Package 'graphscan'

October 14, 2016

Type Package

Title Cluster Detection with Hypothesis Free Scan Statistic

Version 1.1.1

Date 2016-10-11

Author Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer

Maintainer David Abrial <graphscan@clermont.inra.fr>

Description Multiple scan statistic with variable window for one dimension data and scan statistic based on connected components in 2D or 3D.

Encoding UTF-8

License GPL-2 | GPL-3

Depends R (>= 3.0.2), snowfall (>= 1.84-6)

Imports ape, sp, methods, rgl, utils, graphics, grDevices, stats

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-10-14 16:22:50

R topics documented:

graphscan-package *Cluster detection with hypothesis free scan statistic*

Description

Multiple scan statistic with variable window for one dimension data and scan statistic based on connected components in 2D or 3D.

Details

Package:	graphscan
Type:	Package
Version:	1.1.1
Date:	2016-10-11
License:	GPL-2 | GPL-3
Depends:	R (>= 3.0.2), snowfall (>= 1.84-6)
Imports: ape, sp, methods, rgl, utils, graphics, grDevices, stats	
NeedsCompilation:	yes

Index:

This package implements a statistical method for detecting clusters in dimensions 1, 2 and 3 proposed by Cucala (2008,2009). In 1D, this hypothesis multiple scan statistic with variable window in 1D can detect positive clusters only, negative clusters only, or simultaneously positive or negative clusters. Positive clusters correspond to a particularly high concentration in events, while negative clusters correspond to a particularly low concentration in events. The concentration index of Cucala is based on the properties of the distances between order statistics under the hypothesis of uniform distribution of the events. The 1D functions are adapted to study the repartition of mutations along the genome and detect zones with numerous or few mutations when comparing two aligned DNA sequences. When studying a population, i.e. several DNA sequences, pair comparisons can be created and analysed. In nD (2D or 3D), the flexible spatial scan test for case event data is implemented

to detect only the most significant positive cluster. The candidate clusters are build from connected components. The number of candidate clusters is equal to the number of cases minus one. The concentration index of Cucala is based on the properties of the distances between order statistics under the hypothesis of Poisson distribution of the cases. The analysis for nD data returns both the Cucala and the Kulldorff concentration index.

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Kulldorff, M. 1997. A spatial scan statistic, Commun. Statist. - Theory Meth., 26, 6, p. 1481-1496.

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

---

barplot *Barplot of the 1D clusters lengths.*

Description

This function draws a barplot of the total length of the 1D clusters for each events serie among several events series of same length and aligned. The barplot is expressed in frequencies of different percent ranges of the event series length detected as positive, resp. negative, cluster.

Usage

```
barplot(height,...)
```

Arguments

height	a graphscan object containing 1D cluster analysis.
...	arguments to be passed to methods.

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

See Also

graphscan_plot

Examples

```
## Not run:
dna_file<-list.files(path=system.file("extdata",package="graphscan"),
 pattern="fna",full.names=TRUE)
g1<-graphscan_1d(data=dna_file)
g1<-cluster(g1)
barplot(g1)

## End(Not run)
```

`cluster` *Performs cluster analysis on 'graphscan' class object.*

Description

This function performs cluster detections for both 'graphscan_1d' and 'graphscan_nd' objects. For 'graphscan_nd' objects both Kulldorff and Cucala indices are computed.

Usage

```
cluster(gr, n_simulation = NULL, cluster_analysis = NULL,
memory_size = 2000)
```

Arguments

`gr`	an object of class graphscan.
`n_simulation`	number of simulations (default value 199) to compute the p-value. This value is set during the generation of the 'graphscan' object or redefined by this argument.
`cluster_analysis`	type of cluster detection. Possible values are "positive", "negative" or "both". For nD detection only "positive" is possible.
`memory_size`	memory size (default value 2000) to use for the simulation, in mega-bytes(mb, 10^6 bytes). Possible values are integers > 10, and you should try to adapt it to the memory currently available (in the RAM) of your computer.

Details

This is the main function to run the cluster detection analysis from data and parameters contained in an object of class graphscan. The data come from the slot 'data' and the parameters from the slot 'param' of the 'gr' object. The results of the analysis are saved in the slot 'cluster' of the returned object (see value). The analysis for 1D data returns only the concentration index of Cucala. Analysis for nD data returns both the Cucala and the Kulldorff concentration index.

