

# Package ‘signed.backbones’

June 1, 2021

**Title** Extract the Signed Backbones of Weighted Networks

**Version** 0.91.5

**Description** Extract the signed backbones of intrinsically dense weighted networks based on the significance filter and vigor filter as described in the following paper. Please cite it if you find this software useful in your work.  
Furkan Gursoy and Bertan Badur. "Extracting the signed backbone of intrinsically dense weighted networks." Journal of Complex Networks. <[arXiv:2012.05216](https://arxiv.org/abs/2012.05216)>.

**Encoding** UTF-8

**License** GPL-3

**Imports** reshape2

**NeedsCompilation** no

**Author** Furkan Gursoy [aut, cre] (<<https://orcid.org/0000-0001-9701-2814>>)

**Maintainer** Furkan Gursoy <[furkan.gursoy@boun.edu.tr](mailto:furkan.gursoy@boun.edu.tr)>

**Repository** CRAN

**Date/Publication** 2021-06-01 06:50:02 UTC

## R topics documented:

extract . . . . .	1
<b>Index</b>	<b>4</b>

---

extract	<i>Extract the signed backbones of weighted networks</i>
---------	--

---

## Description

Signed backbones are extracted based on the significance filter and vigor filter as described in the following paper. Please cite it if you find this software useful in your work.

Furkan Gursoy and Bertan Badur. "Extracting the signed backbone of intrinsically dense weighted networks." Journal of Complex Networks.

**Usage**

```
extract(
  edgelist,
  directed = TRUE,
  significance_threshold = "15pc",
  vigor_threshold = 0.1,
  return_weights = FALSE,
  return_significance = FALSE,
  max_iteration = 100,
  precision = 1e-07
)
```

**Arguments**

- edgelist** A data frame. First two columns contain node pairs, and the third column contains the edge weights. If `directed = TRUE`, columns should be in this order: source node, target node, edge weight.
- directed** Whether the input network is directed. Defaults to `TRUE`.
- significance\_threshold** Threshold for the significance filter. Defaults to `'15pc'`.  
 (1) If filtering is directly based on alpha values: (1a) If scalar, a single nonnegative value, e.g., 1.23. (1b) If vector, a vector of nonpositive and nonnegative values, e.g., `c(-1.23, 4.56)`. 1.23 is equivalent to `c(-1.23, 1.23)`.  
 (2) If filtering is based on ranking: (2a) If string, a single percentage value in the following format: `'10pc'`. (2b) If vector, a vector of percentage values in the following format: `c('5pc', '5pc')`. `'10pc'` is not equivalent to `c('10pc', '10pc')` since the latter retains 20% of possible links. `'10pc'` is not equivalent `c('5pc', '5pc')` since the latter retains 5% of edges on negative extreme and 5% of edges on positive extreme whereas the former simultaneously considers both extremes.
- vigor\_threshold** Threshold for the vigor filter. Defaults to 0.1.  
 (1) If scalar, a single nonnegative value in the range `[0, 1]`, e.g., 0.33. (2) If vector, a vector of nonpositive and nonnegative values in the ranges `[-1, 0]` and `[0, 1]`, e.g., `c(-0.5, 0.3)`. 0.33 is equivalent to `c(-0.33, 0.33)`.
- return\_weights** Whether the returned backbone should contain the signed link weights that show the intensity of signed links. Defaults to `FALSE`.
- return\_significance** Whether the returned backbone should contain the link significance values that are benchmarked against the `significance_threshold`. Defaults to `FALSE`.
- max\_iteration** Maximum number of iterations to be used in the Iterational Proportional Fitting Procedure. Defaults to 100.
- precision** A small epsilon value to be used in comparison with zero values due to numerical precision issues. Can be left as default. Defaults to `10e-8`.

**Value**

A data frame with shape (n\_backbone\_edges, 3 or 4) containing the edges for the extracted backbone. First two columns contain node pairs, and the third column contains the edge sign. If directed = TRUE, columns are in this order: source node, target node, edge sign. If return\_weights = TRUE, signed edge weights are returned instead of edge sign. If return\_significance = TRUE, a fourth column containing significance values is also returned.

**See Also**

More examples may be found at the project's homepage.

**Examples**

```
net <- data.frame(source = c('a', 'a', 'b', 'b', 'c', 'c'),
  target = c('b', 'c', 'a', 'c', 'b', 'd'),
  weight = c(1, 20, 2, 30, 10, 1))
backbone <- extract(net, directed= TRUE,
  significance_threshold = '20pc',
  vigor_threshold = c(-0.3, 0.2))
```

# Index

- \* **backbone extraction**  
extract, 1
- \* **complex networks**  
extract, 1
- \* **dense networks**  
extract, 1
- \* **information filtering**  
extract, 1
- \* **signed backbones**  
extract, 1
- \* **signed networks**  
extract, 1
- \* **weighted networks**  
extract, 1

extract, 1