Package 'CNprep'

January 20, 2025

Type Package

Title Pre-Process DNA Copy Number (CN) Data for Detection of CN Events

Version 2.2

Date 2022-05-23

Author Alex Krasnitz, Guoli Sun

Maintainer Guoli Sun <guolisun87@gmail.com>

Description DNA copy number data evaluation using both their initial form (copy number as a noisy function of genomic position) and their approximation by a piecewise-constant function (segmentation), for the purpose of identifying genomic regions where the copy number differs from the norm.

License GPL-2

Depends R (>= 2.10), parallel, mclust, rlecuyer, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-24 03:10:02 UTC

Contents

annotexample *Annotation table for ROMA CGH platform and human genome version 17.*

Description

Whole genome annotation table using Representational Oligonucleotide Microarray Analysis (ROMA) CGH platform, human genome version 17.

Usage

```
data(annotexample)
```

Format

A data frame with 83055 observations on the following 3 variables.

`PROBEID` a character vector

`CHROM` a numeric vector

`CHROM.POS` a numeric vector

Details

The values in the chromosome column are all integer, with 23 corresponding to X, 24 to Y and 25 to a set of non-human test probes.

Source

GEO accession GPL9775, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL9775

Examples

```
data(annotexample)
```

applyCNPmask *Apply a mask to a table of copy number events.*

Description

A mask is applied to amplified or deleted segments as tabulated in `segtable`. A decision whether to mask a segment is taken based on what portion of the segment is covered by the mask. A position is chosen at random within a segment to be masked, the flanking segments are extended to that position and the segment to be masked is indicated as such in the value returned.

Usage

```
applyCNPmask(segtable,chrom,startPos,endPos,startProbe,endProbe,
eventIndex,masktable,maskchrom,maskstart,maskend,maskindex,mincover=1,
indexvals=c(-1,1))
```

Arguments

`segtable`	A matrix or a data frame with columns named or enumerated by the values of `chrom`, `startPos`, `endPos`, `startProbe`, `endProbe`, `eventIndex`.
`chrom`	A character string specifying the name for the column in `segtable` tabulating the (integer) chromosome number for each segment.
`startPos`, `endPos`	Character strings or integers specifying the names or numbers of columns in `segtable` that tabulate the (integer) genomic start and end coordinates of each segment.
`startProbe`, `endProbe`	Character strings specifying the names of columns in `segtable` that tabulate the (integer) start and end postions of each segment in internal units such as probe numbers for data of CGH microarray origin.
`eventIndex`	A character string giving the name of a column in `segtable` where copy number variation status of the segments is tabulated.
`masktable`	A matrix or a data frame with columns named or enumerated as given by `maskchrom`, `maskstart`, `maskend`, `maskindex` and with rows corresponding to genomic intervals that comprise the mask.
`maskchrom`, `maskstart`, `maskend`	Character strings or integers specifying the names or numbers of columns in `masktable` that tabulate the chromosome number and (integer) genomic start and end coordinates of the intervals comprising the mask.
`maskindex`	A numeric vector corresponding to `eventIndex`,specifying copy number events status for measuring units.
`mincover`	A numeric value specifying the minimal portion of the segment that must be covered by the mask in order to trigger masking.
`indexvals`	A numeric vector of length 2 specifying the two values in `maskindex` to be matched with values in `eventIndex` to determine the events that are to be masked.

Details

Masking is performed separately for each value in `indexvals`. Segments (rows of `segtable`) with that value of `eventIndex` are examined for coverage by mask intervals with that value of `maskindex` in `masktable`. If the coverage is at least `mincover`, the segment is slated for masking, while its flanking segments are extended to a random point within the segment being masked.

Value

A matrix with same number of observations/rows as `segtable` and with following three columns:

startProbe, endProbe
: An integer vector for the start and end positions of the segments after masking.

toremove
: An integer vector whose values are 1 if the segment is masked and 0 otherwise.

