Package: Tmisc (via r-universe)

September 2, 2024

Title Turner Miscellaneous

Version 1.1.0

Description Miscellaneous utility functions for data manipulation, data tidying, and working with gene expression data and biological sequence data.

URL https://github.com/stephenturner/Tmisc, https://stephenturner.github.io/Tmisc/

BugReports https://github.com/stephenturner/Tmisc/issues

Depends R (>= 4.2.0)

Imports dplyr, tibble, utils, rstudioapi, methods, stats

License GPL-3

LazyData true

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Suggests ggplot2, reshape2

Repository https://stephenturner.r-universe.dev

RemoteUrl https://github.com/stephenturner/Tmisc

RemoteRef HEAD

RemoteSha fd4db9df83cd8232fa2c736eb09159fa08464d69

Contents

`addins` *Insert text at current position.*

Description

Call these function as an addin to insert desired text at the cursor position. After installing Tmisc, hit the Addins menu, and optionally add a keyboard shortcut, e.g., Command+Shift+I, Alt+-, etc.

`are_all_equal` *Are all equal?*

Description

Are all the elements of a numeric vector (approximately) equal?

Usage

```
are_all_equal(x, na.rm = FALSE)
```

Arguments

`x`	A numeric vector.
`na.rm`	Remove missing values (FALSE by default; NAs in x will return NA).

Value

Logical, whether all elements of a numeric vector are equal.

Examples

```
are_all_equal(c(5,5,5))
are_all_equal(c(5,5,5,6))
are_all_equal(c(5,5,5,NA,6))
are_all_equal(c(5,5,5,NA,6), na.rm=TRUE)
5==5.000000001
identical(5, 5.000000001)
are_all_equal(c(5L, 5, 5.000000001))
```

`corner` *Print the top left corner of a data frame*

Description

Prints the first n rows and columns of a data frame or matrix.

Usage

```
corner(x, n = 5)
```

Arguments

`x`	A data.frame.
`n`	The number of rows/columns to print.

Value

The corner of the data frame

Examples

```
corner(mtcars)
corner(iris, n=4)
```

counts2fpkm *Fragments per kilobase per million*

Description

Takes a count matrix and a vector of gene lengths and returns an optionally log2-transformed FPKM matrix. Modified from edgeR.

Usage

```
counts2fpkm(x, length, log = FALSE, prior.count = 0.25)
```

Arguments

x	a matrix of counts.
length	a vector of length nrow(x) giving length in bases.
log	logical, if TRUE, then log2 values are returned.
prior.count	average count to be added to each observation to avoid taking log of zero. Used only if log=TRUE.

Value

A matrix of FPKM values.

Examples

```
set.seed(123)
genecounts <- matrix(sample(c(rep(0, 50), 1:100), 30), nrow=10)
lengths <- sample(1000:10000, 10)
counts2fpkm(genecounts, lengths)
```

ellipses *Truncate a data frame with ellipses.*

Description

Prints the specified number of rows of a data frame, followed by a row of ellipses. Useful for piping to knitr::kable() for printing a truncated table in a markdown document.

Usage

```
ellipses(df, n = 5L)
```

Arguments

df	A data frame.
n	The number of rows to show before an ellipses row.

Value

A data frame truncated by a row of ellipses.

Examples

```
ellipses(mtcars, 5)
```

---

fisherp *Fisher's method to combine p-values*

Description

Uses Fisher's method to combine p-values from different tests.

Usage

```
fisherp(x)
```

Arguments

x	A vector of p-values between 0 and 1.

Value

The combined p-value.

References

Fisher, R.A. (1925). Statistical Methods for Research Workers.

https://en.wikipedia.org/wiki/Fisher%27s_method.

Examples

```
fisherp(c(.042, .02, .001, 0.01, .89))
```

gghues *Emulate ggplot2 default hues*

Description

This will emulate ggplot2's hues, which are equally spaced hues around the color wheel, starting from 15.

Usage

```
gghues(n, start = 15)
```

Arguments

n	The Numeric; number of hues to generate.
start	Numeric; the place on the color wheel to start. ggplot2 default is 15.

Value

A vector of hues

Examples

```
n <- 10
gghues(3)
barplot(rep(1,n), col=gghues(n), names=gghues(n))
barplot(rep(1,n), col=gghues(n, start=15+180), names=gghues(n, start=15+180))
```

gg_na *Plot missing data*

Description

Plots missing data as holes on a black canvas.

Usage

```
gg_na(df)
```

Arguments

df	A data frame

gt2refalt *Two-letter genotype from VCF GT*

Description

Get a two-letter genotype from a VCF GT field. Current implementation is quick and dirty, and only accepts 0/0, 0/1, or 1/1. Any other input to gt will return a missing value.

Usage

```
gt2refalt(gt, ref, alt)
```

Arguments

gt	The genotype field (must be 0/0, 0/1, or 1/1).
ref	The reference allele.
alt	The alternate allele.