Value

Returns a 'graphscan' object containing the analysis in the slot 'cluster'. For 1D data, the slot 'cluster' is a list with three matrices 'cluster_1d_raw', 'cluster_1d' and 'cluster_1d_description'. 'cluster_1d_raw' is a matrix with 7 columns containing the raw results of the analysis. 'xleft' and 'xright' columns indicate the boundaries of each detected cluster. 'index' is the concentration index of Cucala and 'pvalue' is the significance of the cluster. 'positivity' is a boolean indicating if the cluster is positive or negative. 'id_segment' and 'id_serie' are the identifiers respectively of the clusters and the events series. 'cluster_1d' is a matrix with 9 columns containing treated results. Indeed, some clusters are cut into several pieces called segments because they contain one or more included cluster. This matrix indicates for each cluster the start ('xleft') and the end ('xright') of the non-overlapping segments (in 'cluster_1d_raw' matrix clusters are composed by only one segment potentialy overlapping with other detected clusters). Columns 'index','pvalue','positivity', 'id_segment' and 'id_serie' are the same than in 'cluster_1d_raw' matrix. The column 'n_segment' indicate how many segments compose the cluster and 'length' is size of each segment. 'cluster_1d_description' is a matrix with 4 columns giving general informations on all the clusters. 'n_pos' and 'n_neg' give respectively the number of positive and negative clusters for each events series. 'l_pos' and 'l_neg' are respectively the ratio (in percent) of positive and negative clusters total length. For nD data, the slot 'cluster' is a list with two 'SpatialPointsDataFrame' named 'cluster_nd_cucala' and 'cluster_nd_kulldorff' and a vector of characters named 'cluster_nd_description'. The two 'SpatialPointsDataFrame' objects contain the points of the significant cluster, the 'index' of concentration of Cucala or Kulldorff, the 'radius' of circles to draw the cluster area, the 'pvalue', the number of cases and controls present in the cluster. The vector 'cluster_nd_description' gives a brief description. 'memory_size' is used to limit the number of parallel simulations, due to the possible big memory consumption of a simulation. Indeed, without limitation it could use swap memory in the hard drive and highly decrease the performance of the algorithm.The nd algorithm uses kd-tree for the calculation, and is parallelized, so it has a theorical $O(nlog^2n)$ complexity and can work with millions of points up to 15 dimensions on a desktop computer in raisonnable time (for example: 200 simulations on a 2D dataset of 1 million of control points and 30'000 case points has been done in 41 minutes using 3 threads at 2 GHz each).

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn DeGoer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

See Also

graphscan_1d, graphscan_nd, plot, barplot

Examples

```
## Not run:
```

```
# 1d example with 2 fasta format files
# containing each 2 DNA aligned sequences.
dna_file<-list.files(path=system.file("extdata",package="graphscan"),
                     pattern="fna",full.names=TRUE)
g1<-graphscan_1d(data=dna_file)
g1<-cluster(g1)

# 2d example
data(france_two_clusters)
g3<-graphscan_nd(data=france_two_clusters)
g3<-cluster(g3)
graphscan_plot(g3,map=france)

## End(Not run)
```

`events_series` *A 1D cluster example: events series example.*

Description

A set of 9 series of 1000 events generated for the graphscan package.

Usage

```
data(events_series)
```

Format

Data is a list of 9 vectors of reals and a list named 'normalisation_factor' containing the normalisation factor of each events serie.

Source

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn DeGoer.

See Also

graphscan_1d

Examples

```
data(events_series)
g1<-graphscan_1d(data=events_series,
normalisation_factor=normalisation_factor)
```

france_two_clusters *A 2D cluster example: France with two clusters.*

Description

This data set is a 2D example with 1000 points ditributed on the French territory: 26 cases and 974 controls. The outline of France is included in object "france".

Usage

```
data(france_two_clusters)
```

Format

"france_two_clusters" and "france" are of class "SpatialPointsDataFrame"

Source

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer.

