Package ‘backbone’

December 2, 2019

Type Package
Title Extracts the Backbone from Weighted Graphs
Version 1.1.0
Description Provides methods for extracting from a weighted graph
a binary or signed backbone that retains only the significant edges.
The user may input a weighted graph, or a bipartite graph
from which a weighted graph is first constructed via projection.
Backbone extraction methods include the stochastic degree se-
011-0021-0>),
as well as a universal threshold method.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Depends R (>= 2.10)
Imports Matrix, methods, poibin, stats, utils
Suggests igraph, knitr, rmarkdown, speedglm, testthat
VignetteBuilder knitr
URL https://github.com/domagal9/backbone,
https://www.zacharyneal.com/backbone
BugReports https://github.com/domagal9/backbone/issues
NeedsCompilation no
Author Rachel Domagalski [aut, cre],
Zachary Neal [aut],
Bruce Sagan [aut]
Maintainer Rachel Domagalski <domagal9@msu.edu>
Repository CRAN
Date/Publication 2019-12-02 19:30:09 UTC
**R topics documented:**

- backbone .................................................. 2
- backbone.extract ........................................ 3
- davis .......................................................... 3
- fdsm ........................................................... 4
- hyperg ......................................................... 5
- sdsm ........................................................... 6
- universal ...................................................... 7

**Index** 9

---

**backbone**

**backbone: Extracts the Backbone from Weighted Graphs**

**Description**


**Details**

Some features of the package are:

- `'universal'`: returns a unipartite backbone matrix in which values are set to 1 if above the given upper parameter threshold, and set to -1 if below the given lower parameter threshold, and are 0 otherwise.
- `'sdsm'`: computes the proportion of generated edges above or below the observed value using the stochastic degree sequence model. Once computed, use `backbone.extract` to return the backbone matrix for a given alpha value.
- `'fdsm'`: computes the proportion of generated edges above or below the observed value using the fixed degree sequence model. Once computed, use `backbone.extract` to return the backbone matrix for a given alpha value.
- `'hyperg'`: returns a binary or signed adjacency matrix containing the backbone that retains only the significant edges.
- `'backbone.extract'`: returns a positive or signed adjacency matrix containing the backbone: only the significant edges.

For additional documentation and background on the package functions, see `browseVignettes("backbone")`. 
**Description**

‘backbone.extract’ returns a binary or signed adjacency matrix containing the backbone that retains only the significant edges.

**Usage**

```r
backbone.extract(positive, negative = NULL, alpha = 0.05)
```

**Arguments**

- **positive**: Matrix: proportion of times the projected matrix values were above the corresponding matrices generated by `hyperg`, `fdsm`, or `sdsm`.
- **negative**: Matrix: proportion of times the projected matrix values were below the corresponding matrices generated by `hyperg`, `fdsm`, or `sdsm`. If supplied, a signed backbone will be returned; if not supplied, a binary backbone will be returned. Default is set to NULL.
- **alpha**: Real: Precision of significance test (one-tailed if only the positive matrix supplied, two-tailed if positive and negative matrices supplied)

**Value**

backbone Matrix: Binary or signed adjacency matrix of backbone graph.

**Examples**

```r
probs <- sdsm(davis, 100)
bb <- backbone.extract(probs$positive, probs$negative)
```

**davis**

*Davis Southern Women Data Set*

**Description**

A two mode matrix of 18 women and attendance of 14 social events.

**Usage**

```r
data(davis)
```

**Format**

An object of class `matrix` with 18 rows and 14 columns.
The fixed degree sequence model (fdsm)

Description

'fdsm' computes the proportion of generated edges above or below the observed value using the fixed degree sequence model. Once computed, use `backbone.extract` to return the backbone matrix for a given alpha value.

Usage

```r
fdsm(B, trials = 1000, sparse = TRUE, dyad = NULL,
     progress = FALSE)
```

Arguments

- `B` Matrix: Bipartite adjacency matrix
- `trials` Integer: Number of random bipartite graphs generated
- `sparse` Boolean: If sparse matrix manipulations should be used
- `dyad` vector length 2: two row entries i,j. Saves each value of the i-th row and j-th column in each projected B* matrix. This is useful for visualizing an example of the empirical null edge weight distribution generated by the model. These correspond to the row and column indices of a cell in the projected matrix, and can be written as their string row names or as numeric values.
- `progress` Boolean: If `txtProgressBar` should be used to measure progress

Details

During each iteration, `fdsm` computes a new B* matrix using the Curveball algorithm. This is a random bipartite matrix with the same row and column sums as the original matrix B. If the dyad parameter is indicated to be used in the parameters, when the B* matrix is projected, the projected value for the corresponding row and column will be saved. This allows the user to see the distribution of the edge weights for desired row and column.
`hyperg` 5

**Value**

list(positive, negative, dyad_values, summary). positive: matrix of proportion of times each entry of the projected matrix `B` is above the corresponding entry in the generated projection. negative: matrix of proportion of times each entry of the projected matrix `B` is below the corresponding entry in the generated projection. dyad_values: list of edge weight for i,j in each generated projection. summary: a data frame summary of the inputted matrix and the model used including: model name, number of rows, skew of row sums, number of columns, skew of column sums, and running time.

**References**


**Examples**

```r
fdsm_props <- fdsm(davis, trials = 100, sparse = TRUE, dyad=c(3,6))
```

---

**Description**

`hyperg` computes the probability of observing a higher or lower edge weight using the hypergeometric distribution. Once computed, use `backbone.extract` to return the backbone matrix for a given alpha value.

