

# Package: bayesbio (via r-universe)

September 6, 2024

**Title** Miscellaneous Functions for Bioinformatics and Bayesian Statistics

**Version** 1.0.1.9000

**Description** A hodgepodge of hopefully helpful functions. Two of these perform shrinkage estimation: one using a simple weighted method where the user can specify the degree of shrinkage required, and one using James-Stein shrinkage estimation for the case of unequal variances.

**Depends** R (>= 3.2.0)

**Suggests** ggplot2, RISmed, rentrez, testthat

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 5.0.1

**Repository** <https://andymckenzie.r-universe.dev>

**RemoteUrl** <https://github.com/andymckenzie/bayesbio>

**RemoteRef** HEAD

**RemoteSha** 1389283ba9ac8e1778dd7930af35e719a3baf540

## Contents

allDups . . . . .	2
a_hat_mle . . . . .	2
bayesbio . . . . .	3
cbindFill . . . . .	3
covariatesTable . . . . .	4
createStrings . . . . .	4
ggHorizBar . . . . .	5
jaccardSets . . . . .	6
makeMatSym . . . . .	6
mgsub . . . . .	7
nearestTime . . . . .	7

nearestTimeandID . . . . .	8
p.adjust.nlp . . . . .	8
pubmedQuery . . . . .	9
std_error . . . . .	10
subsupDiag . . . . .	10
unequalVarShrink . . . . .	11
weightedShrink . . . . .	11
write.delim . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

allDups	<i>Identify all duplicates values in a vector.</i>
---------	--

---

### Description

By default the base R function duplicated only identifies the duplicated values after the first in a vector as TRUE. This function identifies all of the duplicates as true.

### Usage

```
allDups(x)
```

### Arguments

x	The input vector.
---	-------------------

### Value

A logical vector.

---

a_hat_mle	<i>Likelihood function of the James-Stein shrinkage factor.</i>
-----------	---

---

### Description

To be used in MLE computation of the James-Stein shrinkage factor.

### Usage

```
a_hat_mle(stat, vars, a_hat)
```

### Arguments

stat	Input statistics to be shrinkage estimated.
vars	Corresponding variances of equal length.
a_hat	Shrinkage intensity to be estimated.

**Value**

The likelihood of the function given the parameters.

**References**

<http://projecteuclid.org/euclid.ss/1331729986>

---

bayesbio	<i>bayesbio: Miscellaneous functions useful in bioinformatics and Bayesian statistics</i>
----------	---

---

**Description**

A hodgepodge of hopefully helpful functions. Two of these perform shrinkage estimation: one using a simple weighted method where the user can specify the degree of shrinkage required, and one using James-Stein shrinkage estimation for the case of unequal variances.

---

cbindFill	<i>cbind while converting missing entries to NA.</i>
-----------	--

---

**Description**

cbind usually malfunctions on vector of unequal lengths; this function allows vectors of unequal length to be combined, while filling the missing entries with NAs.

**Usage**

```
cbindFill(...)
```

**Arguments**

... A set of vectors separated by commas.

**Value**

A matrix that combines the inputted vectors.

**References**

<http://r.789695.n4.nabble.com/How-to-join-matrices-of-different-row-length-from-a-list-td3177212.html>;  
<http://stackoverflow.com/a/7962286/560791>

---

covariatesTable	<i>Create a table summarizing covariates segregated by levels of a diagnosis.</i>
-----------------	---

---

### Description

Take a data frame with a diagnosis column and a number of covariate columns and specify the percentage of specified covariate levels in each group and/or the mean +/- sd for quantitative variables for each covariate desired. Although it was designed for generating sample summary tables in the context of bioinformatics experiments and the terminology refers to this, it can be used more generally as well.

### Usage

```
covariatesTable(df, dg_col, percent_cols = NULL, quant_cols = NULL,
  percent_col_cats = NULL, group_names = NULL, row_names = NULL)
```

### Arguments

df	The data frame containing the columns to be extracted, both diagnosis and covariates.
dg_col	Column specifying the diagnosis column name, which is used to split the table. Levels of this value will be used to generate
percent_cols	Character vector of column names specifying the
percent_col_cats	Character vector specifying the values for which the percentage should be calculated for each percent column.
group_names	Optional character vector specifying the groups within the dg_col, which will be used to order the resulting table.
row_names	Optional character vector specifying what the rownames of the resulting table should be.

### Value

A table summarizing the covariates.

---

createStrings	<i>Creates random, unique character strings.</i>
---------------	--

---

### Description

Makes them unique by randomly choosing the character strings; and, in case it is necessary, adding numbers to the end using make.unique.

**Usage**

```
createStrings(number, length, upper = FALSE)
```

**Arguments**

number	Specifies the number of character strings that should be created.
length	Specifies the length of each character string in letters.
upper	Binary parameter specifying whether the character strings should be uppercase. Default = FALSE, so the character strings are all lowercase.

**References**

<http://stackoverflow.com/a/1439541/560791>

---

ggHorizBar	<i>Create a color-labeled horizontal bar plot in ggplot2.</i>
------------	---

---

**Description**

This function takes a data frame and creates a horizontal (by default) bar plot from it while ordering the values.

