Package ‘biomvRCNS’

October 10, 2024

Type Package

Title Copy Number study and Segmentation for multivariate biological data

Version 1.45.0

Date 2021-11-20

Author Yang Du

Maintainer Yang Du <tooyoung@gmail.com>

Description In this package, a Hidden Semi Markov Model (HSMM) and one homogeneous segmentation model are designed and implemented for segmentation genomic data, with the aim of assisting in transcripts detection using high throughput technology like RNA-seq or tiling array, and copy number analysis using aCGH or sequencing.

License GPL (>= 2)

LazyLoad yes

Imports methods, mvtnorm

Depends IRanges, GenomicRanges, Gviz

Suggests cluster, parallel, GenomicFeatures, dynamicTreeCut, Rsamtools, TxDb.Hsapiens.UCSC.hg19.knownGene

biocViews aCGH, CopyNumberVariation, Microarray, Sequencing, Visualization, Genetics

git_url https://git.bioconductor.org/packages/biomvRCNS

git_branch devel

git_last_commit 03672c7

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-10-10

Contents

biomvRCNS-class 2
biomvRGviz 3
biomvRhsmm 4
biomvRmgmr 6

biomvRseg 8
coriell 10
encodeTP53 10
hsmmRun 11
maxGapminRun 12
regionSegAlphaNB 14
regionSegCost 15
simSegData 16
sojournAnno 17
splitFarNeighbour 18
variosm 19

Index **20**

`biomvRCNS-class` *Class* "biomvRCNS"

Description

The default object class returned by `biomvRhsmm`, `biomvRseg` and `biomvRmgmr`

Objects from the Class

Objects can be created by calls of the form `new("biomvRCNS", ...)`.

Slots

`x`: Object of class "GRanges", with range information either from real positional data or just indices, with input data matrix stored in the meta columns. Additional meta columns for the estimated states and associated probabilities for each sample will also be appended following the input data matrix when using `biomvRhsmm`.

`res`: Object of class "GRanges" , each range represent one continuous segment identified, with sample name slot 'SAMPLE' and segment mean slot 'MEAN' stored in the meta columns

`param`: Object of class "list", list of all parameters used in the corresponding model.

Methods

plot `signature(x = "biomvRCNS", y = "ANY")`: ...

show `signature(object = "biomvRCNS")`: ...

Examples

```
showClass("biomvRCNS")
```

biomvRGviz *Plot segmentation result using Gviz*

Description

This function could be called to plot segmentation output, together with the input signal and optional annotation. By default resulting image will be printed to file. The plot method for class biomvRCNS-class also calls this method. See the vignette for a more complete example.

Usage

```
biomvRGviz(exprgr, gmgr = NULL, prange = NULL, regionID = NULL, seggr = NULL, plotstrand = "+", eps =
```

Arguments

exprgr	a GRanges object with one numeric column for the segmentation input signal in its meta DataFrame
gmgr	an optional GRanges object for the annotation, which must have one column named 'TYPE' in its meta DataFrame
prange	an optional vector defining the scope of the plot, the first 3 elements must be formatted as c('seqname', 'start_position', 'end_position')
regionID	a character for the name of the plotted region or gene name or other identifier, will be used in the title of the plot and the output file name
seggr	a GRanges object for the segmentation output, which must have one column named 'STATE' in its meta DataFrame
plotstrand	select which strand to plot, possible values are '+', '-', '*'
eps	whether to output EPS file using postscript, if FALSE then PDF files for each sequence will be generated to the current working folder.
tofile	whether to output graphics file, if FALSE then will plot on the active device and have the trackList returned.
...	other arguments for plot, like main, ylab, cex, or height and width for graphic device.

Details

See the vignette for more details and examples.

Value

Plot graph on the active device or output to EPS/PDF file.