See Also

graphscan_nd

Examples

```
data(france_two_clusters)
sp::plot(france)
points(france_two_clusters[france_two_clusters$cases==1,],
pch=16,col="red")
points(france_two_clusters[france_two_clusters$cases==0,],
pch=16,cex=0.2,col="green")
```

graphscan-class *Class "graphscan"*

Description

Class of graphscan objects used to store data, parameters and results of cluster analysis with the method of Cucala.

Objects from the Class

Objects can be created by calls of the form "graphscan_1d(data, ...)"
or "graphscan_nd(data, ...)".

Slots

`data`: Object of class `"list"`, containing the data as events series.

`param`: Object of class `"list"`, containing the parameters used in the analysis.

`cluster`: Object of class `"list"`, containing the results.

Methods

cluster `signature(gr = "graphscan")`: performed the cluster analysis.

barplot `signature(x = "graphscan")`: draw an barplot of the length of clusters in multiple events series analysis.

graphscan_plot `signature(x = "graphscan")`: plot clusters localisations or the 1D events distributions.

print `signature(x = "graphscan")`: print informations about a graphscan object.

show `signature(object = "graphscan")`: print informations about a graphscan object.

summary `signature(object = "graphscan")`: summary for graphscan objects.

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Kulldorff, M. 1997. A spatial scan statistic, Commun. Statist. - Theory Meth., 26, 6, p. 1481-1496.

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

See Also

graphscan_1d, graphscan_nd

Examples

```
## Not run:
data(events_series)
g4<-graphscan_1d(events_series,normalisation_factor=normalisation_factor)
g4<-cluster(g4)
graphscan_plot(g4)

## End(Not run)
```

graphscan_1d *Creates objects of class 'graphscan' using 1D data.*

Description

This function produces objects of class graphscan_1d.

Usage

```
graphscan_1d(data, format = "fasta", events_series = "all",
id = NULL, n_simulation = 199, cluster_analysis = "both",
normalisation_factor = NULL, alpha = 0.05,
cluster_user_choice = "positive")
```

Arguments

data	main argument to define the format of the data. 'data' can be a vector of character string corresponding to files names of aligned DNA sequences. In this case, the format can be precised with argument 'format'. 'data' can be also a list of class 'DNAbin' produced by 'read.dna' function of 'ape' package. In all cases, DNA sequences must be aligned. Finally 'data' can be a 'list' of numeric vector containing the positions of events. This list is called a series of events. These events are not be necessarily on the same segment and not also necessarily on a [0,1] segment. The argument 'normalisation_factor' allows to fix the upper and lower bounds of each events series.
format	a character string corresponding to the format of the DNA sequences contained in files of argument 'data'. This is used by 'read.dna' function of 'ape' package. Possibles values are "interleaved","sequential","clustal" or "fasta" (default).
events_series	used if 'data' is a set of files names of aligned DNA sequences or list of class 'DNAbin'. 'events_series' can be a list of the form 'list(A,B)' where 'A' and 'B' corresponding to 2 vectors of sequences identifiants. The crossing AxB product is made to obtain a list of the series of events corresponding to the comparison between each sequence from 'A' to each sequence from 'B'. 'events_series' can be also a character string containing "all", in this case all possible comparison between sequences is made.
id	a character string corresponding to the prefix used to create the names of the events series.
n_simulation	number of simulations (default value 199) used to compute the p-values of clusters in a Monte-Carlo process. The value of 'n_simulation' is stored in the slot 'param' and can be modified by the function 'cluster'.
cluster_analysis	a character string corresponding to "positive", "negative" or "both" (default value) to detect respectively only the positives clusters, only the negatives clusters or both positives and negatives clusters. The value of 'cluster_analysis' is stored in the slot 'param' and can be modified by the function 'cluster'.

`normalisation_factor`
: a list of vectors with a size equal to the number of events series. Each vector contains 2 integers: the minimum and the maximum for the events positions of series of events.

  The maximum is the length of the DNA sequences if 'data' argument is a vector of character or an object of class "DNAbin". In these cases, the 'normalisation_factor' is automatically computed by the function 'graphscan_1d'.

  If 'data' is a 'list' of numeric vector containing the positions of events the 'normalisation_factor' must be specified as a 'list' containing the upper and lower bounds of each events series. The values of 'normalisation_factor' are stored in the slot 'param'.

`alpha`
: the threshold of significance (p-value) for keeping the candidate clusters. The value of 'alpha' is stored in the slot 'param'.