**Usage**

```
ggHorizBar(data_df, dataCol, namesCol, labelsCol, decreasing = TRUE)
```

**Arguments**

data_df	Data frame with columns to specify the data values, the row names, and the fill colors of each of the bars.
dataCol	The column name that specifies the values to be plotted.
namesCol	The column name that specifies the corresponding names for each of the bar plots to be plotted.
labelsCol	The column name that specifies the groups of the labels.
decreasing	Logical specifying whether the values in dataCol should be in decreasing order.

**Value**

A ggplot2 object, which can be plotted via the plot() function or saved via the ggsave() function.

---

jaccardSets	<i>Jaccard index of two character vectors.</i>
-------------	--

---

**Description**

This function compares the elements in two character vectors to find the Jaccard index, i.e. the number of intersections divided by the total number of elements in both sets.

**Usage**

```
jaccardSets(set1, set2)
```

**Arguments**

set1	Character vector.
set2	Character vector.

**Value**

A number (one-element numeric vector) specifying the Jaccard index from comparing the two sets.

**References**

[https://en.wikipedia.org/wiki/Jaccard\\_index](https://en.wikipedia.org/wiki/Jaccard_index)

---

makeMatSym	<i>Replace the upper or lower triangle of a matrix with the other to make it symmetric.</i>
------------	---

---

**Description**

The replaced values will be lost following the operation of this function.

**Usage**

```
makeMatSym(mat, replaceUpper = TRUE)
```

**Arguments**

mat	The matrix to be made symmetric.
replaceUpper	Whether the upper triangle of the matrix should be replaced by the lower triangle. Default = TRUE; if FALSE, the lower triangle of the matrix is replaced by the upper triangle.

**Value**

A matrix that has been made symmetric.

---

mgsub	<i>Multiple pattern gsub.</i>
-------	-------------------------------

---

### Description

An extension to gsub that handles vectors of patterns and replacements, avoiding recursion problems associated with overlap at the expense of computation time.

### Usage

```
mgsub(pattern, replacement, x, ...)
```

### Arguments

pattern	Character vector of patterns to match.
replacement	Character vector of replacements for each pattern.
x	Character vector in which the gsub should be performed.
...	Additional arguments to grep.

### References

<http://stackoverflow.com/a/15254254/560791>

---

nearestTime	<i>Merge data frames based on the nearest datetime differences.</i>
-------------	---

---

### Description

Takes two data frames each with time/date columns in date-time or date format (i.e., able to be compared using the function difftime), finds the rows of df2 that minimize the absolute value of the datetime for each of the rows in df1, and merges the corresponding rows of df2 into df1 for downstream processing.

### Usage

```
nearestTime(df1, df2, timeCol1, timeCol2)
```

### Arguments

df1	Data frame containing the dates for which the differences between the other data frame's date column should be minimized for each row.
df2	Data frame containing the dates which should be compared to, as well as other values that should be merged to df1 per minimized date time.
timeCol1	Character vector specifying the date/time column in df1.
timeCol2	Character vector specifying the date/time column in df2.

**Value**

A merged data frame that minimizes datetime differences.

---

nearestTimeandID	<i>Merge data frames based on the nearest datetime differences and an ID column. Also removes duplicate column names from the result.</i>
------------------	---

---

**Description**

Takes two data frames each with time/date columns in date-time or date format (i.e., able to be compared using the function `difftime`), finds the rows of `df2` that minimize the absolute value of the datetime for each of the rows in `df1`, and merges the corresponding rows of `df2` into `df1` for downstream processing.

**Usage**

```
nearestTimeandID(df1, df2, timeCol1, timeCol2, IDcol)
```

**Arguments**

<code>df1</code>	Data frame containing the dates for which the differences between the other data frame's date column should be minimized for each row.
<code>df2</code>	Data frame containing the dates which should be compared to, as well as other values that should be merged to <code>df1</code> per minimized date time.
<code>timeCol1</code>	Character vector specifying the date/time column in <code>df1</code> .
<code>timeCol2</code>	Character vector specifying the date/time column in <code>df2</code> .
<code>IDcol</code>	Must be unique by row in <code>df1</code> . Multiple versions are allowed (and expected at least in some rows, as that is the point of the function) in <code>df2</code> .

**Value**

A merged data frame that minimizes datetime differences.

---

p.adjust.nlp	<i>Adjust p-values where n is less than p.</i>
--------------	--

---

**Description**

This function recapitulates `p.adjust` but allows the number of hypothesis tests `n` to be less than the number of p-values `p`. Statistical properties of the p-value adjustments may not hold.

**Usage**

```
p.adjust.nlp(p, method = p.adjust.methods, n = length(p))
```



**Arguments**

p	Numeric vector of p-values.
method	Correction method.
n	Number of comparisons to be made.