Author(s)

Yang Du

Examples

```
data(coriell)
x<-coriell[coriell[,2]==1,]
xgr<-GRanges(seqnames=paste('chr', x[,2], sep=''), IRanges(start=x[,3], width=1, names=x[,1]))
values(xgr)<-DataFrame(x[,4:5], row.names=NULL)
xgr<-xgr[order(xgr)]

J<-2; maxk<-50
# a uniform inital sojourn, not utilizing positional information
soj<-list(J=J, maxk=maxk, type='gamma', d=cbind(dunif(1:maxk, 1, maxk), dunif(1:maxk, 1, maxk)))
soj$D <- sapply(1:J, function(j) rev(cumsum(rev(soj$d[1:maxk,j]))))
sample<-colnames(coriell)[5]
runout<-hsmmRun(matrix(values(xgr)[,sample]), sample, xgr, soj, emis=list(type='norm', mu=quantile(unlist(x
biomvRGviz(exprgr=xgr, seggr=runout$res, tofile=FALSE)
```

`biomvRhsmm` *Estimating the most likely state sequence using Hidden Semi Markov Model*

Description

The batch function of building Hidden Semi Markov Model (HSMM) to estimate the most likely state sequences for multiple input data series.

Usage

```
biomvRhsmm(x, maxk=NULL, maxbp=NULL, J=3, xPos=NULL, xRange=NULL, usePos='start', emis.type='norm'
```

Arguments

`x`	input data matrix, or a `GRanges` object with input stored in the meta `DataFrame`, assume ordered.
`maxk`	maximum length of stay for the sojourn distribution
`maxbp`	maximum length of stay in bp for the sojourn distribution, given positional information specified in `xPos` / `xRange`
`J`	number of states
`xPos`	a vector of positions for each `x` row
`xRange`	a `IRanges`/`GRanges` object, same length as `x` rows
`usePos`	character value to indicate whether the 'start', 'end' or 'mid' point position should be used to estimate the sojourn distribution
`emis.type`	type of the emission distribution, only the following types are supported: 'norm', 'mvnorm', 'pois', 'nbinom', 'mvt', 't'
`com.emis`	whether to set a common emission prior across different seqnames. if TRUE, the emission will not be updated during individual runs.
`xAnno`	a optional `TxDb` / `GRanges` / `GRangesList` / `list` object used in `sojournAnno` to infer parameters for the sojourn distribution
`soj.type`	type of the sojourn distribution, only the following types are supported: 'nonpara', 'gamma', 'pois', 'nbinom'

q.alpha	a quantile factor controlling the estimated prior for the mean of the emission of each states, seq(from=q.alpha, to=1-q.alpha, length.out=J)
r.var	a ratio factor controlling the estimated prior for the variance / covariance structure of each states. A value larger than 1 tend to allow larger variation in extreme states; a value smaller than 1 will decrease the probability of having extreme state
useMC	TRUE if mclapply should be used to speed up the calculation, use options(mc.cores=n) to set number of parallel processes
cMethod	C algorithm used for the most likely state sequence, 'F-B' or 'Viterbi'
maxit	max iteration of the EM run with Forward-Backward algorithm
maxgap	max distance between neighbouring feature to consider a split
tol	tolerance level of the likelihood change to terminate the EM run
grp	vector of group assignment for each sample, with a length the same as columns in the data matrix, samples within each group would be processed simultaneously if a multivariate emission distribution is available
cluster.m	clustering method for prior grouping, possible values are 'ward','single','complete','average','mcquitt
avg.m	method to calculate average value for each segment, 'median' or 'mean' possibly trimmed
prior.m	method to select emission prior for each state, 'quantile' uses different levels of quantile; the 'cluster' method uses clara function from cluster
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
na.rm	TRUE if NA value should be ignored

Details

This is the batch function of building Hidden Semi Markov Model (HSMM) to estimating the most likely state sequences for multiple input data series. The function will sequentially process each region identified by the distinctive seqnames in x or in xRange if available, or assuming all data from the same region. A second layer of stratification is introduced by the argument grp, which could be used to reflect experimental design. The assumption is that profiles from the same group could be considered homogeneous, thus processed together if emis.type is compatible (currently only with 'mvnorm'). Argument for the sojourn density will be initialized as flat prior or estimated from other data before calling the work horse function hsmmRun. Then for each batch run results will be combined and eventually a biomvRCNS-class object will be returned. See the vignette for more details and examples.

