

# Package ‘quantable’

May 15, 2018

**Type** Package

**Title** Streamline Descriptive Analysis of Quantitative Data Matrices

**Version** 0.3.6

**Date** 2018-04-14

**Author** Witold Wolski <wewolski@gmail.com>

**Maintainer** Witold Wolski <wewolski@gmail.com>

**Depends** R (>= 3.1.0)

**Description** Methods which streamline the descriptive analysis of quantitative matrices. Matrix columns are samples while rows are features i.e. proteins, genes. Includes methods for visualization (e.g. Heatmaps, Volcanos, pairwise QQ, Bland-Altman plot), summary statistics (e.g. CV), data normalization methods (e.g. robustscale). Read function for Progenesis.

**License** GPL-3

**LazyData** TRUE

**Imports** Matrix, RColorBrewer, caret, dplyr, e1071, ggplot2, ggrepel, gplots, readr, reshape2, rlang, plyr, pROC, scales, stringr, tibble, tidyr

**URL** <https://github.com/protViz/quantable>

**BugReports** <https://github.com/protViz/quantable/issues>

**Repository** CRAN

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Date/Publication** 2018-05-15 21:03:41 UTC

## R topics documented:

addSpecialProteins . . . . .	3
altmanbland . . . . .	4
colNAs . . . . .	4
CV . . . . .	5

CVlog . . . . .	5
dat2Matrix . . . . .	6
determineCut . . . . .	6
distmy . . . . .	7
filterSignificant . . . . .	8
fisherExact . . . . .	9
getBlueScale . . . . .	9
getBlueWhiteRed . . . . .	10
getDiv . . . . .	10
getGreensScale . . . . .	11
getRedScale . . . . .	11
getTValuesForVolcano . . . . .	12
getWRValuesForVolcano . . . . .	13
imageColorscale . . . . .	13
imageWithLabels . . . . .	14
imageWithLabelsNoLayout . . . . .	15
image_nan . . . . .	16
jackknifeMatrix . . . . .	17
makeROCplot . . . . .	18
matrix_to_tibble . . . . .	18
multigroupFCDATA . . . . .	19
multigroupVolcano . . . . .	19
mypairs . . . . .	20
mypairsSmooth . . . . .	20
my_jackknife . . . . .	21
onesamplegreaterT . . . . .	22
pairsQQ . . . . .	22
panel.cor . . . . .	23
panel.hist . . . . .	23
plotOneSideVolcano . . . . .	24
ProgenesisBuildAnnotation . . . . .	24
ProgenesisRead . . . . .	25
pvalCorMat . . . . .	26
quantable . . . . .	26
removeDecorrelated . . . . .	27
removeNARows . . . . .	27
robustscale . . . . .	28
rowNAs . . . . .	28
runFun . . . . .	29
runrobustscale . . . . .	29
runTICscale . . . . .	30
scaleByGroup . . . . .	31
setdiff_data.frame . . . . .	32
simpleheatmap . . . . .	32
split2table . . . . .	33
sumtop . . . . .	33
unpivot . . . . .	34
uppertriang . . . . .	34

volcano2G . . . . .	35
volcano2GB . . . . .	36
volcanoplot . . . . .	37
write.tab . . . . .	38
write.vector . . . . .	38
xxx_replace_xxx . . . . .	39

<b>Index</b>	<b>40</b>
--------------	-----------

---

addSpecialProteins	<i>add special labels</i>
--------------------	---------------------------

---

## Description

add special labels

## Usage

```
addSpecialProteins(p, dataX, special, foldchange = "log2FC",
  pvalue = "q.mod", labels = "names")
```

## Arguments

p	ggplot2
dataX	data.frame
special	additional special labels for those entries in the labels column below.
foldchange	name of fold change column
pvalue	name of p-value column
labels	name of labels column

## Examples

```
foldchange <- rnorm(1000)
pvals <- rexp(1000)
names <- sample(colors(),1000,replace=TRUE)

dataX <- data.frame(
  q.mod = pvals,
  log2FC = foldchange,
  names = names
)
library(rlang)
foldchange = "log2FC"
p <- volcano2GB(dataX, pthresh=0.1, log2FCthresh=0.5 , main='test',
  repel.segment.size=0.3,
  repel.text.size=2)
special <- sample(colors(),5)
```

```
p <- addSpecialProteins(p, dataX, special)
p
```

---

altmanbland

*altman-bland plot*


---

### Description

plots  $\text{abs}(x-y)$  against  $(y+x)/2$

### Usage

```
altmanbland(x, y, main = "", pch = ".", log = "")
```

### Arguments

x	- input intensities
y	- input intensities
main	plotting parameters
pch	- plot character
log	- should the x y axis be log transformed possible values "x" "y" or "xy"

### Examples

```
x <- seq(1:300)/10
x <- x + rnorm(length(x),0,0.5)
y <- seq(1:300)/10
y <- y + rnorm(length(y),0,0.5)
altmanbland(y,x)
```

---

colNAs

*get NR of NA's per matrix or data.frame column*


---

### Description

get NR of NA's per matrix or data.frame column

### Usage

```
colNAs(x)
```

### Arguments

x	matrix or data.frame
---	----------------------

---

CV	<i>compute CV for each row in matrix</i>
----	--

---

**Description**

Typically used to create and violin plot

**Usage**

```
CV(data, top = 30, na.rm = TRUE)
```

**Arguments**

data	matrix
top	remove top (default 30) CV
na.rm	default TRUE

**Examples**

```
dat <- matrix(rnorm(1000,10,5), ncol=20)
dim(dat)
cv <- CV(dat, top=5)
length(cv)
stopifnot(length(cv) == 45)
hist(cv)
```