**References**

<http://stackoverflow.com/a/30110186/560791>

---

pubmedQuery	<i>Perform PubMed queries on 2x2 combinations of term vectors.</i>
-------------	--

---

**Description**

Perform PubMed queries on the intersections of two character vectors. This function is a wrapper to `RISmed::EUtilsSummary` with `type = 'esearch'`, `db = 'pubmed'`.

**Usage**

```
pubmedQuery(rowTerms, colTerms, sleepTime = 0.01, ...)
```

**Arguments**

rowTerms	Character vector of terms that should make up the rows of the resulting mention count data frame.
colTerms	Character vector of terms for the columns.
sleepTime	How much time (in seconds) to sleep between successive PubMed queries. If you set this too low, PubMed may shut down your connection to prevent overloading their servers.
...	Additional arguments to <code>RISmed::EUtilsSummary</code>

**Value**

A data frame of the number of mentions for each combination of terms.

---

std_error	<i>Find the standard error of the sampling distribution of a statistic.</i>
-----------	---

---

**Description**

Finds the standard error of a numeric vector (i.e., the standard deviation divided by the square root of the sample size); by default, removes NAs prior to calculation.

**Usage**

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

**Arguments**

x	The numeric vector whose standard error should be calculated.
na.rm	Logical; TRUE indicates that NAs should be removed from the vector prior to calculating the standard error, and vice versa for FALSE.

**Value**

A one-element numeric vector giving the standard error.

---

subsupDiag	<i>Add values to the super- and sub-diagonals of a matrix.</i>
------------	--

---

**Description**

Takes a matrix and adds values to the values that are one above the diagonal (ie the superdiagonal) and the values that are one below the diagonal (ie the subdiagonal).

**Usage**

```
subsupDiag(matrix, x)
```

**Arguments**

matrix	Matrix whose super- and sub-diagonals values should be replaced.
x	Numeric vector used to replace values in the matrix. If the inputted vector is not of the same length as both the super- and sub-diagonals of the matrix, then short vector recycling will occur (e.g., x can be one value to replace all of the super- and sub-diagonals of the matrix with that one value).

**Value**

The original matrix with the values added.

**References**

<http://stackoverflow.com/a/9885186/560791>

---

unequalVarShrink      *Perform James-Stein shrinkage estimation using unequal variances*

---

**Description**

Traditional JS shrinkage estimation assumes equal variances for each of the data points, while this algorithm extends JS shrinkage estimation to entries with different variances.

**Usage**

```
unequalVarShrink(stat, vars, verbose = TRUE)
```

**Arguments**

stat	Input statistics to be shrinkage estimated.
vars	Corresponding variances of equal length.
verbose	Whether information about the algorithm should be reported.

**Value**

A data frame containing the shrinkage estimated statistics.

**References**

<http://projecteuclid.org/euclid.ss/1331729986>

---

weightedShrink      *Weighted shrinkage estimation.*

---

**Description**

Shrink values towards the mean (in the sample or the overall cohort) to an inverse degree to the confidence you assign to that observation.

**Usage**

```
weightedShrink(x, n, m = NULL, meanVal = NULL)
```

**Arguments**

x	Numeric vector of values to be shrunken towards the mean.
n	Numeric vector with corresponding entries to x, specifying the number of observations used to calculate x, or some other confidence weight to associate with x.
m	Number specifying weight of the shrinkage estimation, relative to the number of observations in the input vector n. Defaults to the minimum of n, but this is an arbitrary value and should be explored to find an optimal value for your use case.
meanVal	Number specifying the overall mean towards which the values should be shrunken. Defaults to NULL, in which case it is calculated as the (non-weighted) arithmetic mean of the values in the inputted vector x.

**Value**

A numeric vector with shrunken data values.

**References**

<http://math.stackexchange.com/a/41513>

---

write.delim	<i>Write a data frame to a file with delimiter style.</i>
-------------	---

---

**Description**

A wrapper function for write.table that has the same options as read.delim.

**Usage**

```
write.delim(df, file, row.names = FALSE, col.names = TRUE, sep = "\t",
            quote = FALSE, ...)
```

**Arguments**

df	Data frame to be written.
file	Full or relative path to file to be written.
row.names	Logical indicating whether to include row names.
col.names	Logical indicating whether to include column names.
sep	Deliter to separate fields in the resulting file. Default is tab separation.
quote	Logical indicating whether to put quotes around the resulting values.
...	Additional arguments to write.table.

**Value**

None; side-effect is to write to a file.

# Index

[a\\_hat\\_mle](#), 2  
[allDups](#), 2

[bayesbio](#), 3  
[bayesbio-package \(bayesbio\)](#), 3

[cbindFill](#), 3  
[covariatesTable](#), 4  
[createStrings](#), 4

[ggHorizBar](#), 5

[jaccardSets](#), 6

[makeMatSym](#), 6  
[mgsub](#), 7

[nearestTime](#), 7  
[nearestTimeandID](#), 8

[p.adjust.nlp](#), 8  
[pubmedQuery](#), 9

[std\\_error](#), 10  
[subsupDiag](#), 10

[unequalVarShrink](#), 11

[weightedShrink](#), 11  
[write.delim](#), 12