# Package 'fabisearch'

March 15, 2022

**Title** Change Point Detection in High-Dimensional Time Series Networks

**Version** 0.0.4.4

**Description** Implementation of the Factorized Binary Search (FaBiSearch) methodology for the estimation of the number and the location of multiple change points in the network (or clustering) structure of multivariate high-dimensional time series. The method is motivated by the detection of change points in functional connectivity networks for functional magnetic resonance imaging (fMRI) data. FaBiSearch uses non-negative matrix factorization (NMF), an unsupervised dimension reduction technique, and a new binary search algorithm to identify multiple change points. It requires minimal assumptions. Lastly, we provide interactive, 3-dimensional, brain-specific network visualization capability in a flexible, stand-alone function. This function can be conveniently used with any node coordinate atlas, and nodes can be color coded according to community membership, if applicable. The output is an elegantly displayed network laid over a cortical surface, which can be rotated in the 3-dimensional space. The main routines of the package are detect.cps(), for multiple change point detection, est.net(), for estimating a network between stationary multivariate time series, net.3dplot(), for plotting the estimated functional connectivity networks, and opt.rank(), for finding the optimal rank in NMF for a given data set. The functions have been extensively tested on simulated multivariate high-dimensional time series data and fMRI data. For details on the FaBiSearch methodology, please see Ondrus et al. (2021) <arXiv:2103.06347>. For a more detailed explanation and applied examples of the fabisearch package, please see Ondrus and Cribben (2022), preprint.

**URL** https://github.com/mondrus96/FaBiSearch

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** rgl, reshape2, foreach, doParallel, doRNG, parallel, pkgmaker

**Depends** R (>= 3.10), NMF

**Author** Martin Ondrus [aut, cre],
Ivor Cribben [aut]

**Maintainer** Martin Ondrus <mondrus@ualberta.ca>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-03-15 22:20:02 UTC

# R topics documented:



---

`AALatlas` *Automated Anatomical Labeling (AAL) atlas coordinates*

---

## Description

A dataframe of the Automated Anatomical Labeling (AAL) atlas from the work of Tzourio-Mazoyer et al. (2002) atlas to use with the `net.3dplot()` function. Each row corresponds to a region of interest (ROI) to be plotted using the Montreal Neurological Institute (MNI) space. The first column corresponds to the community labels (in this atlas, there are none, therefore this column is filled with NA), and the second, third, and fourth columns correspond to the X, Y, and Z coordinates of the ROIs in MNI space, respectively. See Tzourio-Mazoyer et al. (2002) <doi:10.1006/nimg.2001.0978> for more details.

## Usage

```
AALatlas
```

## Format

A dataframe with 90 rows and 4 columns/variables.

## Source

doi: 10.1006/nimg.2001.0978

## AALfmri — *90 ROI data from the NYU test-retest resting state fMRI data set*

### Description

A data matrix of the second scan from the first subject from the NYU test-retest resting-state fMRI data set. Variables/nodes are defined using the Tzourio-Mazoyer et al. (2002) Automatic Anatomical Labeling (AAL) atlas, which can be accessed as the `AALatlas` dataframe.

### Usage

```
AALfmri
```

### Format

A data matrix with 197 rows and 90 columns/variables, where each column corresponds to an ROI from the AAL atlas.

### Source

https://www.nitrc.org/projects/nyu_trt/

## adjmatrix — *Adjacency matrix for the NYU test-restest resting-state fMRI data set*

### Description

The adjacency matrix calculated from the `gordfmri` data set, using the Gordon atlas.

### Usage

```
adjmatrix
```

### Format

A 333 * 333 matrix, where each entry takes a value 1 (0) if two nodes are (not) connected by an edge, using the Gordon atlas.

### Source

https://www.nitrc.org/projects/nyu_trt/

`detect.cps` *Multiple change point detection in the network (or clustering) structure of multivariate high-dimensional time series*

## Description

This function detects multiple change points in the network (or clustering) structure of multivariate high-dimensional time series using non-negative matrix factorization and a binary search.