**Usage**

`hyperg(B)`

**Arguments**

`B` Matrix: Bipartite network

**Value**

list(positive, negative, summary). positive gives matrix of probability of ties above the observed value. negative gives matrix of probability of ties below the observed value. summary: a data frame summary of the inputted matrix and the model used including: model name, number of rows, skew of row sums, number of columns, skew of column sums, and running time.
References


Examples

```r
hypergeometric_bb <- hyperg(davis)
```

---

**The stochastic degree sequence model (sdsm)**

**Description**

‘sdsm’ computes the proportion of generated edges above or below the observed value using the stochastic degree sequence model. Once computed, use `backbone.extract` to return the backbone matrix for a given alpha value.

**Usage**

```r
sdsm(B, trials = 0, model = "logit", sparse = TRUE, maxiter = 25, dyad = NULL, alpha = 0.05, tolerance = 0, progress = FALSE)
```

**Arguments**

- **B**: Matrix: Bipartite adjacency matrix
- **trials**: Integer: Number of random bipartite graphs generated. Default is 0.
- **model**: String: A generalized linear model (glm) used to generate random bipartite graphs.
- **sparse**: Boolean: If sparse matrix manipulations should be used
- **maxiter**: Integer: Maximum number of iterations if "model" is a glm.
- **dyad**: vector length 2: two row entries i,j. Saves each value of the i-th row and j-th column in each projected B* matrix. This is useful for visualizing an example of the empirical null edge weight distribution generated by the model. These correspond to the row and column indices of a cell in the projected matrix, and can be written as their string row names or as numeric values.
- **alpha**: Real: proposed alpha threshold to be used for determining statistical significance of edges
- **tolerance**: Real: tolerance for p-value computation using RNA poisson-binomial approximation
- **progress**: Boolean: If `txtProgressBar` should be used to measure progress
universal

Details

The 'model' parameter can take in a 'link' function, as described by \texttt{glm} and \texttt{family}. This can be one of \texttt{c('logit', 'probit', 'cauchit', 'log', 'cloglog')}.

If 'trials'>0, the function uses repeat Bernoulli trials to compute the proportions, using the following steps: During each iteration, \texttt{sdsm} computes a new \(B^*\) matrix using probabilities computed using the 'glm'. This is a random bipartite matrix with about the same row and column sums as the original matrix \(B\). If the dyad\_parameter is indicated to be used in the parameters, when the \(B^*\) matrix is projected, the projected value for the corresponding row and column will be saved. This allows the user to see the distribution of the edge weights for desired row and column.

If 'trials'=0, the proportion of edges above or below the observed values are computed using the Poisson Binomial distribution. These values are approximated using either a Discrete Fourier Transform (DFT method) or a Refined Normal Approximation (RNA method). These functions are described by \texttt{ppoibin}. The RNA method is used by default, unless the computed value is within the margin of 'alpha'-tolerance' and 'alpha'+tolerance', the DFT method is used.

Value

list(positive, negative, dyad\_values, summary). positive: matrix of proportion of times each entry of the projected matrix \(B\) is above the corresponding entry in the generated projection. negative: matrix of proportion of times each entry of the projected matrix \(B\) is below the corresponding entry in the generated projection. dyad\_values: list of edge weight for i,j in each generated projection, included if 'dyad' not NULL and 'trials > 0'. summary: a data frame summary of the inputted matrix and the model used including: model name, number of rows, skew of row sums, number of columns, skew of column sums, and running time.

References


Examples

\begin{verbatim}
sdsm_bt <- sdsm(davis, trials = 100,dyad = c("EVELYN", "CHARLOTTE" ))
sdsm_rna <- sdsm(davis, trials = 0, tolerance = 0)
sdsm_dft <- sdsm(davis, trials = 0, tolerance = 1)
\end{verbatim}

universal | \textit{Compute universal threshold backbone}

Description

'\emph{universal}' returns a unipartite backbone matrix in which values are set to 1 if above the given upper parameter threshold, and set to -1 if below the given lower parameter threshold, and are 0 otherwise.
Usage
universal(M, upper = 0, lower = NULL, bipartite = FALSE)

Arguments
M Matrix: a weighted adjacency matrix or a bipartite adjacency matrix.
upper Real or FUN: upper threshold value or function to be applied to the edge weights. Default is 0.
lower Real or FUN: lower threshold value or function to be applied to the edge weights. Default is NULL.
bipartite Boolean: TRUE if bipartite matrix, FALSE if weighted matrix. Default is FALSE.

Value
list(backbone, summary). backbone: a matrix, Signed (or positive) adjacency matrix of backbone summary: a data frame summary of the inputted matrix and the model used including: model name, number of rows, skew of row sums, number of columns, skew of column sums, and running time.

Examples
test <- universal(davis%*%t(davis), upper = function(x)mean(x)+sd(x), lower=function(x)mean(x))
test2 <- universal(davis, upper = function(x)mean(x)+2*sd(x), lower = 2, bipartite = TRUE)
test3 <- universal(davis, upper = 4, lower = 2, bipartite = TRUE)
## Index

*Topic **datasets**
  - davis, 3
  - backbone, 2
  - backbone-package (backbone), 2
  - backbone.extract, 2, 3, 4–6
  - davis, 3
  - family, 7
  - fdsm, 2, 3, 4
  - glm, 7
  - hyperg, 2, 3, 5
  - ppoibin, 7
  - sdsm, 2, 3, 6
  - txtProgressBar, 4, 6
  - universal, 2, 7