---

CVlog	<i>geometric coefficient of variation (CV for log transformed data) Typically used to create and violin plot</i>
-------	--

---

**Description**

geometric coefficient of variation (CV for log transformed data) Typically used to create and violin plot

**Usage**

```
CVlog(data, top = 30)
```

**Arguments**

data	matrix
top	remove top (default 30) CV

**See Also**

CV

**Examples**

```
dat <- matrix(rnorm(1000,10,5), ncol=20)
dim(dat)
cv <- CVlog(dat, top=5)
length(cv)
stopifnot(length(cv) == 45)
hist(cv)
```

---

dat2Matrix	<i>converts sparse representation to dense where row and col can be character vectors</i>
------------	---

---

**Description**

converts sparse representation to dense where row and col can be character vectors

**Usage**

```
dat2Matrix(row, col, x)
```

**Arguments**

row	row positions
col	column positions
x	values for matrix

**Value**

matrix

---

determineCut	<i>determine best Accuracy cutoff</i>
--------------	---------------------------------------

---

**Description**

determine best Accuracy cutoff

**Usage**

```
determineCut(cases, controls, plot = FALSE, scanstep = 0.01)
```

**Arguments**

cases	a factor of predicted classes
controls	a factor of classes to be used as the true results
plot	create plot of thresholds versus Accuracy
scanstep	step size for threshold estimation

**Examples**

```
library(pROC)
cases <- rnorm(100,-1,1.1)
controls <- rnorm(500,1,1.1)
cut <- determineCut(cases,controls)
makeROCplot(cases,controls,abline=cut)
```

---

distmy	<i>distance among matrix columns with freely choosable distance function</i>
--------	--

---

**Description**

distance among matrix columns with freely choosable distance function

**Usage**

```
distmy(x, func, init = NA, diag = TRUE)
```

**Arguments**

x	data
func	function taking 2 arrays x, y
init	how to initialize the output matrix
diag	should the diagonal be also computed

**Value**

matrix with  $\text{dist}[i,j] = \text{func}(x[,i],[x,j])$

**Examples**

```
mat = matrix(rnorm(10*5000),ncol=10)
redist = distmy(mat,function(x,y){mean(abs(x-y))},init=NA,diag=FALSE)
image(redist)
redist = distmy(mat,cor,init=0,diag=FALSE)
image(redist)
redist = distmy(mat,function(x,y){ks.test(x,y)$p.value},init=1,diag=TRUE)
image(redist)
```

```
hist(uppertriang(redist))
range(uppertriang(redist))
which(redist < 0.05 , arr.ind = TRUE)
```

---

filterSignificant      *filters significant values and returns them as list of data.frames*

---

## Description

filters significant values and returns them as list of data.frames

## Usage

```
filterSignificant(foldchange, pvals, labels = NULL, pthresh = 0.05,
  foldchangethresh = 1, biasAdjust = FALSE)
```

## Arguments

foldchange	log2 fold changes
pvals	p values
labels	e.g. protein ID's
pthresh	pvalue threshold
foldchangethresh	fold change threshold
biasAdjust	adjustment for p-values

## Examples

```
library(quantable)
foldchange <- rnorm(1000)
pvals <- rexp(1000)
filterSignificant(foldchange, pvals,
  rep("blabla",length(pvals)), pthresh=0.1, foldchangethresh=1)
filterSignificant(foldchange[foldchange>0], pvals[foldchange>0],
  rep("blabla",length(pvals[foldchange>0])), pthresh=0.1, foldchangethresh=1)
filterSignificant(foldchange[foldchange<0], pvals[foldchange<0],
  rep("blabla",length(pvals[foldchange<0])), pthresh=0.1, foldchangethresh=1)
tt <- filterSignificant(foldchange, pvals,
  rep("blabla",length(pvals)), pthresh=0.1, foldchangethresh=10)
```



---

fisherExact	<i>get p-values using fishers exact test for count data</i>
-------------	---

---

**Description**

get p-values using fishers exact test for count data

**Usage**

```
fisherExact(x, y, accessions)
```

**Arguments**

x	- array
y	- array
accessions	accession string

**Value**

data frame with accessions, pval, pvaldj (BH adjusted p.values), fchange (log2 FC).

**Examples**

```
accessions <- letters
x <- sample(100,length(letters))
y <- sample(100,length(letters))
res <- fisherExact(x,y,accessions)
volcanoplot(res$fchange, res$pvaladj, labels = res$accessions)
```

---

getBlueScale	<i>create blue color scale</i>
--------------	--------------------------------

---

**Description**

create blue color scale

**Usage**

```
getBlueScale(length = 10, increasing = FALSE)
```

**Arguments**

length	nr of colors
increasing	default FALSE

**Examples**

```
library(scales)
show_col(getBlueScale(5))
```

---

```
getBlueWhiteRed      create blue white red palette
```

---

**Description**

create blue white red palette

**Usage**

```
getBlueWhiteRed(length = 21, increasing = TRUE)
```

**Arguments**

length	nr of colors
increasing	default TRUE

**Examples**

```
library(scales)
show_col(getBlueWhiteRed(21))
```

---

```
getDiv      create divergent palette
```

---

**Description**

create divergent palette

**Usage**

```
getDiv(length = 11, pallete = 3, increasing = TRUE)
```

**Arguments**

length	nr of colors
pallete	there are a few divergent palletes in brewer_pal, default 1
increasing	default TRUE