## Usage

```
detect.cps(
  Y,
  mindist = 35,
  nruns = 50,
  nreps = 100,
  alpha = NULL,
  rank = NULL,
  algtype = "brunet",
  testtype = "t-test"
)
```

## Arguments

| | |
|---|---|
| `Y` | An input multivariate time series in matrix format, with variables organized in columns and time points in rows. All entries in Y must be positive. |
| `mindist` | A positive integer with default value equal to 35. It is used to define the minimum distance acceptable between detected change points. |
| `nruns` | A positive integer with default value equal to 50. It is used to define the number of runs in the NMF function. |
| `nreps` | A positive integer with default value equal to 100. It is used to define the number of permutations for the statistical inference procedure. |
| `alpha` | A positive real number with default value set to NULL. When alpha = NULL, then the p-value calculated for inference on the change points is returned. If alpha = a positive integer value, say 0.05, then it is used to define the significance level for inference on the change points. |
| `rank` | A positive integer, which defines the rank used in the optimization procedure to detect the change points. If rank = NULL, which is also the default value, then the optimal rank is computed. If rank = a positive integer value, say 4, then a predetermined rank is used. |
| `algtype` | A character string, which defines the algorithm to be used in the NMF function. By default it is set to "brunet". See the "Algorithms" section of `nmf` for more information on the available algorithms. |

`testtype` A character string, which defines the type of statistical test to use during the inference procedure. By default it is set to "t-test". The other options are "ks" and "wilcox" which correspond to the Kolmogorov-Smirnov and Wilcoxon tests, respectively.

## Value

A list with the following components :
`rank`: The rank used in the optimization procedure for change point detection.
`change_points`: A table of the detected change points where column "T" is the time of the change point and "stat_test" is the result (either a boolean value if alpha = a positive real number, or the p-value if alpha = NULL) of the t-test.
`compute_time`: The computational time, saved as a "difftime" object.

## Author(s)

Martin Ondrus, <mondrus@ualberta.ca>, Ivor Cribben, <cribben@ualberta.ca>

## References

"Factorized Binary Search: a novel technique for change point detection in multivariate high-dimensional time series networks", Ondrus et al. (2021), <arXiv:2103.06347>.

## Examples

```
## Change point detection for a multivariate data set, sim2, using settings:
## rank = 3, mindist = 99, nruns = 2, and nreps = 2
set.seed(123)
detect.cps(sim2, rank = 3, mindist = 99, nruns = 2, nreps = 2)


# $rank
# [1] 3
#
# $change_points
#     T stat_test
# 1 101 0.3867274
#
# $compute_time
# Time difference of 0.741534 mins
```

---

`est.net` *Sparse network estimation using non-negative matrix factorization (NMF) for data between change points*

---

## Description

This function estimates sparse networks using non-negative matrix factorization (NMF) for data between change points.

## Usage

```
est.net(
  Y,
  lambda,
  nruns = 50,
  rank = "optimal",
  algtype = "brunet",
  changepoints = NULL
)
```

## Arguments

| | |
|---|---|
| Y | An input multivariate time series in matrix format, with variables organized in columns and time points in rows. All entries in Y must be positive. |
| lambda | A positive real number, which defines the clustering method and/or the cutoff value when estimating an adjacency matrix from the computed consensus matrix. If lambda = a positive integer value, say 6, complete-linkage, hierarchical clustering is applied to the consensus matrix and the cutoff is at 6 clusters. If lambda is a vector of positive integer values, say c(4, 5, 6), the same clustering method is applied for each value sequentially. If lambda = a positive real number, say 0.5, entries in the consensus matrix with a value greater than or equal to 0.5 are labeled 1, while entries less than 0.5 are labeled 0. Similarly, if lambda is a vector of positive real numbers, say c(0.1, 0.3, 0.8), the same thresholding method is applied for each value sequentially. |
| nruns | A positive integer with default value equal to 50. It is used to define the number of runs in the NMF function. |
| rank | A character string or a positive integer, which defines the rank used in the optimization procedure to detect the change points. If rank = "optimal", which is also the default value, then the optimal rank is used. If rank = a positive integer value, say 4, then a predetermined rank is used. |
| algtype | A character string, which defines the algorithm to be used in the NMF function. By default it is set to "brunet". See the "Algorithms" section of nmf for more information on the available algorithms. |
| changepoints | A vector of positive integers with default value equal to NULL. It is used to specify whether change points exist in the input $Y$, and thus whether $Y$ should be split into multiple stationary segments and networks estimated separately for each segment. If change points, say c(100, 200) are specified, $Y$ is split at the 100th and 200th row to correspond to 3 stationary segments. Each stationary segment is then estimated sequentially, and a list is returned where each component corresponds to a stationary segment. |

## Value

A matrix (or more specifically, an adjacency matrix) denoting the network (or clustering) structure between components of $Y$. If lambda is a vector, a list of adjacency matrices is returned, where each element of the list corresponds to an element in lambda.

