Package 'DClusterm'

February 12, 2017

Type Package

Title Model-Based Detection of Disease Clusters

Version 0.1

Date 2017-02-10

Author Virgilio Gomez-Rubio, Paula Esther Moraga Serrano, Barry Rowlingson

Maintainer Virgilio Gomez-Rubio <virgilio.gomez@uclm.es>

Depends R (>= 2.10), parallel, sp, spacetime, DCluster

Suggests INLA

Imports methods, xts, lme4, pscl, RColorBrewer, gridExtra, latticeExtra

Description Model-based methods for the detection of disease clusters using GLMs, GLMMs and zero-inflated models.

Additional_repositories https://www.math.ntnu.no/inla/R/stable

License GPL-3

LazyLoad yes

LazyData yes

Collate 'Functions1PAU.R' 'Functions2PAU.R' 'glm.isclusterPAU.R' 'knutils.R'

RoxygenNote 6.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-12 08:31:01

R topics documented:

`brainNM` *Brain cancer in New Mexico, USA, 1973-1991.*

Description

This data set contains the number of incident brain cancer cases in the 32 counties of New Mexico, USA, and each year of the period 1973-1991, and the location of Los Alamos National Laboratory. In addition, the data set also includes for each county and year information about the expected cases, the Standardized Morbidity Ratio (SMR), the FIPS...

Usage

```
data(brainNM)
```

Format

brainst: A STFDF object containing the following information for each county and year:

Observed	Number of observed brain cancer cases
Expected	Number of expected brain cancer cases. Standardisation is done taking the whole time-period and not year-ly to
SMR	Standardized Morbidity Ratio (observed/expected)
Year	Year
FIPS	FIPS Code
ID	ID (from 1 to 32)
IDLANL	Inverse distance to Los Alamos National Laboratory
IDLANLre	Re-scaled Inverse distance to Los Alamos National Laboratory (i.e., IDLANL/mean(IDLANL))

losalamos: A SpatialPoints object which contains the location (in long/lat) of Los Alamos National Laboratory obtained from the Wikipedia: -106.298333, 35.881667.

Source

Data have been downlodad from the SatScan website. Boundaries have been obtained from the U.S. Census Bureau. Cibola and Valencia counties has been merged together.

References

SatScan (c). http://www.spatstat.org

Kulldorff, M., W. F. Athas, E. J. Feurer, B. A. Miller, and C. R. Key (1998). Evaluating cluster alarms: a space-time scan statistic and brain cancer in los alamos, new mexico. American Journal of Public Health 88, 1377-1380.

`CalcStatClusterGivenCenter`

Calls the function to obtain the cluster with the maximum log-likelihood ratio or minimum DIC of all the clusters with the same center and start and end dates.

Description

This function orders the regions according to the distance to a given center and selects the regions with distance to the center less than sqrt(rr). Then it calls glmAndZIP.iscluster() to obtain the cluster with the maximum log-likelihood ratio or minimum DIC of all the clusters with the same center and start and end dates, and where the maximum fraction of the total population inside the cluster is less than fractpop.

Usage

```
CalcStatClusterGivenCenter(point, stfdf, rr, minDateCluster, maxDateCluster,
  fractpop, model0, ClusterSizeContribution)
```

Arguments

`point`	vector with the coordinates of the center of the cluster.
`stfdf`	spatio-temporal class object containing the data.
`rr`	square of the maximum radius of the cluster.
`minDateCluster`	start date of the cluster.
`maxDateCluster`	end date of the cluster.
`fractpop`	maximum fraction of the total population inside the cluster.
`model0`	Initial model (including covariates).
`ClusterSizeContribution`	Variable used to check the fraction of the population at risk in the cluster This can be "glm" for generalized linear models (glm stats), "glmer" for generalized linear mixed model (glmer lme4), "zeroinfl" for zero-inflated models (zeroinfl pscl), or "inla" for generalized linear, generalized linear mixed or zero-inflated models.

Value

vector containing the coordinates of the center, the size, the start and end dates, the log-likelihood ratio or DIC, the p-value and the risk of the cluster with the maximum log-likelihood ratio or minimum DIC.

`CalcStatsAllClusters` *Obtains the clusters with the maximum log-likelihood ratio or minimum DIC for each center and start and end dates.*

Description

This function explores all possible clusters changing their center and start and end dates. For each center and time periods, it obtains the cluster with the maximum log-likelihood ratio or minimum DIC so that the maximum fraction of the total population inside the cluster is less than fractpop, and the maximum distance to the center is less than radius.

