Package 'CORE'

February 19, 2015

Type Package

Title Cores of Recurrent Events

Version 3.0

Author Alex Krasnitz, Guoli Sun

Maintainer Guoli Sun <guolisun87@gmail.com>

Description given a collection of intervals with integer start and end positions, find recurrently targeted regions and estimate the significance of finding. Randomization is implemented by parallel methods, either using local host machines, or submitting grid engine jobs.

License GPL-2

Imports parallel

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-30 00:12:08

R topics documented:

CORE *Cores of Recurrent Events*

Description

Given a collection of intervals s_1,...,s_N, find K intervals c_1,...,c_K which approximately minimize Sum_i Prod_k (1-E(s_i,c_k)), where E(s_i,c_k) is a geometric measure of association between s_i and c_k. Perform permutation tests to estimate the significance of finding.

Usage

```
CORE(dataIn, keep = NULL, startcol = "start", endcol = "end",
chromcol = "chrom", weightcol = "weight", maxmark = 1, minscore = 0,
pow = 1, assoc = c("I", "J", "P"), nshuffle = 0, boundaries = NULL,
seedme = sample(1e+08, 1), shufflemethod = c("SIMPLE", "RESCALE"),
tiny = -1, distrib = c("vanilla", "Rparallel","Grid"), njobs = 1,qmem=NA)
```

Arguments

dataIn	A matrix, a data frame or an object of class "CORE". If dataIn is a matrix or a data frame, it should have columns with names specified by the startcol and endcol arguments, otherwise the function exits with an error.
keep	A character vector. If dataIn is of class "CORE", keep specifies the names of items of dataIn to be kept at their input values. These values take precedence over the corresponding argument values as specified in the function call. keep is ignored if dataIn is not of class "CORE".
startcol	A character string. If dataIn is a matrix or a data frame, startcol specifies the name of the column containing start coordinates of the input intervals. Otherwise startcol is ignored.
endcol	A character string. If dataIn is a matrix or a data frame, endcol specifies the name of the column containing end coordinates of the input intervals. Otherwise endcol is ignored.
chromcol	A character string. If dataIn is a matrix or a data frame, chromcol specifies the name of the column containing chromosome numbers of the input intervals. Otherwise chromcol is ignored.
weightcol	A character string. If dataIn is a matrix or a data frame, weightcol specifies the name of the column containing initial weights of the input intervals. Otherwise weightcol is ignored.
maxmark	An integer for the maximal number of cores to be computed. The actual number of cores to be computed is the smaller of maxmark and the number of cores with scores exceeding minscore.
minscore	A single numeric value for the minimal allowed score of the cores to be reported.
pow	A single numeric value of at least 1 for the power parameter used in computing the association measure beween the cores and the input intervals (see Details).
assoc	A character specifying the type of association measure to be used (see Details).
nshuffle	An integer specifying the number of randomizations to be performed for estimating significance.
boundaries	A matrix or a data frame that must have three columns whose names are given by chromcol, startcol and endcol. These specify the chromosome numbers and their start and end positions (see Details).
seedme	An integer specifying the random number generator seed (see Details).
shufflemethod	A character string specifying the event randomization method used for estimation of significance. If "SIMPLE" (default), each event is placed at random with equal probability for any position where it can fit within chromosome boundaries. If "RESCALE", each event is placed at random in a randomly chosen

chromosome, and the event length is multiplied by the length ratio of the new to the original chromosome.

`tiny` A single numeric value specifying the weight below which events are removed from the input event set.

`distrib` A character string specifying the method of distributed computing used for estimation of significance. If "vanilla" (default), no distributed computing is performed. If "Rparallel", parallel computation with the local machine is performed using functions from CRAN core package parallel, with the number of worker processes being the smaller number of `njobs`,and `nshuffle`. If "Grid", parallel computation with grid engine is performed. The number of submitted array jobs, or cores that are distributed, is the smaller number of `njobs`,and `nshuffle`. When using "Grid", make sure you have write premission to the current work space.

`njobs` If distributed computing is used for estimation of significance, a single integer specifying the desired number of worker processes.

`qmem` A character string that can customize grid engine `qsub` command. The command decides memory size per core(each job). The default substring is "-l virtual_free=2G".