## Author(s)

Martin Ondrus, <mondrus@ualberta.ca>, Ivor Cribben, <cribben@ualberta.ca>

## References

"Factorized Binary Search: a novel technique for change point detection in multivariate high-dimensional time series networks", Ondrus et al. (2021), <arXiv:2103.06347>.

## Examples

```
## Estimating the network for a multivariate data set, "sim2" with the settings:
## nruns = 10 and lambda = 0.5 where the latter specifies the cutoff based method
est.net(sim2, lambda = 0.5, nruns = 4)
```

---

`fabisearch` *Change Point Detection in High-Dimensional Time Series Networks*

---

## Description

Implementation of the Factorized Binary Search (FaBiSearch) methodology for the estimation of the number and the location of multiple change points in the network (or clustering) structure of multivariate high-dimensional time series. The method is motivated by the detection of change points in functional connectivity networks for functional magnetic resonance imaging (fMRI) data. FaBiSearch uses non-negative matrix factorization (NMF), an unsupervised dimension reduction technique, and a new binary search algorithm to identify multiple change points. It requires minimal assumptions. Lastly, we provide interactive, 3-dimensional, brain-specific network visualization capability in a flexible, stand-alone function. This function can be conveniently used with any node coordinate atlas, and nodes can be color coded according to community membership, if applicable. The output is an elegantly displayed network laid over a cortical surface, which can berotated in the 3-dimensional space. The main routines of the package are `detect.cps()`, for multiple change point detection, `est.net()`, for estimating a network between stationary multivariate time series, `net.3dplot()`, for plotting the estimated functional connectivity networks, and `opt.rank()`, for finding the optimal rank in NMF for a given data set. The functions have been extensively tested on simulated multivariate high-dimensional time series data and fMRI data. For details on the FaBiSearch methodology, please see Ondrus et al. (2021) <arXiv:2103.06347>. For a more detailed explanation and applied examples of the `fabisearch` package, please see Ondrus and Cribben (2021), preprint.

## Value

No return value, called for side effects

## See Also

`detect.cps`, `est.net`, `net.3dplot`, `opt.rank`

## Author(s)

Martin Ondrus, <mondrus@ualberta.ca>, Ivor Cribben, <cribben@ualberta.ca>

## References

"Factorized Binary Search: a novel technique for change point detection in multivariate high-dimensional time series networks", Ondrus et al (2021), <arXiv:2103.06347>.

---

`gordatlas` *Gordon atlas coordinates*

---

## Description

A dataframe of the Gordon et al. (2016) atlas to use with the `net.3dplot()` function. Each row corresponds to a region of interest (ROI) to be plotted using the Montreal Neurological Institute (MNI) space. The first column corresponds to the community labels as a string, and the second, third, and fourth columns correspond to the X, Y, and Z coordinates of the ROIs in MNI space, respectively. See Gordon et al. (2016) <doi:10.1093/cercor/bhu239> for more details.

## Usage

```
gordatlas
```

## Format

A dataframe with 333 rows and 4 columns/variables.

## Source

doi: 10.1093/cercor/bhu239

## gordfmri *333 ROI data from the NYU test-retest resting state fMRI data set*

### Description

A data matrix of the second scan from the first subject from the NYU test-retest resting-state fMRI data set. Variables/nodes are defined using the Gordon et al. (2016) atlas , which can be accessed as the `gordatlas` dataframe.

### Usage

```
gordfmri
```

### Format

A data matrix with 197 rows and 333 columns/variables, where each column corresponds to an ROI from the Gordon atlas.

### Source

https://www.nitrc.org/projects/nyu_trt/

## logSP500 *Daily adjusted logarithmic returns for the Standard and Poor's 500*

### Description

A dataframe of the daily adjusted logarithmic returns for the Standard and Poor's 500 (S&P 500) stock market index. Each row corresponds to a trading day from 2018-01-01 to 2021-03-31. Data was retrieved from Yahoo Finance using the `getSymbols()` function from the `quantmod` package.