Value

Returnvalue

Examples

```
gt2refalt(gt="0/0", ref="R", alt="A")
gt2refalt(gt="0/1", ref="R", alt="A")
gt2refalt(gt="1/1", ref="R", alt="A")
gt2refalt(gt="0/2", ref="R", alt="A")
gt2refalt(gt="./.", ref="R", alt="A")
```

jsd *Jensen-Shannon divergence*

Description

Calculates a distance matrix from a matrix of probability distributions using Jensen-Shannon divergence. Adapted from https://enterotype.embl.de/.

Usage

```
jsd(M, pseudocount = 1e-06, normalizeCounts = FALSE)
```

Arguments

M	a probability distribution matrix, e.g., normalized transcript compatibility counts.
pseudocount	a small number to avoid division by zero errors.
normalizeCounts	logical, whether to attempt to normalize by dividing by the column sums. Set to TRUE if this is, e.g., a count matrix.

Value

A Jensen-Shannon divergence-based distance matrix.

References

https://web.archive.org/web/20240131141033/https://enterotype.embl.de/enterotypes.html#dm.

Examples

```
set.seed(42)
M <- matrix(rpois(100, lambda=100), ncol=5)
colnames(M) <- paste0("sample", 1:5)
rownames(M) <- paste0("gene", 1:20)
Mnorm <- apply(M, 2, function(x) x/sum(x))
Mjsd <- jsd(Mnorm)
# equivalently
Mjsd <- jsd(M, normalizeCounts=TRUE)
Mjsd
plot(hclust(Mjsd))
```

---

lowestnonzero *Lowest nonzero values*

Description

Sometimes want to plot p-values (e.g., volcano plot or MA-plot), but if a statistical test returns a zero p-value, this causes problems with visualization on the log scale. This function returns a vector where the zero values are equal to the smallest nonzero value in the vector.

Usage

```
lowestnonzero(x)
```

Arguments

x	A vector of p-values between 0 and 1.

Value

A vector of p-values where zero values are exchanged for the lowest non-zero p-value in the original vector.

Examples

```
lowestnonzero(c(.042, .02, 0, .001, 0, .89))
```

`lsa` *Improved list of objects*

Description

Improved list of objects. Sorts by size by default. Adapted from https://stackoverflow.com/q/1358003/654296.

Usage

```
lsa(
  pos = 1,
  pattern,
  order.by = "Size",
  decreasing = TRUE,
  head = FALSE,
  n = 10
)
```

Arguments

`pos`	numeric. Position in the stack.
`pattern`	Regex to filter the objects by.
`order.by`	character. Either 'Type', 'Size', 'PrettySize', 'Rows', or 'Columns'. This will dictate how the output is ordered.
`decreasing`	logical. Should the output be displayed in decreasing order?
`head`	logical. Use head on the output?
`n`	numeric. Number of objects to display is head is TRUE.

Value

A data.frame with type, size in bytes, human-readable size, rows, and columns of every object in the environment.

Examples

```
a <- rnorm(100000)
b <- matrix(1, 1000, 100)
lsa()
```

---

mat2df *Matrix to pairwise data frame*

Description

Turns a distance matrix into a data frame of pairwise distances.

Usage

```
mat2df(M)
```

Arguments

M	a square pairwise matrix (e.g., of distances).

Value

Data frame with pairwise distances.

Examples

```
M <- matrix(1:25, nrow=5, dimnames=list(letters[1:5], letters[1:5]))
M
```

Mode *Mode*

Description

Returns the mode of a vector. First in a tie wins (see examples).

Usage

```
Mode(x, na.rm = FALSE)
```

Arguments

x	A vector.
na.rm	Remove missing values before calculating the mode (FALSE by default). NAs are counted just like any other element. That is, an NA in the vector won't necessarily result in a return NA. See the first example.

Value

The mode of the input vector.

Examples

```
Mode(c(1,2,2,3,3,3, NA))
Mode(c(1,2,2,3,3,3, NA), na.rm=TRUE)
Mode(c(1,2,2,3,3,3, NA, NA, NA, NA))
Mode(c(1,2,2,3,3,3, NA, NA, NA, NA), na.rm=TRUE)
Mode(c("A", "Z", "Z", "B", "B"))
```

---

nn *Get names and class of all columns in a data frame*

Description

Get names and class of all columns in a data frame in a friendly format.

Usage

```
nn(df)
```

Arguments

df	A data frame.

Value

A data frame with index and class.

Examples

```
nn(iris)
```

o *Open the current working directory on mac*

Description

Open the current working directory on mac

Usage

```
o()
```

Examples

```
## Not run:
o()

## End(Not run)
```

peek *Peek at the top of a text file*

Description

This returns a character vector which shows the top n lines of a file.