Usage

```
CalcStatsAllClusters(thegrid, CalcStatClusterGivenCenter, stfdf, rr,
  typeCluster, sortDates, idMinDateCluster, idMaxDateCluster, fractpop, model0,
  ClusterSizeContribution, numCPUS)
```

Arguments

`thegrid`	grid with the coordinates of the centers of the clusters explored.
`CalcStatClusterGivenCenter`	function to obtain the cluster with the maximum log-likelihood ratio of all the clusters with the same center and start and end dates
`stfdf`	spatio-temporal class object containing the data.
`rr`	square of the maximum radius of the cluster.
`typeCluster`	type of clusters to be detected. "ST" for spatio-temporal clusters or "S" spatial clusters.
`sortDates`	sorted vector of the times where disease cases occurred.
`idMinDateCluster`	index of the closest date to the start date of the cluster in the vector sortDates
`idMaxDateCluster`	index of the closest date to the end date of the cluster in the vector sortDates
`fractpop`	maximum fraction of the total population inside the cluster.
`model0`	Initial model (including covariates). This can be "glm" for generalized linear models (glm stats), "glmer" for generalized linear mixed model (glmer lme4), "zeroinfl" for zero-inflated models (zeroinfl pscl), or "inla" for generalized linear, generalized linear mixed or zero-inflated models.
`ClusterSizeContribution`	Variable used to check the fraction of the population at risk in the cluster
`numCPUS`	Number of cpus used when using parallel to run the method. If parallel is not used numCPUS is NULL.

Value

data frame with information of the clusters with the maximum log-likelihood ratio or minimum DIC for each center and start and end dates. It contains the coordinates of the center, the size, the start and end dates, the log-likelihood ratio or DIC, the p-value and the risk of each of the clusters.

`computeprob` *Computes the probability that a model parameter is <=k from inla marginals*

Description

This function will be used to calculate the P(coeficient variable cluster <=0)

Usage

```
computeprob(func, k)
```

Arguments

`func`	is the inla marginals of the model parameter
`k`	is the cutoff

Value

probability model coefficient <=k

`CreateGridDClusterm` *Creates grid over the study area.*

Description

If the argument thegrid of DetectClustersModel() is null, this function is used to create a rectangular grid with a given step. If step is NULL the step used is equal to 0.2*radius. The grid contains the coordinates of the centers of the clusters explored.

Usage

```
CreateGridDClusterm(stfdf, radius, step)
```

Arguments

`stfdf`	spatio-temporal class object containing the data.
`radius`	maximum radius of the clusters.
`step`	step of the grid.

Value

two columns matrix where each row represents a point of the grid.

`DetectClustersModel` *Detects clusters and computes their significance.*

Description

Searches all possible clusters with start and end dates within minDateUser and maxDateUser, so that the maximum fraction of the total population inside the cluster is less than fractpop, and the maximum distance to the center is less than radius. The search can be done for spatial or spatio-temporal clusters. The significance of the clusters is obtained with a Monte Carlo procedure or based on the chi-square distribution (glm, glmer or zeroinfl models) or DIC (inla models).

Usage

```
DetectClustersModel(stfdf, thegrid = NULL, radius = Inf, step = NULL,
  fractpop, alpha, typeCluster, minDateUser = min(time(stfdf@time)),
  maxDateUser = max(time(stfdf@time)), R = NULL, model0,
  ClusterSizeContribution = "Population")
```

Arguments

`stfdf`	spatio-temporal class object containing the data. See STFDF-class spacetime for details. It contains an object of class Spatial with the coordinates, a time object holding time information, an endTime vector of class POSIXct holding end points of time intervals, and a data.frame with vectors Observed, Expected and potential covariates in each location and time. Note that the function DetectClustersModel does not use the endTime vector. We can define endTime, for example, as the vector of class POSIXct which contains the same dates as the ones contained in the time object.
`thegrid`	two-columns matrix containing the points of the grid to be used. If it is null, a rectangular grid is built.
`radius`	maximum radius of the clusters.
`step`	step of the thegrid built.
`fractpop`	maximum fraction of the total population inside the cluster.
`alpha`	significance level used to determine the existence of clusters.
`typeCluster`	type of clusters to be detected. "ST" for spatio-temporal or "S" spatial clusters.
`minDateUser`	start date of the clusters.
`maxDateUser`	end date of the clusters.
`R`	If the cluster's significance is calculated based on the chi-square distribution or DIC, R is NULL. If the cluster's significance is calculated using a Monte Carlo procedure, R represents the number replicates under the null hypothesis.
`model0`	Initial model (including covariates).