`cluster_user_choice`
: use if 'cluster_analysis="both"'. 'cluster_user_choice' is a string character corresponding to "positive" (default value), "negative" or "random". If two candidates clusters one positive and one negative have the same p-value this argument indicates how to choose between these 2 clusters. The value of 'cluster_user_choice' is stored in the slot 'param'.

Details

This function implements a statistical method for detecting clusters in 1D proposed by Cucala (2008). This hypothesis multiple scan statistic with variable window in 1D can detect positive clusters only, negative clusters only, or simultaneously positive or negative clusters. Positive clusters correspond to a particularly high concentration in events, while negative clusters correspond to a particularly low concentration in events. The concentration index of Cucala is based on the properties of the distances between oder statistics under the hypothesis of uniform distribution of the events. The 1D functions are adapted to study the repartition of mutations along the genome and detect zones with numerous or few mutations when comparing two aligned DNA sequences. When studying a population, i.e. several DNA sequences, pair comparisons can be created and analysed.

Value

'graphscan_1d' returns an object of class 'graphscan' with 3 slots:

`param`
: this slot contains the informations about data and the parameters used to perform the analysis.

`data`
: this slot contains data of events series as a list 'x' of numeric vectors.

`cluster`
: this slot contains the results of the analysis. For 1d, three matrices 'cluster_1d_raw', 'cluster_1d' and 'cluster_1d_description' (see 'cluster' function for more details).

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn DeGoer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

See Also

cluster, graphscan_2d

Examples

```
# example with 2 fasta format files containing each
# 2 DNA aligned sequences.
# the output object contain 6 events series.
dna_file<-list.files(path=system.file("extdata",package="graphscan"),
 pattern="fna",full.names=TRUE)
g1<-graphscan_1d(data=dna_file)

# to perform only 4 comparisons between DNA sequences
# 1 vs 3, 1 vs 4, 2 vs 3 and 2 vs 4.
g2<-graphscan_1d(data=dna_file,events_series=list(1:2,3:4))

# example with 'DNABin' class object :
require(ape)
data(woodmouse)
g3<-graphscan_1d(data=woodmouse)

# example with a list of 9 events series
data(events_series)
g4<-graphscan_1d(events_series,normalisation_factor=normalisation_factor)
```

---

graphscan_nd *Creates objects of class 'graphscan' using 2D or 3D data.*

Description

This function produces objects of class 'graphscan' using 'SpatialPointsDataFrame' class objects for 2D and 3D analysis.

Usage

```
graphscan_nd(data,field_cases = NULL,field_controls = NULL,
n_simulation = 199, alpha = 0.05)
```

Arguments

data	a 'SpatialPointsDataFrame' object containing coordinates of cases and controls points. The 'data.frame' of 'data' must contain two numeric fields: one for the number of cases for each point (named by default 'cases') and one for the number of controls for each point (named by default 'controls'). The minimal number of cases to perform the analysis is 2 and 50 for controls.
field_cases	a character string to define the name of the field containing the number of cases per points. If a field named 'cases' exists in the 'data.frame' this argument is optional.
field_controls	a character string to define the name of the field containing the number of controls per points. If a field named 'controls' exists in the 'data.frame' this argument is optional.
n_simulation	number of simulations (default value 199) used to compute the significance of the clusters i.e. the p-values computed with a Monte-Carlo process. The value of 'n_simulation' is stored in the slot 'param' and can be modified by the function 'cluster'.
alpha	the threshold of significance (p-value) to keep the candidate clusters. The value of 'alpha' is stored in the slot 'param'.

Details

This function implements a statistical method for detecting clusters in nD (2D or 3D) proposed by Cucala (2009). This flexible spatial scan test for case event data is implemented to detect the most significant positive cluster. The candidate clusters are build from connected components. There are the number of case minus one candidate clusters. The concentration index of Cucala is based on the properties of the distances between order statistics under the hypothesis of Poisson distribution of the cases. The analysis for nD data returns both the Cucala and the Kulldorff concentration index.

