

# Package ‘BinMat’

March 6, 2022

**Type** Package

**Title** Processes Binary Data Obtained from Fragment Analysis (Such as AFLPs, ISSRs, and RFLPs)

**Version** 0.1.5

**Description** A molecular genetics tool that processes binary data from fragment analysis. It consolidates replicate sample pairs, outputs summary statistics, and produces hierarchical clustering trees and nMDS plots. This package was developed from the publication available here: <<https://www.sciencedirect.com/science/article/pii/S1049964420306538>>.

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", "clarkevansteenderen")` into the console in R. Two real-world datasets accompany the package: an AFLP dataset of Bunias orientalis samples from Tewes et. al. (2017) <<https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/1365-2745.12869>>, and an ISSR dataset of Nymphaea specimens from Reid et. al. (2021) <<https://www.sciencedirect.com/science/article/pii/S0304377021000218>>.

The authors of these publications are thanked for allowing the use of their data.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Depends** R (>= 3.0)

**Imports** pvclust (>= 2.0), MASS (>= 7.3), stats (>= 3.4.0), graphics (>= 3.4.0), base (>= 3.4.0), ggpubr (>= 0.4.0), tibble (>= 3.1.4)

**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](mailto:vsteenderen@gmail.com)>

**Repository** CRAN

**Date/Publication** 2022-03-06 07:50:14 UTC

## R topics documented:

BinMatInput_ordination . . . . .	2
BinMatInput_reps . . . . .	3
bunias_orientalis . . . . .	3
check.data . . . . .	4
consolidate . . . . .	5
errors . . . . .	5
group.names . . . . .	6
nm.ds . . . . .	7
nymphaea . . . . .	8
peak.remove . . . . .	9
peaks.consolidated . . . . .	9
peaks.original . . . . .	10
scree . . . . .	10
shepard . . . . .	11
upgma . . . . .	12
<b>Index</b>	<b>13</b>

---

BinMatInput\_ordination

*Example input data containing a consolidated binary matrix with groups*

---

### Description

Example input data containing a consolidated binary matrix with groups

### Usage

```
data(BinMatInput_ordination)
```

### Format

A dataframe with columns for loci, and rows of replicate pairs. 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")
nmds(mat, colours = clrs, labs = TRUE)
```

---

BinMatInput_reps	<i>Example input data containing a binary matrix comprising replicate pairs</i>
------------------	---

---

**Description**

Example input data containing a binary matrix comprising replicate pairs

**Usage**

```
data(BinMatInput_reps)
```

**Format**

A dataframe with columns for loci, and rows of replicate 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)
```

---

bunias_orientalis	<i>Example input file of Bunias orientalis AFLP data, taken from Tewes et. al. (2017). This dataset has already been consolidated, and can be used as input for the generation of an nMDS plot. The paper can be found here: &lt;<a href="https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/1365-2745.12869">https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/1365-2745.12869</a>&gt;</i>
-------------------	--

---

**Description**

Example input file of *Bunias orientalis* AFLP data, taken from Tewes et. al. (2017). This dataset has already been consolidated, and can be used as input for the generation of an nMDS plot. The paper can be found here: <<https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/1365-2745.12869>>

**Usage**

```
data(bunias_orientalis)
```

**Format**

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

---

check.data	<i>Checks binary matrix for unwanted characters.</i>
------------	--

---

**Description**

Checks for unwanted values (other than 1, 0, and ?).

**Usage**

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

**Arguments**

x                    A CSV file containing replicate pairs of binary data.

**Value**

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

**Examples**

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

---

consolidate	<i>Consolidates replicate pairs in a binary matrix.</i>
-------------	---

---

**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	<i>Calculates Jaccard and Euclidean error rates.</i>
--------	--

---

**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.  $JE = (f_{10} + f_{01}) / (f_{10} + f_{01} + f_{11})$  and  $EE = (f_{10} + f_{01}) / (f_{10} + f_{01} + f_{11} + f_{00})$ . At each locus,  $f_{01}$  and  $f_{10}$  indicates a case where a 0 was present in one replicate, and a 1 in the other.  $f_{11}$  indicates the shared presence of a band in both replicates, and  $f_{00}$  indicates a shared absence. For example, if a replicate pair comprises Rep1 = 00101 and Rep2 = 01100,  $JE = (1+1)/(1+1+1) = 2/3 = 0.67$ ,  $EE = (1+1)/(1+1+1+2) = 2/5 = 0.4$ .

