormalization . . . . .	49
plotAbsEstCalibrationCurve . . . . .	50
plotAdjustedVsNonAdjustedRatio . . . . .	51
plotExpDesign . . . . .	52
plotIdScoreVsFDR . . . . .	52
plotLogo . . . . .	53

plotMSSignalDistributions . . . . .	54
plotNbIdentificationsVsRT . . . . .	55
plotNbValidDeFeaturesPerFDR . . . . .	55
plotPrecMassErrorDistrib . . . . .	56
plotPrecMassErrorVsScore . . . . .	57
plotQValueVsPValue . . . . .	58
plotROC . . . . .	58
plotRTNorm . . . . .	59
plotRTNormSummary . . . . .	60
plotScoreDistrib . . . . .	61
plotVolcano . . . . .	62
plotXYDensity . . . . .	62
purityCorrectTMT . . . . .	63
removeOutliers . . . . .	64
rollUp . . . . .	65
rtNormalize . . . . .	66
safeQuantAnalysis . . . . .	67
setNbPeptidesPerProtein . . . . .	67
setNbSpectraPerProtein . . . . .	68
sqNormalize . . . . .	69
standardise . . . . .	70
stripACs . . . . .	70
<b>Index</b>	<b>72</b>

---

addIdQvalues	<i>Add identification leve q-values to ExpressionSet (calculated based on target-decoy score distribution)</i>
--------------	----------------------------------------------------------------------------------------------------------------

---

## Description

Add identification leve q-values to ExpressionSet (calculated based on target-decoy score distribution)

## Usage

```
addIdQvalues(eset = eset)
```

## Arguments

eset	ExpressionSet
------	---------------

## Details

if ptm column is part if the ExpressionSet q-values are calculated seperately for modified and non-modified features

No details

## Value

ExpressionSet object

**Note**

No note

**See Also**

[getIdLevelQvals](#)

**Examples**

```
print("No examples")
```

---

addScaffoldPTMFAnnotations

*Add scaffold ptm annotaitons to tmt experiment*

---

**Description**

Add scaffold ptm annotaitons to tmt experiment

**Usage**

```
addScaffoldPTMFAnnotations(eset, file)
```

**Arguments**

eset	ExpressionSet
file	path to Scaffold file

**Value**

ExpressionSet object

**Note**

No note

**References**

No references

**Examples**

```
print("No examples")
```

---

barplotMSSignal	<i>Barplot of ms-signal per column</i>
-----------------	----------------------------------------

---

## Description

Barplot of ms-signal per column

## Usage

```
barplotMSSignal(eset,  
  col = as.character(.getConditionColors(eset)[pData(eset)$condition, ]),  
  method = c("sum", "sharedSignal"), cex.lab = 1.25, cex.axis = 1.25,  
  cex.names = 0.9, labels = rownames(pData(eset)), ...)
```

## Arguments

eset	expressionSet
col	default condition colors
method	c("median","sum","sharedSignal")
cex.lab	default 1.25
cex.axis	default 1.25
cex.names	default 0.9
labels	labels
...	see plot

## Details

No details

## Note

No note

## References

NA

## Examples

```
print("No examples")
```

---

COLORS	<i>color vector</i>
--------	---------------------

---

**Description**

color vector

**Usage**

COLORS

**Format**

An object of class character of length 668.

---

createExpDesign	<i>Create Experimental Design</i>
-----------------	-----------------------------------

---

**Description**

Create Experimental Design

**Usage**

createExpDesign(tag, nbPlex)

**Arguments**

tag	user input tag e.g. 1,2,3:4,5,6 indicating two condition with 3 reps each
nbPlex	tmt 6 or 10 plex

**Details**

The first listed condition is always the control condition  
No details

**Value**

expDesign data.frame

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`createExpressionDataset`*Create ExpressionSet object*

---

**Description**

Create ExpressionSet object

**Usage**

```
createExpressionDataset(expressionMatrix = expressionMatrix,  
  expDesign = expDesign, featureAnnotations = featureAnnotations)
```

**Arguments**

<code>expressionMatrix</code>	matrix of expression signals per feature and sample
<code>expDesign</code>	experimental design data.frame
<code>featureAnnotations</code>	data.frame including e.g: Protein Description, Id score etc.

**Details**

No details

**Value**

ExpressionSet object

**Note**

No note

**References**

NA

**See Also**

[ExpressionSet](#)

**Examples**

```
print("No examples")
```

---

createPairedExpDesign    *Create Paired Expdesign*

---

## Description

Create Paired Expdesign

## Usage

```
createPairedExpDesign(eset)
```

## Arguments

eset	ExpressionSet
------	---------------

## Details

Add subject colum to phenoData design data.frame

## Value

ExpressionSet object

## Note

No note

## References

NA

## See Also

[ExpressionSet](#)

## Examples

```
print("No examples")
```



---

cvBoxplot	<i>C.V. boxplot</i>
-----------	---------------------

---

**Description**

C.V. boxplot

**Usage**

```
cvBoxplot(eset,  
  col = as.character(.getConditionColors(eset)[unique(pData(eset)$condition),  
  ]), cex.names = 0.9, cex.axis = 1.25, cex.lab = 1.25,  
  ylab = "C.V. (%)", ...)
```

**Arguments**

eset	ExpressionSet
col	col
cex.names	default 0.9
cex.axis	default 1.25
cex.lab	default 1.25
ylab	C.V.
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`dotProduct`*Return dotProduct of two vectors*

---

**Description**

Return dotProduct of two vectors

**Usage**

```
dotProduct(u, v, norm = F)
```

**Arguments**

<code>u</code>	vector 1
<code>v</code>	vector 2
<code>norm</code>	dp TRUE/FALSE

**Value**

dp

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`expDesignTagToExpDesign`*Create experimental design data.frame from user input string*

---

**Description**

Create experimental design data.frame from user input string

**Usage**

```
expDesignTagToExpDesign(tag, expDesignDefault)
```

**Arguments**

<code>tag</code>	tag
<code>expDesignDefault</code>	data.frame

**Details**

tag: 1,2:3:4,5,6 condition isControl 1 Condition 1 TRUE 2 Condition 1 TRUE 3 Condition 1 TRUE  
4 Condition 2 FALSE 5 Condition 2 FALSE 6 Condition 2 FALSE

**Value**

data.frame describing experimental design

**Note**

No note

**References**

NA

**Examples**

print("No examples")

---

export	<i>Export content of safeQuantAnalysis object</i>
--------	---------------------------------------------------