ClusterSizeContribution

Indicates the variable to be used as the population at risk in the cluster. This is the variable name to be used by 'fractpop' when checking the fraction of the population inside the cluster. The default column name is 'Population'. This can be "glm" for generalized linear models (glm stats), "glmer" for generalized linear mixed model (glmer lme4), "zeroinfl" for zero-inflated models (zeroinfl pscl), or "inla" for generalized linear, generalized linear mixed or zero-inflated models.

Value

data frame with information of the detected clusters ordered by its log-likelihood ratio value or DIC. Each row represents the information of one of the clusters. It contains the coordinates of the center, the size, the start and end dates, the log-likelihood ratio or DIC, the p-value, the risk of the cluster, and a boolean indicating if it is a cluster (TRUE in all cases).

References

Bilancia M, Demarinis G (2014) Bayesian scanning of spatial disease rates with the Integrated Nested Laplace Approximation (INLA). Statistical Methods & Applications 23(1): 71 - 94. http://dx.doi.org/10.1007/s10260-013-0241-8

Jung I (2009) A generalized linear models approach to spatial scan statistics for covariate adjustment. Statistics in Medicine 28(7): 1131 - 1143.

Fast Bayesian classification for disease mapping and the detection of disease clusters (2017) Gomez-Rubio V, Molitor J, Moraga P. Submitted.

Examples

```
library("DClusterm")
library("xts")
data("NY8")

NY8$Observed <- round(NY8$Cases)
NY8$Expected  <- NY8$POP8 * sum(NY8$Observed) / sum(NY8$POP8)

NY8$x <- coordinates(NY8)[, 1]
NY8$y <- coordinates(NY8)[, 2]

NY8st <- STFDF(as(NY8, "SpatialPolygons"), xts(1, as.Date("1972-01-01")),
NY8@data, endTime = as.POSIXct(strptime(c("1972-01-01"), "%Y-%m-%d"),
tz = "GMT"))

#Model to account for covariates
ny.m1 <- glm(Observed ~ offset(log(Expected)) + PCTOWNHOME + PCTAGE65P +
PEXPOSURE, family = "poisson", data = NY8)

#Indices of areas that are possible cluster centres
idxcl <- c(120, 12, 89, 139, 146)

#Cluster detection adjusting for covariates
ny.cl1 <- DetectClustersModel(NY8st,
```

```
thegrid = as.data.frame(NY8)[idxcl, c("x", "y")],
fractpop = 0.15, alpha = 0.05,
typeCluster = "S", R = NULL, model0 = ny.m1,
ClusterSizeContribution = "POP8")

#Display results
ny.cl1
```

`get.stclusters` *Gets areas in a spatio-temporal cluster*

Description

This function is similar to get.knclusters but it also allows for spatio-temporal clusters.

Usage

```
get.stclusters(stfdf, results)
```

Arguments

`stfdf`	A sp or spacetime object with the information about the data.
`results`	Results from a call to DetectClusterModel

Value

A list with as many elements as clusters in 'results'

`glmAndZIP.iscluster` *Obtains the cluster with the maximum log-likelihood ratio or minimum DIC of all the clusters with the same center and start and end dates.*

Description

This function constructs all the clusters with start date equal to minDateCluster, end date equal to maxDateCluster, and with center specified by the first element of idxorder, so that the maximum fraction of the total population inside the cluster is less than fractpop, and the maximum distance to the center is less than radius. For each one of these clusters, the log-likelihood ratio test statistic for comparing the alternative model with the cluster versus the null model of no clusters (if model is glm, glmer or zeroinfl), or the DIC (if model is inla) is calculated. The cluster with maximum value of the log-likelihood ratio or minimum DIC is returned.