### Examples

```
library(scales)  
show_col(getDiv(21,4))  
show_col(getDiv(21,4, increasing=FALSE))
```

---

getGreensScale	<i>create green color scale</i>
----------------	---------------------------------

---

### Description

create green color scale

### Usage

```
getGreensScale(length = 10, increasing = FALSE)
```

### Arguments

length	nr of colors
increasing	default FALSE

### Examples

```
library(scales)  
show_col(getGreensScale(5))
```

---

getRedScale	<i>create red color scale</i>
-------------	-------------------------------

---

### Description

create red color scale

### Usage

```
getRedScale(length = 10, increasing = FALSE)
```

### Arguments

length	nr of colors
increasing	default FALSE

## Examples

```
library(scales)
show_col(getRedScale(12))
```

---

getTValuesForVolcano    *get p-values of t-test values for volcano*

---

## Description

get p-values of t-test values for volcano

## Usage

```
getTValuesForVolcano(x, y, alternative = "two.sided")
```

## Arguments

x	- one data matrix
y	- second data matrix
alternative	two.sided, less, greater

## Value

list with three fields fchange (fold change) , pval and pvaladj

## Examples

```
a <- t(replicate(200, rnorm(20, runif(1, -3, 3), 1)))
b <- a[1:100, ]
a <- a[101:200, ]
boxplot(t(a[1:20, ]))
boxplot(t(b[1:20, ]))
res <- getTValuesForVolcano(a, b)
volcanoplot(res$fchange , res$pval)
```

---

getWRValuesForVolcano *get p-values of wilcoxon rank sum test for volcano*

---

**Description**

get p-values of wilcoxon rank sum test for volcano

**Usage**

```
getWRValuesForVolcano(x, y, paired = FALSE, adjust = TRUE)
```

**Arguments**

x - one data matrix  
y - second data matrix  
paired a logical indicating whether you want a paired t-test.  
adjust pvalues using Benjamin Hochberg

**Value**

list with two fields fchange (fold change) and pval

**Examples**

```
a <- t(replicate(200, rnorm(20, runif(1, -3, 3), 1)))  
b <- a[1:100, ]  
a <- a[101:200, ]  
boxplot(t(a[1:20, ]))  
boxplot(t(b[1:20, ]))  
res <- getWRValuesForVolcano(a, b)  
volcanoplot(res$fchange, res$pval)
```

---

imageColorscale *if you need an colorscale to you imagelables use this*

---

**Description**

if you need an colorscale to you imagelables use this

**Usage**

```
imageColorscale(x, cex = 1, cex.axis = 0.5, col = heat.colors(12),  
digits = 2, xlim = NULL, breaks)
```

**Arguments**

x	data the data matrix
cex	cex
cex.axis	cex.axis
col	colors used
digits	number of digits on color scale, default 2
zlim	zlim
breaks	optional argument passed to image (see image for more details)

**Examples**

```
x = matrix(rnorm(20*30, 5), ncol=20)
rownames(x) <- 1:30
colnames(x) <- letters[1:20]
imageColorscale(x)
imageColorscale(x,col=getBlueWhiteRed(), zlim=c(-1,1))
imageColorscale(x,col=getBlueWhiteRed(), zlim=c(-5,5))
```

---

 imageWithLabels

*image plot with labels*


---

**Description**

image plot with labels

**Usage**

```
imageWithLabels(x, col.labels = colnames(x), row.labels = rownames(x),
  cex = 1, cex.axis = 0.5, main = NULL, col = heat.colors(12),
  digits = 2, marLeft = graphics::par()$mar,
  marRight = graphics::par()$mar, xlab = "", ylab = "", zlim = NULL,
  na.color = "gray", widths = c(4, 1), ...)
```

**Arguments**

x	matrix
col.labels	rownames(x)
row.labels	colnames(x)
cex	size of labels
cex.axis	size of axis labels
main	main title
col	color map for matrix
digits	number of digits on colorscale, default 2

marLeft	margins of left image see ?par for more detail
marRight	margins of right image see ?par for more detail
xlab	x label
ylab	y label
zlim	z value range, default NULL an determined from x
na.color	na.color
widths	controls the size of left and right pane
...	passed to image

### Examples

```
x = matrix(rnorm(20*30),ncol=20)
rownames(x) <- 1:30
colnames(x) <- letters[1:20]
martmp <- par()$mar
imageWithLabels(x)
imageWithLabels(x,marLeft = c(5,5,2,2),marRight=c(0,0,0,0),xlab="ttt",ylab="bbb")
par(mar = martmp)
```

---

```
imageWithLabelsNoLayout
```

*image plot with labels*

---

### Description

image plot with labels

### Usage

```
imageWithLabelsNoLayout(x, col.labels = colnames(x),
  row.labels = rownames(x), cex = 1, cex.axis = 0.5, main = NULL,
  col = heat.colors(12), digits = 2, xlab = "", ylab = "",
  zlim = NULL, na.color = "gray", textB = NULL, ...)
```

### Arguments

x	matrix
col.labels	rownames(x)
row.labels	colnames(x)
cex	size of labels
cex.axis	size of axis lables
main	main title
col	color map for matrix

digits	number of digits on colorscale, default 2
xlab	x label
ylab	y label
zlim	z value range, default NULL and determined from x
na.color	na.color
textB	indicate if correlation (text) should be added to heatmap. If - with how many digits. default = NULL (do not add).
...	passed to image