Author(s)

Alex Krasnitz

Examples

```
## Not run:
data(segexample)
data(ratexample)
data(normsegs)
data(cnpexample)
segtable<-CNpreprocessing(segall=segexample[segexample[,"ID"]=="WZ1",],
ratall=ratexample,"ID","start","end",chromcol="chrom",bpstartcol="chrom.pos.start",
bpendcol="chrom.pos.end",blsize=50,minjoin=0.25,cweight=0.4,bstimes=50,
chromrange=1:22,distrib="Rparallel",njobs=2,modelNames="E",normalength=normsegs[,1],
normalmedian=normsegs[,2])
#form a eventIndex vector
eventIndex<-rep(0,nrow(segtable))
eventIndex[segtable[,"marginalprob"]<1e-4&segtable[,"negtail"]>
0.999&segtable[,"mediandev"]<0] <- -1
eventIndex[segtable[,"marginalprob"]<1e-4&segtable[,"negtail"]>
0.999&segtable[,"mediandev"]>0] <- 1
segtable<-cbind(segtable,eventIndex)
#form a cnpindex vector
namps17<-cnpexample[cnpexample[,"copy.num"]=="amp",]
aCNPmask<-makeCNPmask(imat=namps17,chromcol=2,startcol=3,endcol=4,
nprof=1203,uthresh=0.02,dthresh=0.008)
ndels17<-cnpexample[cnpexample[,"copy.num"]=="del",]
dCNPmask<-makeCNPmask(imat=ndels17,chromcol=2,startcol=3,endcol=4,
nprof=1203,uthresh=0.02,dthresh=0.008)
cnptable<-rbind(cbind(aCNPmask,cnpindex=1),cbind(dCNPmask,cnpindex=-1))
#run the CNP test
myCNPtable<-applyCNPmask(segtable,"chrom",startPos="chrom.pos.start",
endPos="chrom.pos.end","start","end","eventIndex",masktable=cnptable,"chrom",
maskstart="start",maskend="end",maskindex="cnpindex",mincover=0.005,indexvals=c(-1,1))

## End(Not run)
```

---

cnpexample *Example of a boundary positions table.*

Description

A table of genomic positions for DNA copy-number changing events, collected from genomes of 1203 individuals using Representational Oligonucleotide Microarray Analysis (ROMA) platform.

Usage

```
data(cnpexample)
```

Format

A data frame with 19188 rows and 4 columns.

`copy.num` a character vector indicating whether an event is a gain ("amp") or a loss ("del").

`chrom` a numeric vector indicating which chromosome the event is in.

`chrom.start` a numeric vector of event start positions.

`chrom.end` a numeric vector of event start positions.

Source

Strong association of de novo copy number mutations with autism. Sebat J, Lakshmi B, Malhotra D, Troge J, Lese-Martin C, Walsh T, Yamrom B, Yoon S, Krasnitz A, Kendall J, Leotta A, Pai D, Zhang R, Lee YH, Hicks J, Spence SJ, Lee AT, Puura K, Lehtimaki T, Ledbetter D, Gregersen PK, Bregman J, Sutcliffe JS, Jobanputra V, Chung W, Warburton D, King MC, Skuse D, Geschwind DH, Gilliam TC, Ye K, Wigler M. Science. 2007 Apr 20;316(5823):445-9.

Examples

```
data(cnpexample)
```

CNpreprocessing *Pre-process DNA copy number (CN) data for detection of CN events.*

Description

Description: The package evaluates DNA copy number data, using both their intitial form (copy number as a noisy function of genomic position) and their approximation by a piecewise-constant function (segmentation), for the purpose of identifying genomic regions where the copy number differs from the norm.

Usage

```
CNpreprocessing(segall, ratall = NULL, idcol = NULL, startcol = NULL,
endcol = NULL, medcol = NULL, madcol = NULL, errorcol = NULL,
chromcol = NULL, bpstartcol = NULL, bpendcol = NULL, annot = NULL,
annotstartcol = NULL, annotendcol = NULL, annotchromcol = NULL,
useend = F, blsize = NULL, minjoin = NULL, ntrial = 10, bestbic = -1e+07,
modelNames = "E", cweight = NULL, bstimes = NULL, chromrange = NULL,
myseed = 123, distrib = c("vanilla", "Rparallel"), njobs = 1,
normalength = NULL, normalmedian = NULL, normalmad = NULL,
normalerror = NULL)
```