Usage

```
glmAndZIP.iscluster(stfdf, idxorder, minDateCluster, maxDateCluster, fractpop,
  model0, ClusterSizeContribution)
```

Arguments

stfdf	a spatio-temporal class object containing the data.
idxorder	a permutation of the regions according to their distance to the current center.
minDateCluster	start date of the cluster.
maxDateCluster	end date of the cluster.
fractpop	maximum fraction of the total population inside the cluster.
model0	Initial model (including covariates).
ClusterSizeContribution	Variable used to check the fraction of the population at risk in the cluster This can be "glm" for generalized linear models (glm stats), "glmer" for generalized linear mixed model (glmer lme4), "zeroinfl" for zero-inflated models (zeroinfl pscl), or "inla" for generalized linear, generalized linear mixed or zero-inflated models.

Value

vector containing the size, the start and end dates, the log-likelihood ratio or DIC, the p-value and the risk of the cluster with the maximum log-likelihood ratio or minimum DIC.

---

knbinary *Constructs data frame with clusters in binary format.*

Description

This function constructs a data frame with number of columns equal to the number of clusters. Each column is a binary representation of one of the clusters. The position i of the column is equal to 1 if the polygon i is in the cluster or 0 if it is not in the cluster.

Usage

```
knbinary(datamap, knresults)
```

Arguments

datamap	data of the SpatialPolygonsDataFrame with the polygons of the map.
knresults	data frame with information of the detected clusters. Each row represents the information of one of the clusters. It contains the coordinates of the center, the size, the start and end dates, the log-likelihood ratio, a boolean indicating if it is a cluster (TRUE in all cases), and the p-value of the cluster.

Value

data frame where the columns represent the clusters in binary format. The position i of the column is equal to 1 if the polygon i is in the cluster or 0 if it is not in the cluster.

`mergeknclusters` *Merges clusters so that they are identifed as levels of a factor.*

Description

Given a data frame with clusters that do not overlap this function merges the clusters and construct a factor. The levels of the factor are "NCL" if the polygon of the map is not in any cluster, and "CL" if the polygon i is in cluster i.

Usage

```
mergeknclusters(datamap, knresults, indClustersPlot)
```

Arguments

`datamap`	data of the SpatialPolygonsDataFrame with the polygons of the map.
`knresults`	Data frame with information of the detected clusters. Each row represents the information of one of the clusters. It contains the coordinates of the center, the size, the start and end dates, the log-likelihood ratio, a boolean indicating if it is a cluster (TRUE in all cases), and the p-value of the cluster.
`indClustersPlot`	rows of knresults that denote the clusters to be plotted.

Value

factor with levels that represent the clusters.

`Navarre` *Brain cancer in males in Navarre, Spain, 1988-1994.*

Description

This data set contains the male mortality due to brain cancer in the 40 basic health zones (BHZ) in Navarre over the period 1988-1994, and the neighborhood structure of the BHZ in Navarre. In addition, the data set also includes information about the location of the BHZ, the expected cases, the Standardized Mortality Ratio (SMR), relative risk estimates and 95% confidence intervals.

Usage

```
data(Navarre)
```

Format

brainnav: A SpatialPolygonsDataFrame with 40 polygons representing the basic health zones (BHZ) in Navarre, and the following information about each BHZ:

ZBS	
Basic Health Zone Code NAME	Name
OBSERVED	Number of observed brain cancer cases in males
EXPECTED	Number of expected brain cancer cases in males. They are computed using indirect age-sta
RISK	Relative Risk Estimates
RISKLL	Relative 95% confidence interval, lower limit
RISKUL	Relative 95% confidence interval, upper limit
SMR	Standardized Mortality Ratio (OBSERVED/EXPECTED)
x	x coordinate
y	y coordinate

brainnavnb: A neighbor (nb) object which contains the index numbers of the neighbors of each BHZ.

Source

Data set obtained from Ugarte et al. (2004). Boundaries downloaded in shapefile format from http://idena.navarra.es. These have been thinned to reduce space use.

References

Ugarte, M. D., B. Ibáñez, and A. F. Militino (2004). Testing for poisson zero a inflation in disease mapping. Biometrical Journal 46 (5), 526-539.

Ugarte, M. D., B. Ibáñez, and A. F. Militino (2006). Modelling risks in a disease mapping. Statistical Methods in Medical Research 15, 21-35.

NY8 *Leukemia in an eight-county region of upstate New York, 1978-1982.*

Description

This data set provides the number of incident leukemia cases per census tract in an eight-county region of upstate New York in the period 1978-1982. In addition, the data set also includes information about the location of the census tracts, the population in 1980, the inverse of the distance to the nearest Trichloroethene (TCE) site, the percentage of people aged 65 or more, and the percentage of people who own their home.