### Examples

```
x = matrix(rnorm(20*30), ncol=20)
rownames(x) <- 1:30
colnames(x) <- letters[1:20]
quantile:::image_nan(x, textB=1)

imageWithLabelsNoLayout(x, col = heat.colors(13), textB=2, text.cex=0.6)
imageWithLabelsNoLayout(x, col = heat.colors(12), breaks=seq(min(x), max(x), length=13))
x[3,3] <- NA
imageWithLabelsNoLayout(x, col = heat.colors(12),
breaks=seq(min(x, na.rm=TRUE),
max(x, na.rm=TRUE), length=13))
imageWithLabelsNoLayout(x, xlab="ttt", ylab="bbb")
imageWithLabelsNoLayout(x, xlab="ttt", ylab="bbb", zlim=c(0,2))
```

---

image_nan	<i>Copy of <a href="http://stackoverflow.com/questions/20977477/how-to-assign-a-specific-color-to-na-in-an-image-plot">http://stackoverflow.com/questions/20977477/how-to-assign-a-specific-color-to-na-in-an-image-plot</a></i>
-----------	--

---

### Description

Copy of <http://stackoverflow.com/questions/20977477/how-to-assign-a-specific-color-to-na-in-an-image-plot>

### Usage

```
image_nan(z, col = heat.colors(12), zlim = NULL, na.color = "gray",
outside.below.color = "green", outside.above.color = "green", breaks,
textB = NULL, text.cex = 0.8, ...)
```

### Arguments

z	matrix
col	color gradient
zlim	range



na.color	for displaying NA's.
outside.below.color	color below zlim
outside.above.color	color above zlim
breaks	do not remember (see usage in imageWithLabelsNoLayout)
textB	display numerical values of the matrix as text, default NULL no
text.cex	size of numerical values
...	further parameters for plot method

---

jackknifeMatrix	<i>Compute correlation matrix with jack</i>
-----------------	---

---

### Description

Compute correlation matrix with jack

### Usage

```
jackknifeMatrix(dataX, distmethod, ...)
```

### Arguments

dataX	data.frame with transition intensities per peptide
distmethod	dist or correlation method working with matrix i.e. cor
...	further parameters to method

### Value

summarizes results produced with my\_jackknife

### Examples

```
dataX <- matrix(rnorm(20), ncol=4)
rownames(dataX) <- paste("R", 1:nrow(dataX), sep="")
colnames(dataX) <- paste("C", 1:ncol(dataX), sep="")
tmp <- my_jackknife(dataX, cor, use="pairwise.complete.obs", method="pearson")

jackknifeMatrix(dataX, cor)
jackknifeMatrix(dataX, cor, method="spearman")
```

---

makeROCplot	<i>create density plots of cases and controls and ROC plot</i>
-------------	--

---

**Description**

create density plots of cases and controls and ROC plot

**Usage**

```
makeROCplot(cases, controls, label = "", xlab = "P(X==1)", xlim = NULL,
  abline = NULL)
```

**Arguments**

cases	a factor of predicted classes
controls	a factor of classes to be used as the true results
label	image main label
xlab	label for x axis
xlim	x axis extend
abline	draws vertical abline to indicate best threshold

**Examples**

```
library(pROC)
cases <- rnorm(100,-1,1.5)
controls <- rnorm(300,1,1.5)
makeROCplot(cases,controls)
```

---

matrix_to_tibble	<i>Matrix to tibble (taken from tidyquant)</i>
------------------	--

---

**Description**

Matrix to tibble (taken from tidyquant)

**Usage**

```
matrix_to_tibble(x, preserve_row_names = TRUE, ...)
```

**Arguments**

x	a matrix
preserve_row_names	should rownames be kept (default TRUE)
...	further parameters passed to as_tibble

---

multigroupFCDATA	<i>example data from multigroup analysis</i>
------------------	--

---

**Description**

example data from multigroup analysis

---

multigroupVolcano	<i>plot volcano given multiple conditions</i>
-------------------	---

---

**Description**

plot volcano given multiple conditions

**Usage**

```
multigroupVolcano(misspX, effect = "fc", type = "p.adjust",
  condition = "condition", colour = "colour", xintercept = c(-2, 2),
  pvalue = 0.05, label = NULL, size = 1, segment.size = 0.3,
  segment.alpha = 0.3, ablines = data.frame(fc = c(0, 0), p = c(0.01, 0.05),
  Area = c("p=0.01", "p=0.05")))
```

**Arguments**

misspX	data in long format
effect	column containing effect sizes
type	column containing p-values, q.values etc
condition	column with condition
colour	colouring of points
xintercept	fc thresholds
pvalue	pvalue threshold
label	column containing labels
size	controls size of text
segment.size	controls size of lines
segment.alpha	controls visibility of lines
ablines	adds ablines horizontal and vertical

**Examples**

```
data(multigroupFCDATA)
colnames(multigroupFCDATA)
multigroupVolcano(multigroupFCDATA, effect="logFC",
  type="adj.P.Val", condition="Condition", colour="colour", label="Name" )
```

---

mypairs	<i>normal pairs plot with different pch and plus abline</i>
---------	---

---

**Description**

normal pairs plot with different pch and plus abline

**Usage**

```
mypairs(dataframe, legend = FALSE, pch = ".", ...)
```

**Arguments**

dataframe	data matrix or data.frame as normally passed to pairs
legend	add legend to plots
pch	point type default "."
...	params usually passed to pairs