Value

A biomvRCNS-class object:

x:	Object of class "GRanges", with range information either from real positional data or just indices, with input data matrix stored in the meta columns. Additional meta columns for the estimated states and associated probabilities for each sample or group will also be appended following the input data matrix.
res:	Object of class "GRanges" , each range represent one continuous segment identified, with sample name slot 'SAMPLE', estimated state slot 'STATE' and segment mean slot 'MEAN' stored in the meta columns
param:	Object of class "list", list of all parameters used in the model run, plus the re-estimated emission and sojourn parameters.

Author(s)

Yang Du

References

Guedon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. Journal of Computational and Graphical Statistics, 12(3), 604-639.

See Also

biomvRseg

Examples

```
data(coriell)
xgr<-GRanges(seqnames=paste('chr', coriell[,2], sep=''), IRanges(start=coriell[,3], width=1, names=coriell[
values(xgr)<-DataFrame(coriell[,4:5], row.names=NULL)
xgr<-sort(xgr)
reshsmm<-biomvRhsmm(x=xgr, maxbp=4E4, J=3, soj.type='gamma', emis.type='norm', grp=c(1,2))

## access model parameters
reshsmm@param$soj.par
reshsmm@param$emis.par

## states assigned and associated probabilities
mcols(reshsmm@x)[,-(1:2)]
```

---

biomvRmgmr — *Batch process multiple sequences and samples using max-gap-min-run algorithm for 2 states segmentation*

Description

This is a wrapper function for batch processing multiple sequences and samples using max-gap-min-run algorithm for 2 states segmentation

Usage

```
biomvRmgmr(x, xPos=NULL, xRange=NULL, usePos='start', cutoff=NULL, q=0.9, high=TRUE, minrun=5, maxg
```

Arguments

x	input data matrix, or a GRanges object with input stored in the meta DataFrame, assume ordered.
xPos	a vector of positions for each x row
xRange	a IRanges/GRanges obejct, same length as x rows
usePos	character value to indicate whether the 'start', 'end' or 'mid' point position should be used
cutoff	threshold level above which is considered extreme
q	relative quantile threshold level instead of absolute value for the cutoff

high	TRUE if the cutoff or q here is the lower bound and values greater than the threshold are considered
minrun	minimum run length for the resulting segments
maxgap	maximum genomic distance below which two adjacent qualified tiles can be joined
splitLen	numeric value, maximum length of segments, split if too long
poolGrp	TRUE if samples within the same group should be pooled using median for each feature
grp	vector of group assignment for each sample, with a length the same as columns in the data matrix, samples within each group would be processed simultaneously if a multivariate emission distribution is available
cluster.m	clustering method for prior grouping, possible values are 'ward','single','complete','average','mcquitt
avg.m	method to calculate average value for each segment, 'median' or 'mean' possibly trimmed
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
na.rm	TRUE if NA value should be ignored

Details

This is the batch function to apply maxGapminRun multiple sequence.

Value

A biomvRCNS-class object:

x:	Object of class "GRanges", with range information either from real positional data or just indices, with input data matrix stored in the meta columns.
res:	Object of class "GRanges" , each range represent one continuous segment identified, with sample name slot 'SAMPLE' and segment mean slot 'MEAN' stored in the meta columns
param:	Object of class "list", list of all parameters used in the model run.