Arguments

segall	A matrix or a data frame for segmented copy number profiles. It may have a character column, with a name specified by idcol, and/or numeric columns with names specified by startcol, endcol, medcol, madcol,errorcol ,chromcol, bpstartcol, bpendcol. Each row of segall corresponds to a segment belonging to one of the profiles to be pre-processed.
ratall	A matrix whose rows correspond to genomic positions and columns to copy number profiles. Its matrix elements are functions of copy number, most often log ratios of copy number to the expected standard value, such as 2 in diploid genomes.
idcol	A character string specifying the name for the column in segall tabulating the profile IDs.
startcol, endcol	Character strings specifying the names of columns in segall that tabulate the (integer) start and end postions of each segment in internal units such as probe numbers for data of CGH microarray origin.
medcol, madcol, errorcol	Character strings specifying the names of columns in segall that, for the function of copy number used in the study (typically log ratios), tabulate the (numeric) values for the function (medcol), a measure of its spread (madcol) and its error (errorcol) for the segment.
chromcol	A character string specifying the name for the column in segall tabulating the (integer) chromosome number for each segment.
bpstartcol, bpendcol	Character strings specifying the names of columns in segall that tabulate the (integer) genomic start and end coordinates of each segment.
annot	A matrix or a data frame that contains the annotation for the copy number measurement platform in the study. It is generally expected to contain columns with names specified by annotstartcol, annotendcol, annotchromcol.
annotstartcol, annotendcol, annotchromcol	Character strings specifying the names of columns in annot that tabulate the (integer) genomic start and end coordinates and the chromosome number for each copy number measuring unit, such as a probe in case of CGH microarrays.
useend	A single logical value specifying whether the segment end positions as given by the bpendcol of segall are to be looked up in the annotendcol column of annot (if useend=TRUE) or in the annotstartcol column (default).
blsize	A single integer specifying the bootstrap sampling rate of segment medians to generate input for model-based clustering. The number of times a segment is sampled is then given by the (integer) division of the segment length in internal units by blsize.
minjoin	A single numeric value between 0 and 1 specifying the degree of overlap above which two clusters will be joined into one.
ntrial	A single integer specifying the number of times a model-based clustering is attempted for each profile in order to achieve the highest Bayesian information criterion (BIC).

`bestbic` A single numeric value for initalizing BIC maximization. A large negative value is recommended. The default is `-1e7`.

`modelNames` A vector of character strings specifying the names of models to be used in model-based clustering (see package `mclust` for further details). The default is `"E"`.

`cweight` A single numeric value between 0 and 1 specifying the minimal share of the central cluster in each profile.

`bstimes` A single integer value specifying the number of time the median of each segment is sampled in order to predict the cluster assignment for the segment.

`chromrange` A numeric vector enumerating chromosomes from which segments are to be used for initial model-based clustering.

`myseed` A single integer value to seed the random number generator.

`distrib` One of `"vanilla"`, `"Rparallel"` to specify the distributed computing option for the cluster assignment step. For `"vanilla"` (default) no distributed computing is performed. For `"Rparallel"` the `parallel` package of `R` core is used for multi-core processing.

`njobs` A single integer specifying the number of worker jobs to create in case of distributed computation.

`normalength` An integer vector specifying the genomic lengths of segments in the normal reference data.

`normalmedian`, `normalmad`, `normalerror`
Numeric vectors, of the same length as `normalength`, specifying the segment values, value spreads and errors of the normal reference segments.

Details

Depending on the availability of input, the function will perform the following operations for each copy number profile.

If raw data are available in addition to segment start and end positions, median and MAD of each segment will be computed. For each profile, bootstrap sampling of the segment median values will be performed, and the sample will be used to estimate the error in the median for each segment. Model-dependent clustering (fitting to a gaussian mixture) of the sample will be performed. The central cluster (the one nearest the expected unaltered value) will be identified and, if necessary, merged with adjacent clusters in order to comprise the minimal required fraction of the data. Deviation of each segment from the center, its probabilty to belong to the central cluster and its marginal probability in the central cluster will be computed.

If segment medians or median deviations are available or have been computed, and, in addition, genomic lengths and average values are given for a collection of segments with unaltered copy number, additional estimates will be performed. If median values are available for the unaltered segments, the marginal probability of the observed median or median deviation in the unaltered set will be computed for each segment. Likewise, marginal probabilities for median/MAD and/or median/error will be computed if these statistics are available.