**See Also**

also [pairs](#)

**Examples**

```
tmp = matrix(rep((1:100),times = 4) + rnorm(100*4,0,3),ncol=4)
mypairs(tmp,log="xy",main="small data")
mypairs(tmp,log="xy",main="small data", legend=TRUE)
```

---

mypairsSmooth	<i>smoothScatter pairs</i>
---------------	----------------------------

---

**Description**

smoothScatter pairs

**Usage**

```
mypairsSmooth(dataframe, legend = FALSE, ...)
```

**Arguments**

dataframe	data matrix or data.frame as normally passed to pairs
legend	add legend to plots
...	params usually passed to pairs

**See Also**

also [pairs](#)

**Examples**

```
tmp = matrix(rep((1:100),times = 4) + rnorm(100*4,0,3),ncol=4)
mypairsSmooth(tmp,main="small data", legend=TRUE)
mypairsSmooth(tmp,main="small data", diag.panel=panel.hist)
mypairsSmooth(tmp,log="xy",main="small data", legend=TRUE)
```

---

my_jackknife	<i>copute jack knife</i>
--------------	--------------------------

---

**Description**

copute jack knife

**Usage**

```
my_jackknife(xdata, .method, ...)
```

**Arguments**

xdata	matrix
.method	method i.e. cor, parameters
...	further parameters to .method

**Value**

list with all jackknife matrices

**Examples**

```
xx <- matrix(rnorm(20), ncol=4)
cortest <- function(x){print(dim(x));cor(x)}
my_jackknife(xx, cortest)
my_jackknife(xx, cor, use="pairwise.complete.obs", method="pearson")
```

onesamplegreaterT      *One sample single sided t-test on matrix*

---

**Description**

One sample single sided t-test on matrix

**Usage**

```
onesamplegreaterT(ddtas, adjust = TRUE)
```

**Arguments**

ddtas                    - matrix  
adjust                   - should p-values be adjusted

**Examples**

```
data <- matrix(rnorm(100,1,1), ncol=5)  
dim(data)  
rownames(data) <- 1:20  
onesamplegreaterT(data)
```

---

pairsQQ                    *pairsplot of QQ plots*

---

**Description**

pairsplot of QQ plots

**Usage**

```
pairsQQ(obj, main = "")
```

**Arguments**

obj                        dataframe or matrix  
main                       - title

**See Also**

[qqplot](#) and [pairs](#)

**Examples**

```
tmp = matrix(rep((1:100),times = 4) + rnorm(100*4,0,3),ncol=4)
pairsQQ( tmp)
```

---

panel.cor	<i>correlation panel for pairs plot function (used as default in mypairsSmooth)</i>
-----------	---

---

**Description**

correlation panel for pairs plot function (used as default in mypairsSmooth)

**Usage**

```
panel.cor(x, y, digits = 2, ...)
```

**Arguments**

x	numeric data
y	numeric data
digits	number of digits to display
...	not used

---

panel.hist	<i>histogram panel for pairs function (used as default in mypairsSmooth)</i>
------------	--

---

**Description**

histogram panel for pairs function (used as default in mypairsSmooth)

**Usage**

```
panel.hist(x, ...)
```

**Arguments**

x	numeric data
...	additional parameters passed to rect

plotOneSideVolcano      *Plot and filter data coming from one sample single sided t-test*

---

**Description**

Plot and filter data coming from one sample single sided t-test

**Usage**

```
plotOneSideVolcano(data, p.thresh = 0.05, fc.thresh = 2, main = "")
```

**Arguments**

data                    - matrix generated by onesamplegreaterT  
p.thresh                - p value to filter with, default 0.05  
fc.thresh               - fold change threshold to filter with, default 2  
main                    - main title of plot

**Examples**

```
data <- matrix(rnorm(100,1,1), ncol=5)
dim(data)
rownames(data) <- 1:20
resM <- onesamplegreaterT(data)
plotOneSideVolcano(resM)
data <- matrix(rnorm(100,2,1), ncol=5)
dim(data)
rownames(data) <- 1:20
resM <- onesamplegreaterT(data)
plotOneSideVolcano(resM)
```

---

ProgenesisBuildAnnotation  
                          *build annotation from column names*

---

**Description**

build annotation from column names

**Usage**

```
ProgenesisBuildAnnotation(data)
```



**Arguments**

data                    tibble returned by ProgenesisRead

**Value**

list of tibbles data -

**Examples**

```
if(0){
  file = file.path(path.package("quantable"), "extdata/PG/ProteinMeasurement_inclSingleHits_hi3.csv" )
  tmp <- ProgenesisRead(file)
  colnames(tmp)
  xx <- ProgenesisBuildAnnotation(tmp)
  head(xx$anno)
  colnames(xx$data)
  colnames(tmp)
}
```

---

ProgenesisRead	<i>reads file exportet from progenesis ProgenesisRead</i>
----------------	---

---

**Description**

reads file exportet from progenesis ProgenesisRead

**Usage**

```
ProgenesisRead(file, sep = ",")
```

**Arguments**

file                    path to progenesis peptide or protein file  
 sep                    separator used (progenesis uses depending language settings a , or ;)