---

**Description**

Export content of safeQuantAnalysis object

**Usage**

export(sqa, nbRows = nrow(sqa\$pValue), file = NA)

**Arguments**

sqa	safeQuantAnalysis object
nbRows	Number of rows to export. Features are ordred by increasing minimal p.value
file	file path

**Details**

NA

**Note**

No note

**References**

NA

**See Also**

[safeQuantAnalysis](#)

**Examples**

```
print("No examples")
```

---

```
getAAProteinCoordinates
```

*Get amino acid coordinates on protein*

---

**Description**

Get amino acid coordinates on protein

**Usage**

```
getAAProteinCoordinates(peptideSeq, proteinSeq, aaRegExpr = "[STY]")
```

**Arguments**

peptideSeq	peptide sequence
proteinSeq	protein sequence
aaRegExpr	target AA reg exp

**Details**

NA

**Value**

vector of protein coordinates (mmodification residue number)

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getAllCV	<i>Calculate Coefficient of Variance per feature (Relative standard Deviation) per Condition</i>
----------	--------------------------------------------------------------------------------------------------

---

**Description**

Calculate Coefficient of Variance per feature (Relative standard Deviation) per Condition

**Usage**

```
getAllCV(eset)
```

**Arguments**

eset	ExpressionSet
------	---------------

**Details**

$CV = sd / mean$

**Value**

data.frame of CVs per condition

**Note**

No note

**References**

NA

**See Also**

[getCV](#)

**Examples**

```
print("No examples")
```

---

getAllDotProduct	<i>Return dotProducts to most transition intensities of most intense runs</i>
------------------	-------------------------------------------------------------------------------

---

**Description**

Return dotProducts to most transition intensities of most intense runs

**Usage**

```
getAllDotProduct(eset, nbRefRuns = 4)
```

**Arguments**

eset	ExpressionSet
nbRefRuns	(default top 4)

**Value**

dp

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getAllEBayes	<i>Perform statistical test (mderated t-test), comparing all case to control</i>
--------------	----------------------------------------------------------------------------------

---

**Description**

Perform statistical test (mderated t-test), comparing all case to control

**Usage**

```
getAllEBayes(eset = eset, adjust = F, log = T, method = "pairwise",
  adjustFilter = matrix(F, nrow = nrow(eset), ncol =
    length(levels(pData(eset)$condition)) - 1))
```

**Arguments**

eset	ExpressionSet
adjust	TRUE/FALSE adjust for multiple testing using Benjamini & Hochberg (1995) method
log	T/F log-transform expression values
method	c("all","pairwise")
adjustFilter	matrix T/F do not adjust for multiple testing

**Details**

No details

**Value**

ExpressionSet object

**Note**

No note

**References**

Empirical Bayes method, Smyth (2004), <http://www.ncbi.nlm.nih.gov/pubmed/16646809>

**See Also**

[eBayes](#)

**Examples**

```
print("No examples")
```

---

getBaselineIntensity    *Get signal at zscore x (x standard deviations below mean)*

---

**Description**

Get signal at zscore x (x standard deviations below mean)

**Usage**

```
getBaselineIntensity(intensities, promille = 5)
```

**Arguments**

intensities	refrence run signals
promille	baseline value set as specified promille

**Value**

baseline value

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getCV	<i>Calculate Coefficient of Variance per feature (Relative standard Deviation)</i>
-------	------------------------------------------------------------------------------------

---

**Description**

Calculate Coefficient of Variance per feature (Relative standard Deviation)

**Usage**

```
getCV(data)
```

**Arguments**

data	data.frame of replicate signals
------	---------------------------------

**Details**

$CV = sd / mean$

**Value**

vector of CVs

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```



---

`getExpDesignProgenesisCsv`*Parse Experimental Design from Progenesis Csv Export*

---

**Description**

Parse Experimental Design from Progenesis Csv Export

**Usage**

```
getExpDesignProgenesisCsv(file,  
  expressionColIndices = .getProgenesisCsvExpressionColIndices(file))
```

**Arguments**

<code>file</code>	path to progenesis csv file
<code>expressionColIndices</code>	default <code>.getProgenesisCsvExpressionColIndices(file)</code>

**Details**

No details

**Value**

data.frame describing experimental design

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getGlobalNormFactors	<i>Get normalization factors. calculated as summed/median signal per run (column) over summed/median of first run.</i>
----------------------	------------------------------------------------------------------------------------------------------------------------

---

**Description**

Get normalization factors. calculated as summed/median signal per run (column) over summed/median of first run.

**Usage**

```
getGlobalNormFactors(eset, method = "median")
```

**Arguments**

eset	ExpressionSet
method	c("sum","median")

**Details**

No details

**Value**

vector of normalization factors

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getIBAQEset	<i>Calculate intensity-based absolute-protein-quantification (iBAQ) metric per protein</i>
-------------	--------------------------------------------------------------------------------------------

---

**Description**

Calculate intensity-based absolute-protein-quantification (iBAQ) metric per protein

**Usage**

```
getIBAQEset(eset, proteinDB = NA, peptideLength = c(5, 36),
  nbMiscleavages = 0, proteaseRegExp = .getProteaseRegExp("trypsin"))
```

**Arguments**

eset	protein level ExpressionSet
proteinDB	list protein sequences
peptideLength	peptide length interval (to get number of peptides used for normalization)
nbMiscleavages	number of mis-cleavages allowed when digesting protein sequences in silico (to get number of peptides used for normalization)
proteaseRegExp	protease Reg Exp cleavage rule

**Details**

No details

**Value**

ExpressionSet

**Note**

No note

**References**

Global quantification of mammalian gene expression control, Schwanhauser (2011), <http://www.ncbi.nlm.nih.gov/pubmed/21593866>, Critical assessment of proteome-wide label-free absolute abundance estimation strategies. Ahrne (2013), <http://www.ncbi.nlm.nih.gov/pubmed/23794183>

**Examples**

```
print("No examples")
```

---

getIdLevelQvals	<i>Calculates identification level q-values based on target-decoy score distributions</i>
-----------------	-------------------------------------------------------------------------------------------

---

**Description**

Calculates identification level q-values based on target-decoy score distributions

**Usage**

```
getIdLevelQvals(scores, isDecoy)
```

**Arguments**

scores	peptide/protein identificationscore
isDecoy	vector of TRUE/FALSE

**Details**

$q\text{-value} = (\text{Nb. Decoy Entries at idScore Threshold } S^*) / (\text{Nb. Target Entries at idScore Threshold } S)$ . (\* idScore  $\geq S$ )