Value

The input `segall` data frame to which some or all of the following columns may be bound, depending on the availability of input:

segmedian	Median function of copy number
segmad	MAD for the function of copy number
mediandev	median function of copy number relative to its central value
segerr	error estimate for the function of copy number
segz	the probability that the segment is in the central cluster
marginalprob	marginal probability for the segment in the central cluster
negtail	the probability of finding the deviation as observed or larger in a collection of central segments
negtailnormad	the probability of finding the deviation/MAD as observed or larger in a collection of central segments
negtailnormerror	the probability of finding the deviation/error as observed or larger in a collection of central segments

Author(s)

Alex Krasnitz

Examples

```
data(segexample)
data(ratexample)
data(normsegs)
#small toy example
segtable<-CNpreprocessing(segall=segexample[segexample[,"ID"]=="WZ1",],
ratall=ratexample,"ID","start","end",chromcol="chrom",bpstartcol="chrom.pos.start",
bpendcol="chrom.pos.end",blsize=50,minjoin=0.25,cweight=0.4,bstimes=50,
chromrange=1:3,distrib="Rparallel",njobs=2,modelNames="E",
normalength=normsegs[,1],normalmedian=normsegs[,2])
## Not run:
#Example 1: 5 whole genome analysis, choosing the right format of arguments
segtable<-CNpreprocessing(segall=segexample,ratall=ratexample,"ID","start","end",
chromcol="chrom",bpstartcol="chrom.pos.start",bpendcol="chrom.pos.end",blsize=50,
minjoin=0.25,cweight=0.4,bstimes=50,chromrange=1:22,distrib="Rparallel",njobs=40,
modelNames="E",normalength=normsegs[,1],normalmedian=normsegs[,2])
#Example 2: how to use annotexample, when segment table does not have columns of
#integer postions in terms of  measuring units(probes), such as "mysegs" below
mysegs<-segexample[,c(1,5:12)]
data(annotexample)
segtable<-CNpreprocessing(segall=mysegs,ratall=ratexample,"ID",chromcol="chrom",
bpstartcol="chrom.pos.start",bpendcol="chrom.pos.end",annot=annotexample,
annotstartcol="CHROM.POS",annotendcol="CHROM.POS",annotchromcol="CHROM",
blsize=50,minjoin=0.25,cweight=0.4,bstimes=50,chromrange=1:22,distrib="Rparallel",
njobs=40,modelNames="E",normalength=normsegs[,1],normalmedian=normsegs[,2])

## End(Not run)
```

makeCNPmask *Given a set of copy-number events, create a DNA copy-number mask*

Description

The function takes as an input a set of intervals with integer-valued boundary positions. It then finds interval regions where the event count is above each of two thresholds, upper and lower, and returns those interval regions with the count above the lower threshold that contain interval regions with the count above the upper threshold.

Usage

```
makeCNPmask(imat,chromcol=1,startcol=2,endcol=3,nprof=1,uthresh,dthresh)
```

Arguments

imat	A matrix or a data frame tabulating the chromosome number and endpoint positions of the interval events.
chromcol, startcol, endcol	Character strings or integers specifying the columns of imat containing the chromosome number and the left (right) endpoint positions of the interval events.
nprof	An integer specifying the number of copy number profiles from which the events originate.
uthresh, dthresh	Numeric, specifying the upper and lower thresholds for the event frequency or (if nprof =1) for the event count.

Value

An integer matrix with three columns, called "chrom","start" and "end", specifying the chromosome number and boundary positions of the mask.

Author(s)

Alex Krasnitz, Guoli Sun

Examples

```
#load a table of copy number events collected from 1203 profiles.
data(cnpexample)
#Create a table of gain (amplification) events only.
amps<-cnpexample[cnpexample[,"copy.num"]=="amp",]
#Create a mask using this table.
ampCNPmask<-makeCNPmask(imat=amps,chromcol="chrom",startcol="chrom.start",
endcol="chrom.end",nprof=1203,uthresh=0.02,dthresh=0.008)
```

normsegs *A reference set of segments*

Description

A table of segment lengths and log copy number ratios for a large set of human diploid genomes.

