Package ‘BinMat’

March 6, 2020

Type Package

Title Processes Binary Data Obtained from Fragment Analysis

Version 0.1.2

Description A molecular genetics tool that processes binary data from fragment analysis, such as inter-simple sequence repeats (ISSRs) and amplified fragment length polymorphism (AFLP). It consolidates replicate sample pairs, outputs summary statistics, and produces hierarchical clustering trees and nMDS plots. This package was developed from the M.Sc. thesis entitled ``A genetic analysis of the species and intraspecific lineages of Dactylopius Costa (Hemiptera:Dactylopiidae)'' (van Steenderen, 2019, Rhodes University, Department of Zoology and Entomology, Center for Biological Control (CBC) <https://www.ru.ac.za/centreforbiologicalcontrol/>, Grahamstown, South Africa), <doi:10.13140/RG.2.2.28470.86083>. The GUI version of this package is available on the R Shiny online server at: <https://clarkevansteenderen.shinyapps.io/BINMAT/> , or it is accessible via GitHub by typing: shiny::runGitHub(``BinMat'', ``CJMvS'') into the console in R. Please see the vignette supplied with the package for a worked example, and detailed explanations of functions.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Depends R (>= 3.0)

Imports pvclust (>= 2.0), magrittr, MASS (>= 7.3), stats (>= 3.4.0), graphics (>= 3.4.0), base (>= 3.4.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Clarke van Steenderen [aut, cre] (<https://orcid.org/0000-0002-4219-446X>)

Maintainer Clarke van Steenderen `<vsteenderen@gmail.com>`

Repository CRAN

Date/Publication 2020-03-06 18:20:06 UTC

R topics documented:

BinMatInput_ordination *Example input data containing a consolidated binary matrix with grouping information*

Description

Example input data containing a consolidated binary matrix with grouping information

Usage

```
data(BinMatInput_ordination)
```

Format

A dataframe with columns for loci, and rows for samples. Grouping information is in the second column.

Examples

```
data(BinMatInput_ordination)
mat = BinMatInput_ordination
group.names(mat)
scree(mat)
shepard(mat)
clrs = c("red", "green", "black")
shp = c(16,16,16)
nmds(mat, colours = clrs, shapes = shp, labs = TRUE)
```

BinMatInput_reps *Example input data containing a binary matrix comprising replicate sample pairs*

Description

Example input data containing a binary matrix comprising replicate sample pairs

Usage

```
data(BinMatInput_reps)
```

Format

A dataframe with columns for loci, and rows for replicate sample pairs.

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
check.data(mat)
cons = consolidate(mat)
pks = peaks.consolidated(cons)
err = errors(cons)
rem = peak.remove(cons, 4)
clust = upgma(cons)
```

check.data *Checks binary matrix for unwanted characters.*

Description

Checks for unwanted values (other than 1, 0, and ?) in the data set.

Usage

```
check.data(x)
```

Arguments

x	A CSV file containing replicate sample pairs of binary data.

Value

Index positions where unwanted values occur (row, column).

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
check.data(mat)
```

`consolidate` *Consolidates replicate pairs in a binary matrix.*

Description

Reads in a binary matrix comprising replicate pairs and consolidates each pair into a consensus read. For each replicate pair at each locus, 1 & 1 -> 1 (shared presence), 0 & 0 -> 0 (shared absence), 0 & 1 -> ? (ambiguity).

Usage

```
consolidate(x)
```

Arguments

`x`	A CSV file containing replicate pairs of binary data. See the example input file "BinMatInput_reps".

Value

Consolidated binary matrix.

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
cons = consolidate(mat)
```

`errors` *Calculates Jaccard and Euclidean error rates.*

Description

Calculates the Jaccard and Euclidean error rates for the dataset. Jaccard's error does not take shared absences of bands as being biologically meaningful. Jaccard Error = (f10 + f01)/(f10 + f01 + f11) and Euclidean Error = (f10 + f01)/(f10 + f01 + f11 + f00). At each locus, f01 and f10 indicates a case where a 0 was present in one replicate, and a 1 in the other. f11 indicates the shared presence of a band in both replicates, and f00 indicates a shared absence. For example, if a replicate pair comprises Rep1 = 00101 and Rep2 = 01100, Jaccard Error = (1+1)/(1+1+1) = 2/3 = 0.67, Euclidean Error = (1+1)/(1+1+1+2) = 2/5 = 0.4.

Usage

```
errors(x)
```

Arguments

x	Consolidated binary matrix.

Value

JE (Jaccard Error), EE (Euclidean Error), and standard deviations.

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
cons = consolidate(mat)
errors(cons)
```

group.names *Outputs group names specified in the input file for the creation of an nMDS plot.*

Description

Returns group names in the uploaded consolidated binary data. This will help in knowing which colours are assigned to which group name.

Usage

```
group.names(x)
```

Arguments

x	Consolidated binary matrix with grouping information in column 2.

Value

Scree plot.