**Examples**

```
if(0){
  print("example does not run because can't ship large files")
  file = file.path(path.package("quantable"), "extdata/PG/PeptideMeasurement_inclSingleHits_hi3.csv" )
  tmp <- ProgenesisRead(file)
  colnames(tmp)
  head(tmp)

  file = file.path(path.package("quantable"), "extdata/PG/ProteinMeasurement_inclSingleHits_hi3.csv" )
  tmp <- ProgenesisRead(file)
```

```
colnames(tmp)
head(tmp)
}
```

---

pvalCorMat                      *pvalues for correlation matrix -*

---

### Description

Benjamin Hochberg - adjusted tests for significance of correlations among rows. Transpose your matrix if you want to have it among columns

### Usage

```
pvalCorMat(x, alternative = "two.sided", method = "spearman")
```

### Arguments

x	data matrix
alternative	(see cor.test)
method	(see cor.test)

### Examples

```
mat = matrix(rnorm(10*20),ncol=10)
res = pvalCorMat(mat)
image(res$pval)
image(res$cor)
```

---

quantable                      *Streamline descriptive analysis of quantitative data matrices*

---

### Description

Streamline descriptive analysis of quantitative data matrices

---

removeDecorrelated      *remove decorrelated rows*

---

**Description**

remove decorrelated rows

**Usage**

```
removeDecorrelated(ff, corThreshold = 0.7, tr = identity)
```

**Arguments**

ff	matrix or data frame
corThreshold	correlation threshold
tr	intensity transformation to apply

---

removeNArows      *Removes rows with more than thresh NA's from matrix*

---

**Description**

Removes rows with more than thresh NA's from matrix

**Usage**

```
removeNArows(obj, thresh = 0)
```

**Arguments**

obj	matrix or dataframe
thresh	- maximum number of NA's / row - if more the row will be removed

**Value**

matrix

**Examples**

```
x = matrix(rnorm(10*10),ncol=10)
dim(x)
x[3,3] = NA
x = removeNArows(x)
dim(x)
```

---

robustscale	<i>robust scaling uses median an mad instead of mean and row applies the scaling to the columns (samples) by default</i>
-------------	--

---

**Description**

robust scaling uses median an mad instead of mean and row applies the scaling to the columns (samples) by default

**Usage**

```
robustscale(data, dim = 2, center = TRUE, scale = TRUE,
            preserveScale = TRUE)
```

**Arguments**

data	matrix or data.frame
dim	should rows (1) or columns (2:default) be scaled
center	subtract median (default:TRUE)
scale	scale by mad (default:FALSE)
preserveScale	default TRUE , equalize scales but do not change them

**Examples**

```
library(quantable)
tmp = matrix(rep((1:100),times = 4) + rnorm(100*4,0,3),ncol=4)
mean = c(20,30,10,40)
sd = c(4,3,4,5)
tmp = sweep(tmp,2,sd,"*")
tmp = sweep(tmp,2,mean,"+")
boxplot(tmp)
tmp = robustscale(tmp)
boxplot(tmp$data)
```

---

rowNAs	<i>get NR of NA's per matrix or data.frame row</i>
--------	--

---

**Description**

get NR of NA's per matrix or data.frame row

**Usage**

```
rowNAs(x)
```

**Arguments**

x	matrix or data.frame
---	----------------------

---

runFun	<i>running function (default median absolute deviation)</i>
--------	---

---

**Description**

running function (default median absolute deviation)

**Usage**

```
runFun(aref, k = 301, func = mad)
```

**Arguments**

aref	data array
k	window in data points, default 300
func	default med but can be any function taking a vector and returning a summary

**See Also**

[runmed](#)

**Examples**

```
x = rnorm(500)
x = c(x, rnorm(1000, 3, 2))
x = c(x, runif(1000, 4, 6))
y = runFun(x, k=51, func=mad)
hist(y)#[500:490]
y2 = runFun(x, k=51, func=median)
plot(x, pch="*")
lines(y2, col=2, lwd=3)
lines(y2+y, col=3, lwd=3)
lines(y2-y, col=3, lwd=3)
tic = runFun(x, k=51, func=function(x,...){mean(x)})
plot(x, pch=".")
abline(h=0, col=2)
lines(tic, col=3, lwd=3)
```

---

runrobscale	<i>running robust scaling of arefw</i>
-------------	--

---

**Description**

running robust scaling of arefw

**Usage**

```
runrobscale(arefw, k = 101, scale = TRUE)
```

**Arguments**

```
arefw      - data array to scale
k          - windows
scale      - should also scaling be applied
```

**Value**

list with scaled data runmed used to center the data and runmad the running MAD used for scaling

**Examples**

```
res = c(rnorm(1000,0,1),rnorm(2000,4,3))
res2 = runrobscale(res)
par(mfrow=c(2,1))
plot(res,type="p",pch="x",col=1,cex=0.3)
lines(res2$runmed,col=3)

y = runFun( res2$scaled, k=51, func=mad )
#hist(y)
y2 = runFun(res2$scaled,k=51,func=median)
plot(res2$scaled,pch="*")
lines(y2,col=2,lwd=3)
lines(y2+y,col=3,lwd=3)
lines(y2-y,col=3,lwd=3)
```

---

runTICscale

*running total ion count scaling (TIC)*


---

**Description**

running total ion count scaling (TIC)

**Usage**

```
runTICscale(arefw, k = 101)
```

**Arguments**

```
arefw      a series to scale
k          - the smoothing window
```

**Value**

list with fields scaled - contains scaled data and mean - averages of window k

**See Also**

correctIntRTv2 for context

**Examples**

```
res = c(rnorm(1000,3,2),rnorm(2000,8,1))
res2 = runTICscale(res)
plot(res, type="p", pch=".", col=1, cex=0.5)
lines(1:length(res), res2$mean, col=3)
points(res2$scaled, pch=".", cex=3, col=2)
```