**Value**

vector of q.values

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getImpuritiesMatrix	<i>Get Thermo TMT impurity matrix</i>
---------------------	---------------------------------------

---

**Description**

Get Thermo TMT impurity matrix

**Usage**

```
getImpuritiesMatrix(plexNb = 6)
```

**Arguments**

plexNb	integer, 6 or 10 plex
--------	-----------------------

**Details**

No details

**Value**

impurity matrix matrix

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getIntSumPerProtein	<i>Sum up raw intensities per protein and channel. keep track of number of summed spectra and unique peptides</i>
---------------------	-------------------------------------------------------------------------------------------------------------------

---

**Description**

Sum up raw intensities per protein and channel. keep track of number of summed spectra and unique peptides

**Usage**

```
getIntSumPerProtein(intData, proteinACs, peptides, minNbPeptPerProt = 1)
```

**Arguments**

intData	data.frame of intensities per channel
proteinACs	vector of protein accession numbers
peptides	vector of peptide sequences
minNbPeptPerProt	minimal number of peptides per protein

**Details**

NA  
No details

**Value**

list containing 3 objects 1) data.frame of channel intensities per protein ac, 2) vector listing number of summed spectra per protein, 3) vector listing number of summed peptides per protein

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getKinaseFreq	<i>Get kinase matching frequency of each phospho peptide subsequence</i>
---------------	--------------------------------------------------------------------------

---

**Description**

Get kinase matching frequency of each phospho peptide subsequence

**Usage**

```
getKinaseFreq(phosphoSeqs)
```

**Arguments**

phosphoSeqs      vector of phospho peptide sub sequences 'PARVVRpSRREEEE'

**Value**

ExpressionSet object

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getKinases	<i>Get all kinases matching phospho peptide sub sequence</i>
------------	--------------------------------------------------------------

---

**Description**

Get all kinases matching phospho peptide sub sequence

**Usage**

```
getKinases(phosphoSeq)
```

**Arguments**

phosphoSeq      scalar peptide sub sequence 'PARVVRpSRREEEE'

**Value**

ExpressionSet object

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getLOD	<i>Return dilution curve limit of detection</i>
--------	-------------------------------------------------

---

**Description**

Return dilution curve limit of detection

**Usage**

```
getLOD(dCurve, method = "blank")
```

**Arguments**

dCurve	data.frame
method	c("blank","low")

**Value**

lod

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getLoocvFoldError	<i>Leave-One-Out Cross Validate Qunatification Model</i>
-------------------	----------------------------------------------------------

---

**Description**

Leave-One-Out Cross Validate Qunatification Model

**Usage**

```
getLoocvFoldError(df)
```

**Arguments**

df	data.frame of two columns 1) "signal" - ms metric 2) "cpc" absolute quantity
----	------------------------------------------------------------------------------

**Details**

No details

**Value**

data.frame of fold errors per (left-out) protein

**Note**

No note

**References**

NA

**See Also**

NA

**Examples**

```
print("No examples")
```

---

getMaxIndex	<i>get index of max in vecotr of numeric values</i>
-------------	-----------------------------------------------------

---

**Description**

get index of max in vecotr of numeric values

**Usage**

```
getMaxIndex(v)
```

**Arguments**

v	vector
---	--------



---

getMeanCenteredRange    *Get modification coordinates on protein*

---

**Description**

Get modification coordinates on protein

**Usage**

```
getMeanCenteredRange(d, nbSd = 4)
```

**Arguments**

d	numeric vector
nbSd	range spanning number of sd frmo mean

**Details**

NA

**Value**

vector range boundries

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getModifProteinCoordinates  
                          *Get modification coordinates on protein*

---

**Description**

Get modification coordinates on protein

**Usage**

```
getModifProteinCoordinates(modifAnnot, peptideSeq, proteinSeq, format = 1)
```

**Arguments**

modifAnnot	modification as annotated by progenesis. E.g. '[15] Phospho (ST) [30] Phospho (ST)'
peptideSeq	peptide sequence
proteinSeq	protein sequence
format	c(1,2) 1. progenesis 2. scaffold

**Details**

NA

**Value**

vector of protein coordinates (mmodification residue number)

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getMotifFreq	<i>Get motif matching frequency of each phospho peptide subsequences</i>
--------------	--------------------------------------------------------------------------

---

**Description**

Get motif matching frequency of each phospho peptide subsequences

**Usage**

```
getMotifFreq(phosphoSeqs)
```

**Arguments**

phosphoSeqs      vector of phospho peptide subsequences 'PARVVRpSRREEEEE'

**Value**

ExpressionSet object

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getMotifX	<i>Create motif-x peptide annotation</i>
-----------	------------------------------------------

---

**Description**

Create motif-x peptide annotation

**Usage**

```
getMotifX(modifPos, peptide, proteinSeq, motifLength = 4)
```

**Arguments**

modifPos	vector positions
peptide	peptide sequence
proteinSeq	protein sequence
motifLength	motif flanking sequence

**Details**

motif-x example PGDYS\*TTPG

**Value**

vector of motifs

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`getNbDetectablePeptides`*Get number peptides passing defined length criteria*

---

**Description**

Get number peptides passing defined length criteria

**Usage**

```
getNbDetectablePeptides(peptides, peptideLength = c(5, 36))
```

**Arguments**

<code>peptides</code>	list of peptides
<code>peptideLength</code>	vector of two integers defining peptide length range

**Details**

No details

**Value**

integer corresponding to number of detectable peptides

**Note**

No note

**Examples**

```
print("No examples")
```

---

`getNbMisCleavages`*Get number of mis-cleavages perp peptide*

---

**Description**

Get number of mis-cleavages perp peptide

**Usage**

```
getNbMisCleavages(peptide, protease = "trypsin")
```

**Arguments**

<code>peptide</code>	character vector
<code>protease</code>	regular expression

**Details**

NA

**Value**

vector of integers

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`getNbPeptidesPerProtein`*Get number of peptides per protein*

---

**Description**

Get number of peptides per protein

**Usage**`getNbPeptidesPerProtein(eset)`**Arguments**

<code>eset</code>	ExpressionSet
-------------------	---------------

**Details**

NA

**Value**

table

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`getNbSpectraPerProtein`*Get number of spectra per protein*

---

**Description**

Get number of spectra per protein

**Usage**

```
getNbSpectraPerProtein(eset)
```

**Arguments**

eset	ExpressionSet
------	---------------

**Details**

NA

**Value**

table

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`getPeptides`*Digest protein*

---

**Description**

Digest protein

**Usage**

```
getPeptides(proteinSeq, proteaseRegExp = .getProteaseRegExp("trypsin"),  
            nbMiscleavages = 0)
```