Author(s)

Yang Du

See Also

biomvRhsmm maxGapminRun

Examples

```
data(coriell)
xgr<-GRanges(seqnames=paste('chr', coriell[,2], sep=''), IRanges(start=coriell[,3], width=1, names=coriell[
values(xgr)<-DataFrame(coriell[,4:5], row.names=NULL)
xgr<-xgr[order(xgr)]
resseg<-biomvRmgmr(x=xgr, minrun=3000, maxgap=1500, q=0.9, grp=c(1,2))
```

biomvRseg *Homogeneous segmentation of multi-sample genomic data*

Description

The function will perform a two stage segmentation on multi-sample genomic data from array experiment or high throughput sequencing data.

Usage

```
biomvRseg(x, maxk=NULL, maxbp=NULL, maxseg=NULL, xPos=NULL, xRange=NULL, usePos='start', family='n
```

Arguments

x	input data matrix, or a GRanges object with input stored in the meta DataFrame
maxk	maximum length of a segment
maxbp	maximum length of a segment in bp, given positional information specified in xPos / xRange / or x
maxseg	maximum number of segment the function will try
xPos	a vector of positions for each x row
xRange	a IRanges/GRanges object, same length as x rows
usePos	character value to indicate whether the 'start', 'end' or 'mid' point position should be used
family	family of x distribution, only the following types are supported: 'norm', 'nbinom', 'pois'
penalty	penalty method used for determining the optimal number of segment using likelihood, possible values are 'none','AIC','AICc','BIC','SIC','HQIC', 'mBIC'
twoStep	TRUE if a second stage merging will be performed after the initial group segmentation
segDisp	TRUE if a segment-wise estimation of dispersion parameter rather than using a overall estimation
useMC	TRUE if mclapply should be used to speed up the calculation for nbinom dispersion estimation
useSum	TRUE if using grand sum across sample / x columns, like in the tilingArray solution
comVar	TRUE if assuming common variance across samples (x columns)
maxgap	max distance between neighbouring feature to consider a split
tol	tolerance level of the likelihood change to determining the termination of the EM run
grp	vector of group assignment for each sample, with a length the same as columns in the data matrix, samples within each group would be processed simultaneously if a multivariate emission distribution is available
cluster.m	clustering method for prior grouping, possible values are 'ward','single','complete','average','mcquitt
avg.m	method to calculate average value for each segment, 'median' or 'mean' possibly trimmed

`trim`	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
`na.rm`	`TRUE` if `NA` value should be ignored

Details

A homogeneous segmentation algorithm, using dynamic programming like in `tilingArray`; however capable of handling count data from sequencing.

Value

A `biomvRCNS-class` object:

`x:`	Object of class `"GRanges"`, with range information either from real positional data or just indices, with input data matrix stored in the meta columns.
`res:`	Object of class `"GRanges"` , each range represent one continuous segment identified, with sample name slot 'SAMPLE' and segment mean slot 'MEAN' stored in the meta columns
`param:`	Object of class `"list"`, list of all parameters used in the model run.

References

Piegorsch, W. W. (1990). Maximum likelihood estimation for the negative binomial dispersion parameter. Biometrics, 863-867.

Picard,F. et al. (2005) A statistical approach for array CGH data analysis. BMC Bioinformatics, 6, 27.

Huber,W. et al. (2006) Transcript mapping with high density oligonucleotide tiling arrays. Bioinformatics, 22, 1963-1970. .

Zhang, N. R. and Siegmund, D. O. (2007). A Modified Bayes Information Criterion with Applications to the Analysis of Comparative Genomic Hybridization Data. Biometrics 63 22-32.

Robinson MD and Smyth GK (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics, 9, 321-332

See Also

`biomvRhsmm`

Examples

```
data(coriell)
xgr<-GRanges(seqnames=paste('chr', coriell[,2], sep=''), IRanges(start=coriell[,3], width=1, names=coriell[
values(xgr)<-DataFrame(coriell[,4:5], row.names=NULL)
xgr<-xgr[order(xgr)]
resseg<-biomvRseg(x=xgr, maxbp=4E4, maxseg=10, family='norm', grp=c(1,2))
```

coriell — *Array CGH data set of Coriell cell lines*

Description

These are two data array CGH studies sets of Corriel cell lines taken from the reference below.