Details

The three measures of association specified by `assoc` are defined as follows (|| denotes the length of an interval). For "I" (inclusion) E(s_i,c_k) = (|c_k|/|s_i|)^pow if c_k is contained in s_i and 0 otherwise. For "J" (Jaccard) E(s_i,c_k) = J(s_i,c_k)^pow, where J is the Jaccard index. For "P" (piercing) E(s_i,c_k) = 1 if c_k is contained and 0 otherwise. In all cases the left (right) boundary of an optimal c_k is one of the left (right) boundaries in the set of input interval events. In addition, there are no event interval boundaries in the interior of an optimal c_k in case "P".

The `boundaries` argument is used for assessing statistical significance of the solution. If `boundaries` is not specified, the chromosome boundaries for each chromosome are taken to be the leftmost left and the rightmost right boundaries of all events in the chromosome.

If significance of finding is estimated, the random number generator stream, and hence the resultant estimate, only depends on `seedme` and is independent of the parallelization option chosen.

Value

An object of class "CORE" with the following items.

`input` A matrix with four columns called "chrom", "start", "end" and "weight", specifying the input interval events.

`call` A character string specifying the function call.

`coreTable` A matrix with columns named "start", "end" and "score", for start and end positions and CORE scores of the cores found by the algorithm.

`seedme` If significance estimate was performed, the random number generator seed.

`assoc` One of "I", "J" or "P", indicating the geometric measure of association used.

`shufflemethod` One of "SIMPLE" or "RESCALE", indicating the randomization method used.

p	A numeric vector of the length equal to the row dimension of coreTable containing estimated p-values for the cores.
simscores	A matrix with the row dimension equal to that of coreTable and nshuffle columns, containing core scores computed for nshuffle sets of randomized events.
minscore	A single numeric value for the minimal score of the reported cores.
maxmark	A single numeric value for the requested maximal number of cores to be computed.
tiny	A single numeric value for the weight below which events were removed from the input set.
pow	A single numeric value for the power used in computing the association measures.
boundaries	A matrix with three columns named "chrom", "start" and "end", indicating chromosome numbers and boundary positions used for estimation of significance.

Author(s)

Alex Krasnitz,Guoli Sun

Examples

```
#Compute 3 cores and perform no randomization
#(meaningless for estimate of significance).
data(testInputCORE)
data(testInputBoundaries)
myCOREobj<-CORE(dataIn=testInputCORE,maxmark=3,nshuffle=0,
boundaries=testInputBoundaries,seedme=123)
## Not run:
#Extend this computation to a much larger number of randomizations,
#using 2 cores of a host computer.
newCOREobj<-CORE(dataIn=myCOREobj,keep=c("maxmark","seedme","boundaries"),
nshuffle=20,distrib="Rparallel",njobs=2)
#When using "Grid", make sure you have write premission to the current
#work space.
newCOREobj<-CORE(dataIn=myCOREobj,keep=c("maxmark","seedme","boundaries"),
nshuffle=20,distrib="Grid",njobs=2)

## End(Not run)
```

---

testInputBoundaries *A table of chromosome boundary positions for DNA copy number analysis*

Description

The entire length of the genome was divided into 50009 bins, with chromosomes laid out in the usual genomic order: 1,...,22,X,Y. Each observation in the table provides the start and end bin numbers of each chromosome (chrom).

Usage

```
data(testInputBoundaries)
```

Format

A data frame with 24 observations on the following 3 variables.

`chrom` a numeric vector

`start` a numeric vector

`end` a numeric vector

References

Navin N, Kendall J, Troge J, Andrews P, Rodgers L, McIndoo J, Cook K, Stepansky A, Levy D, Esposito D et al. 2011. Tumour evolution inferred by single-cell sequencing. Nature 472(7341): 90-U119.

Examples

```
data(testInputBoundaries)
## maybe str(testInputBoundaries) ; plot(testInputBoundaries) ...
```

---

testInputCORE — *A table of DNA copy number gain events observed in 100 individual tumor cells*

Description

Each observation in the table corresponds to a DNA copy number gain event in one of 100 individual breast cancer cells. The entire length of the genome was divided into 50009 bins. An event is an interval in chromosome `chrom` whose start and end bin numbers are given by `start` and `end`.

Usage

```
data(testInputCORE)
```

Format

A data frame with 2490 observations on the following 3 variables.

`chrom` a numeric vector

`start` a numeric vector

`end` a numeric vector

References

Navin N, Kendall J, Troge J, Andrews P, Rodgers L, McIndoo J, Cook K, Stepansky A, Levy D, Esposito D et al. 2011. Tumour evolution inferred by single-cell sequencing. Nature 472(7341): 90-U119.

Examples

```
data(testInputCORE)
## maybe str(testInputCORE) ; plot(testInputCORE) ...
```

Index