**Arguments**

proteinSeq      protein sequence  
proteaseRegExp   protease Regular Expression  
nbMisleavages    default 0

**Details**

No details

**Value**

vector of peptides

**Note**

No note

**Examples**

```
print("No examples")
```

---

getRatios

*Calculate ratios, comparing all case to control*

---

**Description**

Calculate ratios, comparing all case to control

**Usage**

```
getRatios(eset, method = "median", log2 = T)
```

**Arguments**

eset              ExpressionSet  
method           median, mean, paired  
log2              transform

**Details**

No details

**Value**

ExpressionSet object

**Note**

No note

## References

NA

## Examples

```
print("No examples")
```

---

getRTNormFactors	<i>Get retentiontime base normalization factors</i>
------------------	-----------------------------------------------------

---

## Description

Get retentiontime base normalization factors

## Usage

```
getRTNormFactors(eset, minFeaturesPerBin = 100)
```

## Arguments

eset	ExpressionSet
minFeaturesPerBin	minumum number of features per bin. If nb. features are < minFeaturesPerBin -> include neighbouring bins.

## Details

No details

## Value

data.frame normalization factors per retention time bin (minute)

## Note

No note

## References

In Silico Instrumental Response Correction Improves Precision of Label-free Proteomics and Accuracy of Proteomics-based Predictive Models, Lyutvinskiy et al. (2013), <http://www.ncbi.nlm.nih.gov/pubmed/23589346>

## Examples

```
print("No examples")
```



---

getScoreCutOff	<i>Get score cutoff for a given fdr cut-off</i>
----------------	-------------------------------------------------

---

**Description**

Get score cutoff for a given fdr cut-off

**Usage**

```
getScoreCutOff(scores, isDecoy, fdrCutOff = 0.01)
```

**Arguments**

scores	peptide/protein identificationscore
isDecoy	vector of TRUE/FALSE
fdrCutOff	[0,1]

**Details**

NA

**Value**

scoreCutoff

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getSignalPerCondition	<i>Summarize replicate signal per condition (min)</i>
-----------------------	-------------------------------------------------------

---

**Description**

Summarize replicate signal per condition (min)

**Usage**

```
getSignalPerCondition(eset, method = "median")
```

**Arguments**

eset	ExpressionSet
method	median (default), mean, max, min, sd

**Details**

No details

**Value**

data.frame of per condition signals

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

getTopX

*Calculate Mean of X most intense features*

---

**Description**

Calculate Mean of X most intense features

**Usage**

```
getTopX(entryData, topX = 3)
```

**Arguments**

entryData	data.frame listing feature intensities of one entry. Typically rows corresponds to Peptide entries of one protein
topX	best X flyers

**Details**

No details

**Value**

vector of topX intensities per column (sample)

**Note**

No note

## References

Absolute quantification of proteins by LCMSE: A virtue of parallel MS acquisition, Silva (2006), <http://www.ncbi.nlm.nih.gov/pubmed/16219938>, Critical assessment of proteome-wide label-free absolute abundance estimation strategies. Ahrne (2013), <http://www.ncbi.nlm.nih.gov/pubmed/23794183>

## Examples

```
print("No examples")
```

---

getUserOptions	<i>Read User Specified Command Line Options</i>
----------------	-------------------------------------------------

---

## Description

Read User Specified Command Line Options

## Usage

```
getUserOptions(version = version)
```

## Arguments

version	Safequant version number
---------	--------------------------

## Details

No details

## Value

user options list

## Note

No note

## References

NA

## Examples

```
print("No examples")
```

---

ggDilutionCurve	<i>Plot dilution curve</i>
-----------------	----------------------------

---

**Description**

Plot dilution curve

**Usage**

```
ggDilutionCurve(dCurve, lod, title = "")
```

**Arguments**

dCurve	data.frame columns concentration, intensity
lod	limit of detection
title	plot title

**Value**

ggplot

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

globalNormalize	<i>Normalize, Norm factors calculated as median signal per run (column) over median of first run.</i>
-----------------	-------------------------------------------------------------------------------------------------------

---

**Description**

Normalize, Norm factors calculated as median signal per run (column) over median of first run.

**Usage**

```
globalNormalize(eset, globalNormFactors)
```

**Arguments**

eset	ExpressionSet
globalNormFactors	globalNormFactors

**Details**

No details

**Value**

eset ExpressionSet

**Note**

No note

**References**

NA

**See Also**

getGlobalNormFactors

**Examples**

```
print("No examples")
```

---

hClustHeatMap	<i>Hierarchical clustering heat map, cluster by runs intensity, features by ratio and display log2 ratios to control median</i>
---------------	---------------------------------------------------------------------------------------------------------------------------------

---

**Description**

Hierarchical clustering heat map, cluster by runs intensity, features by ratio and display log2 ratios to control median

**Usage**

```
hClustHeatMap(eset, conditionColors = .getConditionColors(eset),
  breaks = seq(-2, 2, length = 20), dendogram = "column",
  legendPos = "left", ...)
```

**Arguments**

eset	ExpressionSet
conditionColors	data.frame of colors per condition
breaks	default seq(-2,2,length=20)
dendogram	see heatmap.2 gplots
legendPos	see legend
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

isCon

*Check if protein is a contaminant entry*

---

**Description**

Check if protein is a contaminant entry

**Usage**

```
isCon(ac)
```

**Arguments**

ac                      vector of protein accession numbers

**Details**

contaminant proteins are typically annotated as: CON\_P0000

**Value**

vector TRUE/FALSE

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

isDecoy	<i>Check if protein is a decoy entry</i>
---------	------------------------------------------

---

**Description**

Check if protein is a decoy entry

**Usage**

```
isDecoy(ac)
```

**Arguments**

ac	vector of protein accession numbers
----	-------------------------------------

**Details**

decoy proteins are typically annotated as: REV\_P0000

**Value**

vector TRUE/FALSE

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

isStrippedACs	<i>Check if ACs are in "non-stripped" uniprot format e.g. "sp Q8CHJ2 AQP12_MOUSE"</i>
---------------	---------------------------------------------------------------------------------------

---

**Description**

Check if ACs are in "non-stripped" uniprot format e.g. "sp|Q8CHJ2|AQP12\_MOUSE"

**Usage**

```
isStrippedACs(acs)
```

**Arguments**

acs	accession numbers
-----	-------------------

**Details**

TRUE if less than 10

**Value**

boolean TRUE/FALSE

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

kinaseMotif	<i>Kinase motifs</i>
-------------	----------------------

---

**Description**

Human Protein Reference Database Serine/Threonine motifs [http://www.hprd.org/serine\\_motifs](http://www.hprd.org/serine_motifs)  
The variables are as follows:

**Usage**

```
kinaseMotif
```

**Format**

A data frame with 175 rows and 2 variables:

**motif** kinase motif)

**kinase** kinase



---

`maPlotSQ`*ma-plot*

---

**Description**

ma-plot

**Usage**

```
maPlotSQ(eset, sample = colnames(exprs(eset))[1], cex.lab = 1.5,  
  cex.axis = 1.5, lwd = 2, pch = 1, col = rgb(0, 100, 0, 50,  
  maxColorValue = 255), ...)
```

**Arguments**

<code>eset</code>	ExpressionSet
<code>sample</code>	selected condition
<code>cex.lab</code>	default 1.5
<code>cex.axis</code>	default 1.5
<code>lwd</code>	default 2
<code>pch</code>	default 1
<code>col</code>	green transparent
<code>...</code>	see plot

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`missinValueBarplot`*Plot Percentage of Features with with missing values*

---

**Description**

Plot Percentage of Features with with missing values

**Usage**

```
missinValueBarplot(eset,  
  col = as.character(.getConditionColors(eset)[pData(eset)$condition, ]),  
  cex.axis = 1.25, cex.lab = 1.25, ...)
```

**Arguments**

eset	ExpressionSet
col	col
cex.axis	cex.axis
cex.lab	cex.lab
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

option_list	<i>Command Line Option List</i>
-------------	---------------------------------

---

**Description**

Command Line Option List

**Usage**

```
option_list
```

**Format**

An object of class list of length 30.

---

pairsAnnot	<i>Plot lower triangle Pearsons <math>R^2</math>. Diagonal text, upper triangle all against all scatter plots with lm abline</i>
------------	----------------------------------------------------------------------------------------------------------------------------------

---

## Description

Plot lower triangle Pearsons  $R^2$ . Diagonal text, upper triangle all against all scatter plots with lm abline

## Usage

```
pairsAnnot(data, textCol = rep(1, ncol(data)), diagText = colnames(data),  
  col = rgb(0, 100, 0, 50, maxColorValue = 255), isHeatCol = F,  
  cexTxt = 2, ...)
```

## Arguments

data	data.frame
textCol	text color
diagText	diagnoal text
col	dot col
isHeatCol	heat colors
cexTxt	cex txt
...	see plot

## Details

No details

## Note

No note

## References

NA

## Examples

```
print("No examples")
```

parseMaxQuantProteinGroupTxt

*Parse MaxQuant Protein Group Txt*

---

## Description

Parse MaxQuant Protein Group Txt

## Usage

```
parseMaxQuantProteinGroupTxt(file = file, expDesign = expDesign,  
  method = "auc")
```

## Arguments

file	path to MaxQuant Protein txt file
expDesign	experimental design data.frame
method	auc (area under curve) or spc (spectral count)

## Details

No details

## Value

ExpressionSet object

## Note

No note

## References

NA

## See Also

[ExpressionSet](#)

## Examples

```
print("No examples")
```

---

parseProgenesisFeatureCsv

*Parse Progenesis Feature Csv Export*

---

## Description

Parse Progenesis Feature Csv Export

## Usage

```
parseProgenesisFeatureCsv(file = file,
  expDesign = getExpDesignProgenesisCsv(file), method = "auc")
```

## Arguments

file	path to Progenesis Feature csv file
expDesign	experimental design data.frame
method	auc (area under curve) or spc (spectral count)

## Details

No details

## Value

ExpressionSet object

## Note

No note

## References

NA

## See Also

[ExpressionSet](#)

## Examples

```
print("No examples")
```

---

`parseProgenesisPeptideMeasurementCsv`*Parse Progenesis Peptide Measurement Csv Export*

---

## Description

Parse Progenesis Peptide Measurement Csv Export

## Usage

```
parseProgenesisPeptideMeasurementCsv(file, expDesign = expDesign,  
  method = "auc",  
  expressionColIndices = .getProgenesisCsvExpressionColIndices(file, method =  
    method), uniqueProteins = F)
```

## Arguments

<code>file</code>	path to Progenesis Peptide Measurement csv file
<code>expDesign</code>	experimental design data.frame
<code>method</code>	auc (area under curve) or spc (spectral count)
<code>expressionColIndices</code>	default .getProgenesisCsvExpressionColIndices()
<code>uniqueProteins</code>	T/F keep unique peptides only

## Details

No details

## Value

ExpressionSet object

## Note

No note

## References

NA

## See Also

[ExpressionSet](#)

## Examples

```
print("No examples")
```

---

parseProgenesisProteinCsv  
*Parse Progenesis Protein Csv*

---

## Description

Parse Progenesis Protein Csv

## Usage

```
parseProgenesisProteinCsv(file = file, expDesign = expDesign,
  method = "auc")
```

## Arguments

file	path to Progenesis Protein csv file
expDesign	experimental design data.frame
method	auc (area under curve) or spc (spectral count)

## Details

No details

## Value

ExpressionSet object

## Note

No note

## References

NA

## See Also

[ExpressionSet](#)

## Examples

```
print("No examples")
```

---

parseScaffoldPTMReport

Parse scaffold PTM Spectrum Report

---

**Description**

Parse scaffold PTM Spectrum Report

**Usage**

parseScaffoldPTMReport(file)

**Arguments**

file                      path to Scaffold file

**Details**

No details

**Value**

data.frame

**Note**

No note

**References**

NA

**Examples**

print("No examples")

---

parseScaffoldRawFile

Parse scaffold output .xls file (RAW export)

---

**Description**

Parse scaffold output .xls file (RAW export)

**Usage**

parseScaffoldRawFile(file, expDesign = expDesign, keepFirstAcOnly = FALSE,  
isPurityCorrect = T)



**Arguments**

file	path to Scaffold file
expDesign	experimental design data.frame
keepFirstAcOnly	TRUE/FALSE If multiple ACs in Accession.Numbers filed. Then keep the first one only
isPurityCorrect	do purity correction

**Details**

No details

**Value**

ExpressionSet object

**Note**

No note

**References**

NA

**See Also**

[ExpressionSet](#)

**Examples**

```
print("No examples")
```

---

perFeatureNormalization

*Per Feature Normalization*

---

**Description**

Per Feature Normalization

**Usage**

```
perFeatureNormalization(eset, normFactors)
```

**Arguments**

eset	ExpressionSet
normFactors	matrix normalization factors (logged) (row names are proteins)