Format

A data frame containing five variables: first is clone name, second is clone chromosome, third is clone position, fourth and fifth are log2ratio for two cell lines.

References

http://www.nature.com/ng/journal/v29/n3/suppinfo/ng754_S1.html

Snijders et al., Assembly of microarrays for genome-wide measurement of DNA copy number, Nature Genetics, 2001

encodeTP53 — *mapped RNA-seq data from ENCODE*

Description

The data contains gene expression and transcript annotations in the region of the human TP53 gene (region (chr17:7,560,001-7,610,000 from the Human February 2009 (GRCh37/hg19) genome assembly), which is part of the long RNA-seq data generated by ENCODE/Cold Spring Harbor Lab, containing 2 cell types (GM12878 and K562) with 2 replicates each.

The alignment files were pulled from UCSC (http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeC
And subsequently reads were counted in each non-overlapping 25bp window for the region (chr17:7,560,001-7,610,000). The example code to generate this count `GRanges` is available in the vignette.

The regional annotation of TP53 RNAs isoforms were derived from the ENCODE Gene Annotations (GENCODE), sub-setted to only isoforms of TP53 gene. http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeGencodeV4/wgEncodeGencodeManualV4.gtf.gz).

This dataset is used in the package vignette to illustrate a use case of transcript detection.

Format

Containing two `GRanges` objects, one for the sample count and one for the regional annotation of gene TP53

References

http://dx.doi.org/10.1371%2Fjournal.pbio.1001046 The ENCODE Project Consortium (2011) A User's Guide to the Encyclopedia of DNA Elements (ENCODE). PLoS Biol 9(4): e1001046. doi:10.1371/journal.pbio.1001046

hsmmRun *Estimating the most likely state sequence using Hidden Semi Markov Model*

Description

This is the working horse of the `biomvRhsmm`

Usage

```
hsmmRun(x, xid="sampleid", xRange, soj, emis, cMethod='F-B', maxit=1, maxgap=Inf, tol=1e-06, avg.m=
```

Arguments

`x`	input data matrix or vector, ordered wrt. position
`xid`	name of the sample
`xRange`	a `IRanges`/`GRanges` object, same length as `x` rows
`soj`	a list object containing the relevant sojourn distribution parameters
`emis`	a list object containing the relevant emission distribution parameters
`cMethod`	C algorithm used for the most likely state sequence, 'F-B' or 'Viterbi'
`maxit`	max iteration of the EM run with Forward-Backward algorithm
`maxgap`	max distance between neighbouring feature to consider a split
`tol`	tolerance level of the likelihood change to terminate the EM run
`avg.m`	method to calculate average value for each segment, 'median' or 'mean' possibly trimmed
`trim`	the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
`na.rm`	`TRUE` if `NA` value should be ignored
`com.emis`	whether to set a common emission prior across different seqnames. if TRUE, the emission will not be updated during individual runs.

Details

The function fits a Hidden-semi Markov model for the input data matrix / vector, which should contains ordered data from a continuous region on one chromosome. The model will start with flat prior for the initial state probability and transition probability, while emission parameter for each state will be estimated using different quantiles of the input controlled by argument `q.alpha` and `r.var`. Argument for the sojourn density should be provided via the list object `soj`, which is either initialized as flat prior or estimated from other data in a previous call. The positional information in the `xRange` is used for the optional spiting of physically distant features and construction of returning `GRanges` object `res`.

Value

a list object,

yhat	a "Rle" object, the most likely state sequence, same length as x rows number
yp	"Rle", the associated state probability, same length as x rows number
res	Object of class "GRanges" , each range represent one continuous segment identified, with sample name slot 'SAMPLE', estimated state slot 'STATE' and segment mean slot 'MEAN' stored in the meta columns

Author(s)

Yang Du

References

Guedon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. Journal of Computational and Graphical Statistics, 12(3), 604-639.