**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	<i>Outputs group names specified in the input file for the creation of an nMDS plot.</i>
-------------	--

---

**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 nMDS 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 = c("dodgerblue", "black", "red", "green3", "orange", "darkblue", "gold2",
             "darkgreen", "darkred", "grey", "darkgrey", "magenta", "darkorchid", "purple",
             "brown", "coral3", "turquoise", "deeppink", "lawngreen", "deepskyblue", "tomato",
             "yellow", "yellowgreen", "royalblue", "olivedrab", "midnightblue", "indianred1",
             "darkturquoise"),
  labs = FALSE,
  legend_pos = "right",
  include_ellipse = FALSE,
  ellipse_type = "norm",
  dimension1 = 1,
  dimension2 = 2
)
```

### Arguments

<code>x</code>	Consolidated binary matrix with grouping information in the second column.
<code>dist_meth</code>	Distance method. Set to "binary" by default. Other options are "euclidean", "maximum", "manhattan", "canberra", or "minkowski".
<code>k_val</code>	Number of dimensions for the nMDS plot. Set to 2 by default.
<code>pt_size</code>	Point size for symbols on the plot. Set to 1 by default.
<code>colours</code>	Vector containing colours to be assigned to groups. This can be changed to the options available in the RColorBrewer palette set (e.g. "Set1"). See <a href="http://applied-r.com/rcolorbrewer-palettes/">http://applied-r.com/rcolorbrewer-palettes/</a> for more palette options. Alternatively, the colours can be set manually using, for example, <code>c("red", "green", "blue")</code> , thereby setting a colour for each group in your dataset. There are 28 default colours that will be set automatically to your groups.
<code>labs</code>	Indicate whether labels should appear on the graph or not (TRUE or FALSE). Default = FALSE.

legend_pos	Indicate the position of the legend. Default = "right", but other options are "left", "bottom", "top", or "none"
include_ellipse	Indicate whether ellipses should be included around groups. Default = FALSE.
ellipse_type	Select the type of ellipses to include around groups. Options are "convex", "confidence", "t", "norm", and "euclid". See the <code>ggpubr::ggscatter()</code> function documentation for more details.
dimension1	Indicate the first dimension to plot (1, 2, or 3) for the x axis. If <code>k = 2</code> , the first two dimensions will automatically be plotted. If <code>k = 3</code> , select between the three.
dimension2	Indicate the second dimension to plot (1, 2, or 3) for the y axis

**Value**

nMDS plot.

**Examples**

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

---

nymphaea	<i>Example input file of Nymphaea ISSR data, taken from Reid et. al. (2021). This dataset has already been consolidated, and can be used as input for the generation of an nMDS plot. The paper can be found here: &lt;<a href="https://www.sciencedirect.com/science/article/pii/S0304377021000218">https://www.sciencedirect.com/science/article/pii/S0304377021000218</a>&gt;</i>
----------	--

---

**Description**

Example input file of Nymphaea ISSR data, taken from Reid et. al. (2021). This dataset has already been consolidated, and can be used as input for the generation of an nMDS plot. The paper can be found here: <<https://www.sciencedirect.com/science/article/pii/S0304377021000218>>

**Usage**

```
data(nymphaea)
```

**Format**

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



---

peak.remove	<i>Removes samples with peaks equal to or less than a specified threshold value.</i>
-------------	--

---

**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	<i>Calculates peak numbers for a consolidated data set (total, maximum, and minimum).</i>
--------------------	---

---

**Description**

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

**Usage**

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

**Arguments**

x	Binary matrix comprising replicate pairs.
---	---

**Value**

Peak information.

**Examples**

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

---

peaks.original	<i>Calculates peak numbers for the data set with all replicates (total, maximum, and minimum).</i>
----------------	--

---

**Description**

Returns 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	<i>Draws a scree plot.</i>
-------	----------------------------

---

**Description**

Creates a scree plot for the nMDS. 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	<i>Creates a shepard plot.</i>
---------	--------------------------------

---

**Description**

Creates a Shepard plot for the nMDS. 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 a 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

<code>x</code>	Consolidated binarx matrix.
<code>bts</code>	Bootstrap replications. Set to 10 by default.
<code>size</code>	Size of plot. Set to 0.55 by default.
<code>lab_size</code>	Size of label text. Set to 0.55 by default.
<code>method</code>	Distance method. Set to 'binary' (=Jaccard distance) by default.
<code>hclust</code>	Clustering method. Set to 'average' (=UPGMA) by default
<code>fromFile</code>	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

## \* datasets

BinMatInput\_ordination, [2](#)

BinMatInput\_reps, [3](#)

bunias\_orientalis, [3](#)

nymphaea, [8](#)

BinMatInput\_ordination, [2](#)

BinMatInput\_reps, [3](#)

bunias\_orientalis, [3](#)

check.data, [4](#)

consolidate, [5](#)

errors, [5](#)

group.names, [6](#)

nmds, [7](#)

nymphaea, [8](#)

peak.remove, [9](#)

peaks.consolidated, [9](#)

peaks.original, [10](#)

scree, [10](#)

shepard, [11](#)

upgma, [12](#)