**Details**

Example Usage: Normalize phospho peptide signals for Protein Changes

**Value**

ExpressionSet object

**Note**

No note

**References**

No references

**Examples**

```
print("No examples")
```

---

```
plotAbsEstCalibrationCurve
```

*Plot absolut Estimation calibration Curve*

---

**Description**

Plot absolut Estimation calibration Curve

**Usage**

```
plotAbsEstCalibrationCurve(fit, dispElements = c("formula", "lowess",
  "stats"), xlab = "Conc. (CPC) ", ylab = "Pred. Conc. (CPC) ",
  predictorName = paste("log10(", names(coef(fit))[2], ")", sep = ""),
  text = F, cex.lab = 1, cex.axis = 1, cex.text = 1, cex.dot = 1,
  main = "", ...)
```

**Arguments**

fit	simple log-linear model
dispElements	c("formula","lowess","stats")
xlab	xlab
ylab	ylab
predictorName	predictorName
text	add names beside each dot
cex.lab	expansion factor for axis labels
cex.axis	expansion factor for axis
cex.text	expansion factor for legend
cex.dot	expansion factor for plotted dots
main	main
...	see plot

### Note

No note

### References

NA

### Examples

```
print("No examples")
```

---

`plotAdjustedVsNonAdjustedRatio`

*Plot adjusted tmt ratios vs original ratios*

---

### Description

Plot adjusted tmt ratios vs original ratios

### Usage

```
plotAdjustedVsNonAdjustedRatio(ratio, unAdjustedRatio)
```

### Arguments

<code>ratio</code>	<code>data.frame</code>
<code>unAdjustedRatio</code>	<code>data.frame</code>

### Details

plot adjusted tmt ratios vs original ratios

### Note

No note

### References

NA

### Examples

```
print("No examples")
```

---

plotExpDesign	<i>Display experimental design, high-lighting the control condition</i>
---------------	-------------------------------------------------------------------------

---

**Description**

Display experimental design, high-lighting the control condition

**Usage**

```
plotExpDesign(eset, condColors = .getConditionColors(eset), version = "X")
```

**Arguments**

eset	ExpressionSet
condColors	condition colors
version	version number

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotIdScoreVsFDR	<i>Plot FDR vs. identification score</i>
------------------	------------------------------------------

---

**Description**

Plot FDR vs. identification score

**Usage**

```
plotIdScoreVsFDR(idScore, qvals, qvalueThrs = 0.01,  
  ylab = "False Discovery Rate", xlab = "Identification Score", ...)
```

**Arguments**

idScore	vector of identification scores
qvals	vector of q-valres
qvalueThrs	threshold indicated by horizontal line
ylab	default False Discovery Rate
xlab	default Identification Score
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotLogo

*Plot sequence logo*

---

**Description**

Plot sequence logo

**Usage**

```
plotLogo(motif, bgPeptides = "ACDEFGHIKLMNPQRSTVWY", main = "",  
         targetResidues = c("S", "T", "Y"), ic.scale = F, ...)
```

**Arguments**

motif	list of target residue centered motifs
bgPeptides	peptides used to calculate residue background frequency (default uniform )
main	see plot
targetResidues	default [STY]
ic.scale	logical. If TRUE, the height of each column is proportional to its information content. Otherwise, all columns have the same height.

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

```
plotMSSignalDistributions
```

*Plot ms.signal distributions*

---

**Description**

Plot ms.signal distributions

**Usage**

```
plotMSSignalDistributions(d, col = 1:100, ylab = "Frequency",  
  xlab = "MS-Signal", ...)
```

**Arguments**

d	matrix of ms-signals
col	color
ylab	default "Frequency"
xlab	default "MS-Signal"
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`plotNbIdentificationsVsRT`*Plot the number of identified Features per Reteintion Time minute.*

---

**Description**

Plot the number of identified Features per Reteintion Time minute.

**Usage**

```
plotNbIdentificationsVsRT(eset, cex.axis = 1.25, cex.lab = 1.25,  
  col = "blue", lwd = 2, ...)
```

**Arguments**

<code>eset</code>	ExpressionSet
<code>cex.axis</code>	default 1.25
<code>cex.lab</code>	default 1.25
<code>col</code>	default "blue"
<code>lwd</code>	default 2
<code>...</code>	see plot see plot

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`plotNbValidDeFeaturesPerFDR`*Plot Total Number of differentially Abundant Features (applying ratio cutoff) vs. qValue/pValue for all conditions*

---

**Description**

Plot Total Number of differentially Abundant Features (applying ratio cutoff) vs. qValue/pValue for all conditions

**Usage**

```
plotNbValidDeFeaturesPerFDR(sqa, upRegulated = T, log2RatioCufOff = log2(1),  
  pvalCutOff = 1, isLegend = T, isAdjusted = T, ylab = "Nb. Features",  
  xlim = NA, ylim = NA, ...)
```

**Arguments**

sqa	SafeQuantAnalysis Object
upRegulated	TRUE/FALSE select for upregulated features
log2RatioCufOff	log2 ratio cut-off
pvalCutOff	pValue/qValue cut-off
isLegend	TRUE/FALSE display legend
isAdjusted	TRUE/FALSE qValues/pValue on x-axis
ylab	default Nb. Features
xlim	see plot
ylim	see plot
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotPrecMassErrorDistrib

*Plot Precursor Mass Error Distribution*

---

**Description**

Plot Precursor Mass Error Distribution

**Usage**

```
plotPrecMassErrorDistrib(eset, pMassTolWindow = c(-10, 10), ...)
```

**Arguments**

eset	ExpressionSet
pMassTolWindow	Precursor Mass Error Tolerance Window
...	see plot



**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotPrecMassErrorVsScore

*Plot precursorMass error v.s score highlighting decoy and displaying user specified user specified precursor mass filter*

---

**Description**

Plot precursorMass error v.s score highlighting decoy and displaying user specified user specified precursor mass filter

**Usage**

```
plotPrecMassErrorVsScore(eset, pMassTolWindow = c(-10, 10), ...)
```

**Arguments**

eset	ExpressionSet
pMassTolWindow	Precursor Mass Error Tolerance Window
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotQValueVsPValue	<i>Plot qValue vs pValue</i>
--------------------	------------------------------

---

**Description**

Plot qValue vs pValue

**Usage**

```
plotQValueVsPValue(sqa, lim = c(0, 1), ...)
```

**Arguments**

sqa	SafeQuantAnalysis Object
lim	x-axis and y-axis range
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotROC	<i>Plot Number of Identifications vs. FDR</i>
---------	-----------------------------------------------