Value

'graphscan_nd' returns an object of class 'graphscan' with 3 slots:

param	this slot contains the informations about the data and the parameters used to perform the analysis.
data	this slot contains a list with one item 'x': the 'SpatialPointsDataFrame' of the 'data' argument.
cluster	this slot contains the results of the analysis: a list with three items two 'SpatialPointsDataFrame' named 'cluster_nd_cucala' and 'cluster_nd_kulldorff' and a vector of string named 'cluster_nd_description' (see 'cluster' function for more details).

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn DeGoer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Kulldorff, M. 1997. A spatial scan statistic, Commun. Statist. - Theory Meth., 26, 6, p. 1481-1496.

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

See Also

graphscan_1d, cluster

Examples

```
data(france_two_clusters)
g3<-graphscan_nd(data=france_two_clusters)
```

`graphscan_plot` *Plot clusters localisations or the 1D events distributions.*

Description

plot clusters localisations in a 1D, 2D or 3D space. In 1D, the distribution (in frequencies) of number of events on each position can be ploted.

Usage

```
graphscan_plot(x,events_series=1,map=NULL,indice="cucala",
sphere=TRUE,projection=FALSE,...)
```

Arguments

`x`	a graphscan object containing a cluster analysis.
`events_series`	a numeric or character vector containing 1D cluster identifiants of the events series to draw. If 'events_series="all"' the distribution of number of clusters on each position is plotted.
`map`	a 'SpatialPolygons' or 'SpatialPolygonsDataFrame' object to add to the 2D graph of clusters localisations (generally the outline of the studied region).
`indice`	a character string used in nD, to define the type of index to draw. Possible values are "cucala" (default) and "kulldorff".
`sphere`	a boolean ("TRUE" by default) to define if the spheres used to represent the 3D envelope of the cluster are drawn.
`projection`	a boolean ("FALSE" by default) to draw a projection in 2D of a 3D cluster. Three plots are drawn respectively for 'y vs x', 'z vs x' and 'z vs y'.
`...`	further arguments passed to or from other methods.

Details

To draw the distribution of number of events on each position in 1D, the events series must be of same length and aligned. The 3D representation of cluster use 'OpenGL' (http://www.opengl.org) an environment for interactive 2D and 3D graphics. If the number of cases points is very important old computers will display graphics quite slowly. In this case, use the option "projection=TRUE".

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

See Also

barplot

Examples

```
## Not run:
# 1D example:
require(ape)
data(woodmouse)
g1<-graphscan_1d(data=woodmouse)
g1<-cluster(g1)
graphscan_plot(g1,events_series=3)
dev.new()
graphscan_plot(g1,events_series="all")

# 2D example:
data(france_two_clusters)
g2<-graphscan_nd(data=france_two_clusters)
g2<-cluster(g2)
graphscan_plot(g2,map=france)

# 3D example:
data(sample3d)
g3<-graphscan_nd(data=sample3d)
g3<-cluster(g3)
graphscan_plot(g3,projection=TRUE) # 2D plot
graphscan_plot(g3) # 3D plot

## End(Not run)
```

sample3d *A 3D cluster example.*

Description

This data set is a 3D example with 1060 points ditributed into a cube. These points are devided into 60 cases and 1000 controls.

Usage

```
data(sample3d)
```

Format

"sample3d" is of class "SpatialPointsDataFrame"

Source

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer.

See Also

graphscan_nd

Examples

```
data(sample3d)
g1<-graphscan_nd(data=sample3d)
```

summary *Summarize a graphscan object.*

Description

This function give a summary of a graphscan object.

Usage

```
summary(object,...)
```

Arguments

object	a graphscan object containing or not a cluster analysis.
...	arguments to be passed to methods.

Author(s)

Robin Loche, Benoit Giron, David Abrial, Lionel Cucala, Myriam Charras-Garrido, Jocelyn De-Goer. Maintainer: David Abrial <graphscan@clermont.inra.fr>

References

Cucala, L. 2008. A hypothesis-free multiple scan statistic with variable window, Biometrical Journal, 2, p. 299-310.

Cucala, L. 2009. A flexible spatial scan test for case event data, Computational Statistics and Data Analysis, 53, p. 2843-2850.

Examples

```
## Not run:
dna_file<-list.files(path=system.file("extdata",package="graphscan"),
 pattern="fna",full.names=TRUE)
g1<-graphscan_1d(data=dna_file)
summary(g1)

## End(Not run)
```

Index