---

scaleByGroup

*scale data given group*

---

**Description**

scale data given group

**Usage**

```
scaleByGroup(data, protGroup, plot = FALSE, scale = TRUE, center = TRUE)
```

**Arguments**

data	matrix
protGroup	names matching rownames of data
plot	plot diagnostics
scale	scale default TRUE
center	center default TRUE

---

setdiff\_data.frame     *setdiff for data frames*

---

**Description**

setdiff for data frames

**Usage**

```
setdiff_data.frame(x, y)
```

**Arguments**

x	data.frame
y	data.frame

---

simpleheatmap     *heatmap2 facade*

---

**Description**

heatmap2 facade

**Usage**

```
simpleheatmap(pln, main = "", distf = dist, hclustf = hclust,
  labRow = "", palette = getBlueWhiteRed(), margins = c(5, 5),
  scale = "none", ...)
```

**Arguments**

pln	or dataframe with numerical values
main	title
distf	distance function
hclustf	clustering function
labRow	row labels
palette	color palette
margins	control margins of heatmap
scale	c(row, column or none)
...	other parameters to heatmap.2



**Examples**

```
tmp = matrix(rep((1:100),times = 4) + rnorm(100*4,0,3),ncol=4)
mean = c(20,30,10,40)
sd = c(4,3,4,5)
tmp = sweep(tmp,2,sd,"*")
tmp = sweep(tmp,2,mean,"+")
par(mar=c(5,5,5,5))
simpleheatmap(tmp,ColSideColors=c("red","blue","pink","black"))
simpleheatmap(tmp)
```

---

split2table	<i>splits names and creates a matrix</i>
-------------	--

---

**Description**

splits names and creates a matrix

**Usage**

```
split2table(names, split = "\\|\\_")
```

**Arguments**

names	vector with names
split	patter to use to split

**Value**

matrix

**Examples**

```
dat = c("bla_ra0/2_run0","bla_ra1/2_run0","bla_ra2/2_run0")
split2table(dat,split="\_|\\/")
```

---

sumtop	<i>sums top rows of matrix uses median of row to determine row order</i>
--------	--

---

**Description**

sums top rows of matrix uses median of row to determine row order

**Usage**

```
sumtop(x, top = 3)
```

**Arguments**

x                    a matrix  
 top                 how many to rows to sum (default 3)

**Examples**

```
tmp <- rbind(rep(1,times = 4), rep(2,times=4), rep(3,times=4), rep(4,times=4))
res <- sumtop(tmp)
stopifnot(res == c(9,9,9,9))
```

---

unpivot	<i>unpivot data matrix</i>
---------	----------------------------

---

**Description**

unpivot data matrix

**Usage**

```
unpivot(m)
```

**Arguments**

m                    matrix or dataframe

**Examples**

```
x = matrix(1:25,ncol=5)
x = as.data.frame(x)
colnames(x) = letters[1:5]
rownames(x) = LETTERS[1:5]
unpivot(x)
```

---

uppertriang	<i>get values of upper triangle from matrix</i>
-------------	---

---

**Description**

get values of upper triangle from matrix

**Usage**

```
uppertriang(mat, diag = FALSE)
```

**Arguments**

mat	matrix
diag	default = FALSE

**Examples**

```
t = matrix(1:25,ncol=5)
uppertriang(t)
```

---

volcano2G

*DEPRECATED Volcano plot using ggplot and ggrepel*


---

**Description**

DEPRECATED Volcano plot using ggplot and ggrepel

**Usage**

```
volcano2G(foldchange, pvals, labels, pthresh = 0.1, log2FCThresh = 0.5,
  main = NULL, xlab = "log2 FC", ylab = "-log10(Q Value)", xlim = c(-5,
  5), ylim = c(0, -log10(min(pvals, na.rm = TRUE))), size = 1,
  segment.size = 0.3, segment.alpha = 0.3, pseudo = NULL,
  colors = NULL)
```

**Arguments**

foldchange	vector with fold changes
pvals	vector with pvalues
labels	vector with labels
pthresh	pvalue threshold
log2FCThresh	log2 FC threshold
main	main title
xlab	xlab
ylab	ylab
xlim	xlim
ylim	ylim
size	see geom_text_repel
segment.size	see geom_text_repel
segment.alpha	see geom_text_repel
pseudo	usually q.mod containing NAs
colors	specify colors for specific lables.

**Examples**

```
rm(list=ls())
library(quantable)
foldchange <- rnorm(1000)
pvals <- rexp(1000)
volcano2G(foldchange, pvals, labels=rep("abcde", length(pvals)),
  pthresh=0.1, log2FCThresh=0.5, main='test', size=2, segment.size=0.3)
```

---

volcano2GB

*Volcano with more control*


---

**Description**

Volcano with more control

**Usage**

```
volcano2GB(dataX, foldchange = "log2FC", pvalue = "q.mod",
  labels = "names", pthresh = 0.1, log2FCThresh = 0.5, main = NULL,
  xlab = "log2 FC", ylab = "-log10(Q Value)", repel.text.size = 1,
  repel.segment.size = 0.5, repel.segement.alpha = 0.5, pseudo = NULL)
```