---

**Description**

Plot Number of Identifications vs. FDR

**Usage**

```
plotROC(qvals, qvalueThrs = 0.01, xlab = "False Discovery Rate",
  ylab = "Nb. Valid Identifications", xlim = c(0, 0.1), breaks = 100,
  col = "blue", lwd = 1.5, ...)
```

**Arguments**

qvals	vector of q-values
qvalueThrs	threshold indicated by vertical line
xlab	default "False Discovery Rate"
ylab	default "Nb. Valid Identifications"
xlim	default c(0,0.1)
breaks	see breaks for hist function
col	default blue
lwd	default 1.5
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotRTNorm

---

*Plot all retention time profile overalying ratios*


---

**Description**

Plot all retention time profile overalying ratios

**Usage**

```
plotRTNorm(rtNormFactors, eset, samples = 1:ncol(rtNormFactors), main = "",
...)
```

**Arguments**

rtNormFactors	data.frame of normalization factor per r.t bin and sample, obtained by getRT-NormFactors
eset	ExprssionSet
samples	specify samples (sample numbers) to be plotted
main	main
...	see plot see plot

**Details**

No details

**Note**

No note

**References**

In Silico Instrumental Response Correction Improves Precision of Label-free Proteomics and Accuracy of Proteomics-based Predictive Models, Lyutvinskiy et al. (2013), <http://www.ncbi.nlm.nih.gov/pubmed/23589346>

**See Also**

[getRTNormFactors](#)

**Examples**

```
print("No examples")
```

---

plotRTNormSummary	<i>Plot all retention time normalization profiles</i>
-------------------	-------------------------------------------------------

---

**Description**

Plot all retention time normalization profiles

**Usage**

```
plotRTNormSummary(eset,
  col = as.character(.getConditionColors(eset)[pData(eset)$condition, 1]),
  ...)
```

**Arguments**

eset	ExpressionSet
col	condition colors
...	see plot

**Details**

No details

**Note**

No note

**References**

In Silico Instrumental Response Correction Improves Precision of Label-free Proteomics and Accuracy of Proteomics-based Predictive Models, Lyutvinskiy et al. (2013), <http://www.ncbi.nlm.nih.gov/pubmed/23589346>

**See Also**[getRTNormFactors](#)**Examples**

```
print("No examples")
```

---

plotScoreDistrib	<i>Plot identifications target decoy distribution</i>
------------------	-------------------------------------------------------

---

**Description**

Plot identifications target decoy distribution

**Usage**

```
plotScoreDistrib(targetScores, decoyScores, xlab = "Identification Score",  
  ylab = "Counts", ...)
```

**Arguments**

targetScores	target Scores
decoyScores	decoy Scores
xlab	default "Identification Score"
ylab	default "Counts"
...	see plot

**Details**

No details

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotVolcano	<i>Plots volcano, data points colored by max cv of the 2 compared conditions</i>
-------------	----------------------------------------------------------------------------------

---

**Description**

Plots volcano, data points colored by max cv of the 2 compared conditions

**Usage**

```
plotVolcano(obj, ratioThrs = 1, pValueThreshold = 0.01, adjusted = T, ...)
```

**Arguments**

obj	safeQuantAnalysis object or data.frame
ratioThrs	default 1
pValueThreshold	default 0.01
adjusted	TRUE/FALSE plot qValues or pValues on y-axis
...	see plot

**Details**

data.frame input object should contain 3 columns (ratio,qValue,cv)

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

plotXYDensity	<i>Scatter plot with density coloring</i>
---------------	-------------------------------------------

---

**Description**

Scatter plot with density coloring

**Usage**

```
plotXYDensity(x, y, isFitLm = T, legendPos = "bottomright",
  disp = c("abline", "R", "Rc"), pch = 20, ...)
```

**Arguments**

x	number vector
y	number vector
isFitLm	fit linear model
legendPos	see legend
disp	c("abline","R","Rc") display options
pch	see plot
...	see plot

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

purityCorrectTMT	<i>Correct channel intensities based on Reporter ion Isotopic Distributions</i>
------------------	---------------------------------------------------------------------------------

---

**Description**

Correct channel intensities based on Reporter ion Isotopic Distributions

**Usage**

```
purityCorrectTMT(tmtData, impurityMatrix = impurityMatrix)
```

**Arguments**

tmtData	data.frame containing tmt channel intensities
impurityMatrix	correction matrix

**Details**

Same method as MSnbase, and described in Breitwieser et al. 2012 (Book Chapter)

**Value**

data.frame of corrected tmt intensities

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

removeOutliers	<i>Set value to NA if it deviates with more than 1.5 * IQR from lower/upper quantile</i>
----------------	------------------------------------------------------------------------------------------

---

**Description**

Set value to NA if it deviates with more than 1.5 \* IQR from lower/upper quantile

**Usage**

```
removeOutliers(x, na.rm = TRUE, ...)
```

**Arguments**

x	vector numeric
na.rm	logical indicating whether missing values should be removed.
...	quantile args

**Details**

No details

**Note**

No note

**References**

NA

**See Also**

NA

**Examples**

```
print("No examples")
```



---

rollUp	<i>Roll up feature intensities per unique column combination</i>
--------	------------------------------------------------------------------

---

**Description**

Roll up feature intensities per unique column combination

**Usage**

```
rollUp(eset, method = "sum", featureDataColumnName = c("proteinName"))
```

**Arguments**

eset	ExpressionSet
method	"sum", "mean" or "top3"
featureDataColumnName	vector of column names e.g. peptide or proteinName

**Details**

featureDataColumnName = c("peptide","charge","ptm"), method= c("sum"), sums up intensities per peptide modification charge state

**Value**

ExpressionSet object

**Note**

No note

**References**

No references

**Examples**

```
print("No examples")
```

---

rtNormalize	<i>Normalization data per retention time bin</i>
-------------	--------------------------------------------------

---

**Description**

Normalization data per retention time bin

**Usage**

```
rtNormalize(eset, rtNormFactors)
```

**Arguments**

eset	ExpressionSet
rtNormFactors	obtained using getRTNormFactors

**Details**

Normalize for variations in electrospray ionization current.