### Usage

```
logSP500
```

### Format

A dataframe with 815 rows and 500 columns/variables.

net.3dplot *3D network plot of an adjacency matrix between pairs of change points*

## Description

This function takes an adjacency matrix of a brain network and returns a 3D plot of it.

## Usage

```
net.3dplot(A, ROIs = NULL, colors = NULL, coordROIs = NULL, labels = FALSE)
```

## Arguments

| | |
|---|---|
| A | An adjacency matrix to be plotted (in numerical matrix format). |
| ROIs | Either a vector of character strings specifying the communities to plot, or a vector of integers specifying which ROIs to plot by their ID. By default it is set to NULL, and all communities and ROIs are plotted. Communities available for the Gordon atlas are: "Default", "SMhand", "SMmouth", "Visual", "FrontoParietal", "Auditory", "None", "CinguloParietal", "RetrosplenialTemporal", "CinguloOperc", "VentralAttn", "Salience", and "DorsalAttn". |
| colors | A vector of character strings specifying the hex codes for node colors to distinguish each community. By default, each community is given a predefined, unique color. |
| coordROIs | A dataframe of community tags and Montreal Neurological Institute (MNI) coordinates for regions of interest (ROIs) to plot, which is by default set to NULL and uses the Gordon atlas. See ?gordon.atlas for an example using the Gordon atlas. Format of the dataframe is as follows: first column is a string of community labels, then the subsequent three columns are the x, y, and z coordinates, respectively. See AALatlas and gordatlas for examples. |
| labels | A boolean value denoting whether to add labels to nodes; if set to TRUE, this will add node labels to the plot, and if set to FALSE, will not. By default this is set to FALSE. |

## Value

A 3D network plot of an adjacency matrix between pairs of change points, or for data without change points.

## Author(s)

Martin Ondrus, <mondrus@ualberta.ca>, Ivor Cribben, <cribben@ualberta.ca>

## References

"Factorized Binary Search: a novel technique for change point detection in multivariate high-dimensional time series networks", Ondrus et al. (2021), <arXiv:2103.06347>.

## Examples

```
## Plotting a 333 * 333 adjacency matrix "adjmatrix" with red, blue, and green
## nodes to denote the "Default", "SMhand", and "Visual" communities
comms = c("Default", "SMhand", "Visual")
colrs = c("#FF0000", "#00FF00", "#0000FF")
net.3dplot(adjmatrix, ROIs = comms, colors = colrs)
```

---

`opt.rank` *Finds the optimal rank for non-negative matrix factorization (NMF)*

---

## Description

This function finds the optimal rank for non-negative matrix factorization (NMF).

## Usage

```
opt.rank(Y, nruns = 50, algtype = "brunet")
```

## Arguments

| | |
|---|---|
| `Y` | An input multivariate time series in matrix format, with variables organized in columns and time points in rows. All entries in Y must be positive. |
| `nruns` | A positive integer with default value equal to 50. It is used to define the number of runs in the NMF function. |
| `algtype` | A character string, which defines the algorithm to be used in the NMF function. By default it is set to "brunet". See the "Algorithms" section of `nmf` for more information on the available algorithms. |

## Value

A positive integer representing the optimal rank.

## Author(s)

Martin Ondrus, <mondrus@ualberta.ca>, Ivor Cribben, <cribben@ualberta.ca>

## References

"Factorized Binary Search: a novel technique for change point detection in multivariate high-dimensional time series networks", Ondrus et al. (2021), <arXiv:2103.06347>.

## Examples

```
## Finding the optimal rank for an input data set "sim2" with nruns = 4
set.seed(123)
opt.rank(sim2, nruns = 4)
# [1] "Finding optimal rank"
# [1] "Optimal rank: 2"
# [1] 2
```

---

`sim2` *A simulated data set (see simulation 2 from Ondrus et al., 2021)*

---

## Description

A simulated data set (see simulation 2 from Ondrus et al., 2021). The data is generated from a multivariate Gaussian distribution with 2 clusters, where the correlation between nodes in the same cluster is 0.75 and between different clusters is 0.2 for the first 100 time points. The vertex labels are randomly reshuflled for the second 100 time points. Hence, there is one change point at t=100.

## Usage

```
sim2
```

## Format

A matrix with 200 rows and 80 columns/variables.

# Index