The dataset also provides the locations of the TCE sites.

Usage

```
data(NY8)
```

Format

A SpatialPolygonsDataFrame with 281 polygons representing the census tracts, and the following information about each census tract:

AREANAME	Name
AREAKEY	Identifier
X	x coordinate
Y	y coordinate
POP8	Population in 1980
TRACTCAS	Number of leukemia cases rounded to 2 decimals
PROPCAS	Ratio of the number of leukemia cases to the population in 1980
PCTOWNHOME	Percentage of people who own their home
PCTAGE65P	Percentage of people aged 65 or more
Z	
AVGIDIST	
PEXPOSURE	Inverse of the distance to the nearest TCE site
Cases	Number of leukemia cases
Xm	x coordinate (in meters)
Ym	y coordinates(in meters)
Xshift	Shifted Xm coordinate
Yshift	Shifted Ym coordinate

Source

Waller and Gotway (2004) and Bivand et al. (2008)

References

Bivand, R.S., E. J. Pebesma and V. Gómez-Rubio (2008). Applied Spatial Data Analysis with R. Springer.

Waller, L., B. Turnbull, L. Clark, and P. Nasca (1992). Chronic disease surveillance and testing of clustering of disease and exposure: application to leukemia incidence in tce-contamined dumpsites in upstate New York. Environmetrics 3, 281-300

Waller, L. A. and C. A. Gotway (2004). Applied Spatial Statistics for Public Health Data. John Wiley & Sons, Hoboken, New Jersey. http://web1.sph.emory.edu/users/lwaller/WGindex.htm

---

`PlotClustersNoOverlap` *Plots the clusters that do not overlap.*

Description

This function plots the detected clusters that do not overlap. There are as many windows as different start dates. All clusters with the same start date are represented in the same window.

Usage

```
PlotClustersNoOverlap(statsAllClustersNoOverlap, colors, map)
```

Arguments

statsAllClustersNoOverlap	data frame with information of the detected clusters that no overlap. Each row represents the information of one of the clusters. It contains the coordinates of the center, the size, the start and end dates, the log-likelihood ratio, a boolean indicating if it is a cluster (TRUE in all cases), and the p-value of the cluster.
colors	vector with the colors of the clusters.
map	SpatialPolygonsDataFrame with the polygons of the map.

Value

plots of the detected clusters for each start date.

SelectStatsAllClustersNoOverlap *Removes the overlapping clusters.*

Description

Function DetectClustersModel() detects duplicated clusters. This function reduces the number of clusters by removing the overlapping clusters.

Usage

```
SelectStatsAllClustersNoOverlap(stfdf, statsAllClusters)
```

Arguments

stfdf	spatio-temporal class object containing the data.
statsAllClusters	data frame with information of the detected clusters obtained with DetectClustersModel().

Value

data frame with the same information than statsAllClusters but only for clusters that do not overlap.

`SetVbleCluster` *Constructs a variable that indicates the locations and times that pertain to a cluster.*

Description

This function constructs a variable that indicates the locations and times that pertain to a cluster. Each position of the variable is equal to 1 if it corresponds to a location and time inside the cluster, and 0 otherwise. This is one of the explanatory variables used in the glmAndZIP.iscluster function to model the observed cases.

Usage

```
SetVbleCluster(stfdf, idTime, idSpace)
```

Arguments

`stfdf`	spatio-temporal class object containing the data.
`idTime`	vector with the indexes of the stfdf object corresponding to the time inside the cluster.
`idSpace`	vector with the indexes of the stfdf object corresponding to the locations inside the cluster.

Value

vector with 1's or 0's that indicates the locations and times that pertain to a cluster.

`slimknclusters` *Remove overlapping clusters*

Description

This function slims the number of clusters down. The spatial scan statistic is known to detect duplicated clusters. This function aims to reduce the number of clusters by removing duplicated and overlapping clusters.

Usage

```
slimknclusters(d, knresults, minsize = 1)
```

Arguments

`d`	Data.frame with data used in the detection of clusters.
`knresults`	Object returned by function opgam() with the clusters detected.
`minsize`	Minimum size of cluster (default to 1).

Value

A subset of knresults with non-overlaping clusters of at least minsize size.

Index