**Value**

data.frame normalization factors per retention time bin (minute)

**Note**

No note

**References**

In Silico Instrumental Response Correction Improves Precision of Label-free Proteomics and Accuracy of Proteomics-based Predictive Models, Lyutvinskiy et al. (2013), <http://www.ncbi.nlm.nih.gov/pubmed/23589346>

**See Also**

[getRTNormFactors](#)

**Examples**

```
print("No examples")
```

---

safeQuantAnalysis	<i>safeQunat s3 class</i>
-------------------	---------------------------

---

**Description**

safeQunat s3 class

**Usage**

```
safeQuantAnalysis(eset = eset, method = c("global", "naRep", "pairwise"),  
  intensityAdjustmentObj = NA, fcThrs = 1)
```

**Arguments**

eset	ExpressionSet
method	c("global","naRep","rt","quantile","pairwise","all")
intensityAdjustmentObj	list
fcThrs	fold change threshold

---

setNbPeptidesPerProtein	<i>Set nbPeptides coulmn of featureData</i>
-------------------------	---------------------------------------------

---

**Description**

Set nbPeptides coulmn of featureData

**Usage**

```
setNbPeptidesPerProtein(eset)
```

**Arguments**

eset	ExpressionSet
------	---------------

**Details**

NA

**Value**

eset

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

setNbSpectraPerProtein

*Set nbPeptides coulumn of featureData*

---

**Description**

Set nbPeptides coulumn of featureData

**Usage**

```
setNbSpectraPerProtein(eset)
```

**Arguments**

eset	ExpressionSet
------	---------------

**Details**

NA

**Value**

eset

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

---

`sqNormalize`*Normalize*

---

**Description**

Normalize

**Usage**

```
sqNormalize(eset, method = "global")
```

**Arguments**

<code>eset</code>	ExpressionSet
<code>method</code>	<code>c("global","rt","quantile")</code>

**Details**

No details

**Value**

eset ExpressionSet

**Note**

No note

**References**

NA

**See Also**

`getGlobalNormFactors`, `getRTNormFactors`

**Examples**

```
print("No examples")
```

---

standardise

*Standardise data*

---

**Description**

Standardise data

**Usage**

```
standardise(d)
```

**Arguments**

d                      vector or data.frame or matrix

**Details**

No details

**Value**

vector or data.frame or matrix

**Note**

No note

**Examples**

```
print("No examples")
```

---

stripACs

*strip uniprot format e.g. "sp|Q8CHJ2|AQP12\_MOUSE" -> Q8CHJ2*

---

**Description**

strip uniprot format e.g. "sp|Q8CHJ2|AQP12\_MOUSE" -> Q8CHJ2

**Usage**

```
stripACs(acs)
```

**Arguments**

acs                      accession numbers

**Details**

TRUE if less than 10

**Value**

vector character

**Note**

No note

**References**

NA

**Examples**

```
print("No examples")
```

# Index

## \*Topic **datasets**

COLORS, [6](#)

kinaseMotif, [40](#)

option\_list, [42](#)

## \*Topic **normalization**

getGlobalNormFactors, [18](#)

getLoocvFoldError, [24](#)

globalNormalize, [36](#)

removeOutliers, [64](#)

sqNormalize, [69](#)

addIdQvalues, [3](#)

addScaffoldPTMFAnnotations, [4](#)

barplotMSSignal, [5](#)

COLORS, [6](#)

createExpDesign, [6](#)

createExpressionDataset, [7](#)

createPairedExpDesign, [8](#)

cvBoxplot, [9](#)

dotProduct, [10](#)

eBayes, [15](#)

expDesignTagToExpDesign, [10](#)

export, [11](#)

ExpressionSet, [7](#), [8](#), [44–47](#), [49](#)

getAAProteinCoordinates, [12](#)

getAllCV, [13](#)

getAllDotProduct, [14](#)

getAllEBayes, [14](#)

getBaselineIntensity, [15](#)

getCV, [13](#), [16](#)

getExpDesignProgenesisCsv, [17](#)

getGlobalNormFactors, [18](#)

getIBAQset, [18](#)

getIdLevelQvals, [4](#), [19](#)

getImpuritiesMatrix, [20](#)

getIntSumPerProtein, [21](#)

getKinaseFreq, [22](#)

getKinases, [22](#)

getLOD, [23](#)

getLoocvFoldError, [24](#)

getMaxIndex, [24](#)

getMeanCenteredRange, [25](#)

getModifProteinCoordinates, [25](#)

getMotifFreq, [26](#)

getMotifX, [27](#)

getNbDetectablePeptides, [28](#)

getNbMisCleavages, [28](#)

getNbPeptidesPerProtein, [29](#)

getNbSpectraPerProtein, [30](#)

getPeptides, [30](#)

getRatios, [31](#)

getRTNormFactors, [32](#), [60](#), [61](#), [66](#)

getScoreCutOff, [33](#)

getSignalPerCondition, [33](#)

getTopX, [34](#)

getUserOptions, [35](#)

ggDilutionCurve, [36](#)

globalNormalize, [36](#)

hClustHeatMap, [37](#)

isCon, [38](#)

isDecoy, [39](#)

isStrippedACs, [39](#)

kinaseMotif, [40](#)

maPlotSQ, [41](#)

missinValueBarplot, [41](#)

option\_list, [42](#)

pairsAnnot, [43](#)

parseMaxQuantProteinGroupTxt, [44](#)

parseProgenesisFeatureCsv, [45](#)

parseProgenesisPeptideMeasurementCsv,  
[46](#)

parseProgenesisProteinCsv, [47](#)

parseScaffoldPTMReport, [48](#)

parseScaffoldRawFile, [48](#)

perFeatureNormalization, [49](#)

plotAbsEstCalibrationCurve, [50](#)

plotAdjustedVsNonAdjustedRatio, [51](#)

plotExpDesign, [52](#)

plotIdScoreVsFDR, [52](#)



plotLogo, [53](#)  
plotMSSignalDistributions, [54](#)  
plotNbIdentificationsVsRT, [55](#)  
plotNbValidDeFeaturesPerFDR, [55](#)  
plotPrecMassErrorDistrib, [56](#)  
plotPrecMassErrorVsScore, [57](#)  
plotQValueVsPValue, [58](#)  
plotROC, [58](#)  
plotRTNorm, [59](#)  
plotRTNormSummary, [60](#)  
plotScoreDistrib, [61](#)  
plotVolcano, [62](#)  
plotXYDensity, [62](#)  
purityCorrectTMT, [63](#)  
  
removeOutliers, [64](#)  
rollUp, [65](#)  
rtNormalize, [66](#)  
  
safeQuantAnalysis, [11](#), [67](#)  
setNbPeptidesPerProtein, [67](#)  
setNbSpectraPerProtein, [68](#)  
sqNormalize, [69](#)  
standardise, [70](#)  
stripACs, [70](#)