Usage

```
peek(x, n = 5)
```

Arguments

x	a filename
n	the number of lines to return

Value

A character vector of the first n lines of the file.

Examples

```
## Not run:
filename <- tempfile()
x <- matrix(round(rnorm(10^4), 2), 1000, 10)
colnames(x) <- letters[1:10]
write.table(x, file = filename, row.names = FALSE, quote = FALSE)
peek(filename)

## End(Not run)
```

quartet *Anscombe's Quartet data (tidy)*

Description

Tidy version of built-in Anscombe's Quartet data. Four datasets that have nearly identical linear regression properties, yet appear very different when graphed.

Usage

```
quartet
```

Format

Data frame with three columns, set, x, y.

revcomp *Reverse complement*

Description

Reverse complements a sequence.

Usage

```
revcomp(x)
```

Arguments

x	A sequence to reverse complement

Value

The sequence, reverse complemented

Examples

```
revcomp("GATTACA")
sapply(c("GATTACA", "CATATTAC"), revcomp)
```

saveit *Rename objects while saving.*

Description

Allows you to rename objects as you save them. See https://stackoverflow.com/a/21248218/654296.

Usage

```
saveit(..., file = stop("'file' must be specified"))
```

Arguments

...	Objects to save.
file	Filename/path where data will be saved.

Examples

```
## Not run:
foo <- 1
saveit(bar=foo, file="foobar.Rdata")

## End(Not run)
```

sicb *Write sessionInfo to the clipboard*

Description

Writes output of `sessionInfo()` to the clipboard. Only works on Mac.

Usage

```
sicb()
```

Examples

```
## Not run:
sicb()

## End(Not run)
```

strSort *Sort characters in a string*

Description

Alphabetically sorts characters in a string. Vectorized over x.

Usage

```
strSort(x)
```

Arguments

x	A string to sort.

Value

A sorted string.

Examples

```
strSort("cba")
strSort("zyxcCbB105.a")
strSort(c("cba", "zyx"))
strSort(c("cba", NA))
```

Thist *Histograms with overlays*

Description

Plot a histogram with either a normal distribution or density curve overlay.

Usage

```
Thist(x, overlay = "normal", col = "gray80", ...)
```

Arguments

x	A numeric vector.
overlay	Either "normal" (default) or "density" indicating whether a normal distribution or density curve should be plotted on top of the histogram.
col	Color of the histogram bars.
...	Other arguments to be passed to hist().

Examples

```
set.seed(42)
x <- rnorm(1000, mean=5, sd=2)
Thist(x)
Thist(x, overlay="density")
Thist(x^2)
Thist(x^2, overlay="density", breaks=50, col="lightblue2")
```

Tpairs *Better scatterplot matrices*

Description

A matrix of scatter plots with rugged histograms, correlations, and significance stars. Much of the functionality borrowed from `PerformanceAnalytics::chart.Correlation()`.

Usage

```
Tpairs(x, histogram = TRUE, gap = 0, ...)
```

Arguments

x	A numeric matrix or data.frame.
histogram	Overlay a histogram on the diagonals?
gap	distance between subplots, in margin lines.
...	arguments to be passed to or from other methods.

Examples

```
Tpairs(iris[-5])
Tpairs(iris[-5], pch=21, bg=gghues(3)[factor(iris$Species)], gap=1)
```

%like% *x like y*

Description

Returns a logical vector of elements of x matching the regex y.

Usage

```
x %like% pattern
```

Arguments

x	a vector (numeric, character, factor)
pattern	a vector (numeric, character, factor), matching the mode of x

Value

A logical vector with length equal to x of things in x that are like y.

See Also

%like%, %nlike%, %nin%,

Examples

```
(Name <- c("Mary","George","Martha"))
Name %in% c("Mary")
Name %like% "^Mar"
Name %nin% c("George")
Name %nlike% "^Mar"
```

%nin% *x not in y*

Description

Returns a logical vector of elements of x that are not in y.

Usage

```
x %nin% table
```

Arguments

x	a vector (numeric, character, factor)
table	a vector (numeric, character, factor), matching the mode of x

Value

A logical vector with length equal to x of things in x that aren't in y.

See Also

%like%, %nlike%, %nin%,

Examples

```
1:10 %nin% seq(from=2, to=10, by=2)
c("a", "b", "c") %nin% c("a", "b")
```

`%nlike%` *x not like y*

Description

Returns a logical vector of elements of x not matching the regex y.

Usage

```
x %nlike% pattern
```

Arguments

x	a vector (numeric, character, factor)
pattern	a vector (numeric, character, factor), matching the mode of x

Value

A logical vector with length equal to x of things in x that aren't like y.

See Also

%like%, %nlike%, %nin%,

Examples

```
(Name <- c("Mary","George","Martha"))
Name %in% c("Mary")
Name %like% "^Mar"
Name %nin% c("George")
Name %nlike% "^Mar"
```

Index