**Arguments**

dataX	dataX frame
foldchange	column name with fold change plotted on X
pvalue	column with pvalue or qvalue plotted as -log10 on y axes
labels	column containing lables
pthresh	horizontal abline
log2FCThresh	vertical abline
main	main plot title
xlab	xlab
ylab	ylab
repel.text.size	ggrepel parameter
repel.segment.size	ggrepel parameter
repel.segement.alpha	ggrepel parameter
pseudo	add pseudo fold changes

**Examples**

```
rm(list=ls())

library(ggrepel)
library(quantable)
foldchange <- rnorm(1000)
pvals <- rexp(1000)
names <- sample(colors(),1000,replace=TRUE)

dataX <- data.frame(q.mod = pvals,
  log2FC = foldchange,
  names = names )

volcano2GB(dataX)
b <- volcano2GB(dataX, pthresh=0.1, log2FCthresh=0.5 ,
  main='test', repel.segment.size=0.3,repel.text.size=2)
b
```

---

volcanoplot

*volcano plot*


---

**Description**

volcano plot

**Usage**

```
volcanoplot(foldchange, pvals, pthresh = 0.05, foldchangethresh = 1,
  xlab = "log2(T/N)", ylab = "-log10(P)", labels = NULL, cex = 0.6,
  cex.point = 1, xlim = NULL, main = NULL, biasAdjust = FALSE)
```

**Arguments**

foldchange	- fold change values
pvals	pvalues
pthresh	pvalue threshold
foldchangethresh	threshold of foldchange
xlab	- x axis label
ylab	- y axis label
labels	- optional labels
cex	size of labels
cex.point	- point size
xlim	- xlim
main	- main title
biasAdjust	- if bias in foldchanges exists (i.e. if median of fold changes does not equal 0) you can use this option to adjust for it.

**Examples**

```

library(quantable)
foldchange <- rnorm(1000)
pvals <- rexp(1000)
length(foldchange)
length(pvals)
volcanoplot(foldchange, pvals, pthresh=0.1, foldchangethresh=1,main='test')
volcanoplot(foldchange, pvals, pthresh=0.1,
  foldchangethresh=1,main='test', labels=rep("abcde", length(pvals)))

volcanoplot(foldchange, pvals,pthresh=0.1, foldchangethresh=3,main='test')
abline(v=0.05,col=2)

```

---

write.tab	<i>write table in tab delimited no quotes no row.names (usefull for exporting i.e. list of foldchanges) uses write.table TODO : validate if not redundant with other write functions</i>
-----------	--

---

**Description**

write table in tab delimited no quotes no row.names (usefull for exporting i.e. list of foldchanges) uses write.table TODO : validate if not redundant with other write functions

**Usage**

```
write.tab(x, file)
```

**Arguments**

x	vector to write
file	file to write to

---

write.vector	<i>write vectors as single column table (usefull for exporting i.e. protein id's) uses write.table</i>
--------------	--

---

**Description**

write vectors as single column table (usefull for exporting i.e. protein id's) uses write.table

**Usage**

```
write.vector(x, file)
```

**Arguments**

x	vector to write
file	file to write to

---

xxx_replace_xxx	<i>replace patterns (vector) with replacements (vector) in string or string vector. uses gsub and perl=TRUE taken from <a href="https://stackoverflow.com/questions/26676045">https://stackoverflow.com/questions/26676045</a></i>
-----------------	--

---

**Description**

replace patterns (vector) with replacements (vector) in string or string vector. uses gsub and perl=TRUE taken from <https://stackoverflow.com/questions/26676045>

**Usage**

```
xxx_replace_xxx(string, patterns, replacements)
```

**Arguments**

string	string or vector of strings to replace in
patterns	pattern or vector of patterns to replace
replacements	replacements string, needs to have same length as replacement

# Index

addSpecialProteins, 3  
altmanbland, 4

colNAs, 4  
CV, 5  
CVlog, 5

dat2Matrix, 6  
determineCut, 6  
distmy, 7

filterSignificant, 8  
fisherExact, 9

getBlueScale, 9  
getBlueWhiteRed, 10  
getDiv, 10  
getGreensScale, 11  
getRedScale, 11  
getTValuesForVolcano, 12  
getWRValuesForVolcano, 13

image\_nan, 16  
imageColorscale, 13  
imageWithLabels, 14  
imageWithLabelsNoLayout, 15

jackknifeMatrix, 17

makeROCplot, 18  
matrix\_to\_tibble, 18  
multigroupFCDATA, 19  
multigroupVolcano, 19  
my\_jackknife, 21  
mypairs, 20  
mypairsSmooth, 20

onesamplegreaterT, 22

pairs, 20–22  
pairsQQ, 22

panel.cor, 23  
panel.hist, 23  
plotOneSideVolcano, 24  
ProgenesisBuildAnnotation, 24  
ProgenesisRead, 25  
pvalCorMat, 26

qqplot, 22  
quantable, 26  
quantable-package (quantable), 26

removeDecorrelated, 27  
removeNArows, 27  
robustscale, 28  
rowNAs, 28  
runFun, 29  
runmed, 29  
runrobustscale, 29  
runTICscale, 30

scaleByGroup, 31  
setdiff\_data.frame, 32  
simpleheatmap, 32  
split2table, 33  
sumtop, 33

unpivot, 34  
uppertriang, 34

volcano2G, 35  
volcano2GB, 36  
volcanoplot, 37

write.tab, 38  
write.vector, 38

xxx\_replace\_xxx, 39