Examples

```
mat = BinMatInput_ordination
group.names(mat)
```

nmds *Creates a non-metric multidimensional scaling plot (nMDS).*

Description

Creates an non-metric multidimensional scaling plot from a consolidated binary matrix with grouping information. Colours and shapes of plotted points need to be specified. For example, if there are two groups, then: clrs = c("red", "blue"), sh = c(16, 16). This assigns red to the first group name, and blue to the second. Both will have a pch shape of 16 (round dot). These two vectors are then passed to the function nmds() as: colours = clrs, shapes = sh.

Usage

```
nmds(
  x,
  dist_meth = "binary",
  k_val = 2,
  pt_size = 1,
  colours,
  shapes,
  labs = FALSE
)
```

Arguments

x	Consolidated binary matrix with grouping information in the second column.
dist_meth	Distance method. Set to "binary" by default. Other options are "euclidean", "maximum", "manhattan", "canberra", or "minkowski".
k_val	Number of dimensions for the nMDS plot. Set to 2 by default.
pt_size	Point size for symbols on the plot. Set to 1 by default.
colours	Vector containing colours to be assigned to groups.
shapes	Vector containing pch values for shapes to be used for points.
labs	Indicate whether labels should appear on the graph or not (TRUE or FALSE). Default = FALSE.

Value

nMDS plot.

Examples

```
mat = BinMatInput_ordination
group.names(mat)
clrs = c("red", "green", "black")
shp = c(16,16,16)
nmds(mat, colours = clrs, shapes = shp, labs = TRUE)
```

peak.remove — *Removes samples with peaks equal to or less than a specified threshold value.*

Description

Removes samples with a peak number less than a specified value.

Usage

```
peak.remove(x, thresh)
```

Arguments

x	Binary matrix (consolidated or original).
thresh	Peak threshold value for removal.

Value

Filtered dataset, and either the row name/s or row number/s of samples that were removed.

Examples

```
mat = BinMatInput_ordination
new = peak.remove(mat, 4)
```

peaks.consolidated — *Calculates peak numbers for a consolidated data set (total, maximum, and minimum).*

Description

Returns the total, maximum, and minimum number of peaks in the binary matrix.

Usage

```
peaks.consolidated(x)
```

Arguments

x	Consolidated binary matrix (consolidated in BinMat, or a pre-processed matrix from file).

Value

Peak information.

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
cons = consolidate(mat)
peaks.consolidated(cons)
```

---

peaks.original   *Calculates peak numbers for the data set with all replicates (total, maximum, and minimum).*

Description

Returns the total, maximum, and minimum number of peaks in the binary matrix.

Usage

```
peaks.original(x)
```

Arguments

x	Binary matrix comprising replicate pairs.

Value

Peak information.

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
peaks.original(mat)
```

---

scree   *Draws a scree plot.*

Description

Creates a scree plot for the non-metric multidimensional scaling plot. This indicates the optimum number of dimensions to use to minimise the stress value. The stress value is indicated by a red dotted line at 0.15. Values equal to or below this are considered acceptable.

Usage

```
scree(x, dimensions = 4, dist_meth = "binary")
```

Arguments

`x` Consolidated binary matrix with grouping information in column 2.

`dimensions` Number of dimensions to plot. Set to 4 by default.

`dist_meth` Distance method. Set to "binary" by default. Other options are "euclidean", "maximum", "manhattan", "canberra", or "minkowski".

Value

Scree plot.

Examples

```
mat = BinMatInput_ordination
scree(mat)
```

`shepard` *Creates a shepard plot.*

Description

Creates a Shepard plot for the non-metric multidimensional scaling plot. This indicates the ’goodness of fit’ of the original distance matrix vs the ordination representation. A high R-squared value is favourable.

Usage

```
shepard(x, k_val = 2, dist_meth = "binary")
```

Arguments

`x` Consolidated binary matrix.

`k_val` Number of dimensions. Set to 2 by default.

`dist_meth` Distance method. Set to "binary" by default. Other options are "euclidean", "maximum", "manhattan", "canberra", or "minkowski".

Value

Shepard plot.

Examples

```
mat = BinMatInput_ordination
shepard(mat)
```

upgma *Draws a hierarchical clustering tree (UPGMA).*

Description

Creates an unweighted pair group method with arithmetic mean (UPGMA) hierarchical clustering tree, with a specified number of bootstrap repetitions.

Usage

```
upgma(
  x,
  bts = 10,
  size = 0.55,
  lab_size = 0.55,
  method = "binary",
  hclust = "average",
  fromFile = FALSE
)
```

Arguments

`x`	Consolidated binarx matrix.
`bts`	Bootstrap replications. Set to 10 by default.
`size`	Size of plot. Set to 0.55 by default.
`lab_size`	Size of label text. Set to 0.55 by default.
`method`	Distance method. Set to 'binary' (=Jaccard distance) by default.
`hclust`	Clustering method. Set to 'average' (=UPGMA) by default
`fromFile`	Indicates whether the binary data used by the function has been consolidated by BinMat, or whether it comes from the user's own file. Set to FALSE by default (in the assumption that the data has been consolidated by BinMat, and that that object is being passed to the function).

Value

UPGMA tree

Examples

```
data(BinMatInput_reps)
mat = BinMatInput_reps
cons = consolidate(mat)
clust = upgma(cons)
```

Index