Usage

```
data(normsegs)
```

Format

A data matrix with 43497 rows/segments and 2 columns/variables.

`length` a numeric vector of segment genomic length

`segmedian` a numeric vector of segment median computed from log copy number ratio

Details

The table originates in a set of copy number profiles of over a 1000 individuals, obtained using Representational Oligonucleotide Microarray Analysis (ROMA) technology. To ensure ploidy of 2 segments from X and Y chromosomes and segemnts shorter than 5Mb were excluded.

Source

Science. 2007 Apr 20;316(5823):445-9. Epub 2007 Mar 15.

Strong association of de novo copy number mutations with autism.

Sebat J, Lakshmi B, Malhotra D, Troge J, Lese-Martin C, Walsh T, Yamrom B, Yoon S, Krasnitz A, Kendall J, Leotta A, Pai D, Zhang R, Lee YH, Hicks J, Spence SJ, Lee AT, Puura K, Lehtimaki T, Ledbetter D, Gregersen PK, Bregman J, Sutcliffe JS, Jobanputra V, Chung W, Warburton D, King MC, Skuse D, Geschwind DH, Gilliam TC, Ye K, Wigler M.

Examples

```
data(normsegs)
```

`ratexample` *Example of copy number log ratio data*

Description

Log ratio data for 5 breast cancer genomes, derived using Representational Oligonucleotide Microarray Analysis (ROMA), an array-based hybridization method that uses genomic complexity reduction based on representations.

Usage

```
data(ratexample)
```

Format

A log ratio data matrix with rows of 83055 oligonucleotide probes, and columns of 5 breast tumors.

Details

The values are natural log copy number ratios, consistent with data in `segexample` (segmented data for these tumors) and `normsegs`. These copy number ratios are normalized using an intensity-based lowess curve fitting algorithm.

Source

Genome Res. 2006 Dec;16(12):1465-79.

Novel patterns of genome rearrangement and their association with survival in breast cancer.

Hicks J, Krasnitz A, Lakshmi B, Navin NE, Riggs M, Leibu E, Esposito D, Alexander J, Troge J, Grubor V, Yoon S, Wigler M, Ye K, Borresen-Dale AL, Naume B, Schlicting E, Norton L, Hagerstrom T, Skoog L, Auer G, Maner S, Lundin P, Zetterberg A.

Examples

```
data(ratexample)
#Plot the whole genome log ratio data for the first profile
#Note X and Y chromosomes at the far  right of the plot
plot(ratexample[,1])
```

segexample *Example of a segmented copy number table.*

Description

Segmented log ratio data for 5 breast cancer genomes, derived using Representational Oligonucleotide Microarray Analysis (ROMA) platform. ROMA detects genomic amplifications and deletions with boundaries defined at a resolution of 50 kb. In this segmented table, each row represnts a segment.

Usage

```
data(segexample)
```

Format

A data frame with 479 rows/segments and 12 columns/variables.

`ID` a character vector of profile IDs

`start` a numeric vector (segment start probe number)

`end` a numeric vector (segment end probe number)

`num.probes` a numeric vector (number of probes in the segment)

`seg.median` a numeric vector (median log ratio)

`chrom` a numeric vector (chromosome number)

`chrom.pos.start` a numeric vector (genomic start)

`chrom.pos.end` a numeric vector (genomic end)

`cytoband.start` a character vector (cytogenetic band start)

`cytoband.end` a character vector (cytogenetic band end)

`abs.pos.start` a numeric vector (genomic start, absolute)

`abs.pos.end` a numeric vector (genomic end, absolute)

Details

Segment medians are computed from log copy number ratio. The corresponding raw data table is `ratexample` in this package.

Source

Genome Res. 2006 Dec;16(12):1465-79.

Novel patterns of genome rearrangement and their association with survival in breast cancer.

Hicks J, Krasnitz A, Lakshmi B, Navin NE, Riggs M, Leibu E, Esposito D, Alexander J, Troge J, Grubor V, Yoon S, Wigler M, Ye K, Borresen-Dale AL, Naume B, Schlicting E, Norton L, Hagerstrom T, Skoog L, Auer G, Maner S, Lundin P, Zetterberg A.

Examples

```
data(segexample)
```

Index