See Also

biomvRhsmm

Examples

```
data(coriell)
# select only chr1
x<-coriell[coriell[,2]==1,]
xgr<-GRanges(seqnames=paste('chr', x[,2], sep=''), IRanges(start=x[,3], width=1, names=x[,1]))
values(xgr)<-DataFrame(x[,4:5], row.names=NULL)
xgr<-xgr[order(xgr)]

J<-2 ; maxk<-50
# a uniform initial sojourn, not utilizing positional information, just the index
soj<-list(J=J, maxk=maxk, type='gamma', d=cbind(dunif(1:maxk, 1, maxk), dunif(1:maxk, 1, maxk)))
soj$D <- sapply(1:J, function(j) rev(cumsum(rev(soj$d[1:maxk,j]))))
# run 1 sample only, Coriell.13330
sample<-colnames(coriell)[5]
runout<-hsmmRun(matrix(values(xgr)[,sample]), sample, xgr, soj, emis=list(type='norm', mu=range(x[,4:5]), va
```

---

maxGapminRun — *Max-gap-min-run algorithm for 2 states segmentation*

Description

A custom Max-gap-min-run implementation using physical position for gap and run length calculation.

Usage

```
maxGapminRun(x, xPos = NULL, xRange = NULL, cutoff = NULL, q = 0.9, high=TRUE, minrun = 5, maxgap = 2,
```

Arguments

x	a numeric vector for the input signal
xPos	a numeric vector, same length as x, carrying positional information for each element of x
xRange	an IRanges object, same length as x, carrying range information for each element of x
cutoff	numeric value used as cut-off, optional if q is specified
q	numeric value used to derive cut-off of x, as the q quantile of x , optional if cutoff is specified
high	TRUE if the cutoff or q here is the lower bound and values greater than the threshold are considered
minrun	minimum run length for the resulting segments
maxgap	maximum genomic distance below which two adjacent qualified tiles can be joined
splitLen	numeric value, maximum length of segments, split if too long
na.rm	TRUE if NA value should be ignored

Details

A custom Max-gap-min-run implementation using physical position for gap and run length calculation.

Value

a list of segment starts and ends indices

IS	the start index for each segment
IE	the end index for each segment
CUTOFF	the cutoff value used in the run
MG	the parameter value for maxgap
MR	the parameter value for minrun
SPL	the parameter value for splitLen

Author(s)

Yang Du

See Also

biomvRhsmm biomvRseg biomvRmgmr

Examples

```
x<-rpois(50, 10)
xpos<-rnorm(50, 300, 100)
xpos<-xpos[order(xpos)]
maxGapminRun(x, xpos, cutoff=9.5, maxgap=30, minrun=100)
```

`regionSegAlphaNB` *Estimate matrix of dispersion parameter alpha (size) used in* `regionSegCost` *for negative binomial distributed* `x`.

Description

Estimate matrix of dispersion parameter alpha (size) used in `regionSegCost` for negative binomial distributed `x`.

Usage

```
regionSegAlphaNB(x, maxk = NULL, segs = NULL, useMC = FALSE, tol=1e-06)
```

Arguments

`x`	The input data matrix or vector
`maxk`	Maximum number of index to search forward
`segs`	Starting indices (excluding 1) for the candidate segments, for the second stage model, `maxk` will be overridden with `length(segs)+1`.
`useMC`	TRUE if `mclapply` should be used to speed up the calculation
`tol`	tolerance level for the convergence criteria in the maximum likelihood estimation of negative binomial distribution dispersion parameter.

Details

Estimate matrix of dispersion parameter alpha (size) used in `regionSegCost` for negative binomial distributed `x`.

Value

Matrix with `maxk` rows and `nrow(x)` columns, or a `length(segs)+1` square matrix for the second stage model.

References

Piegorsch, W. W. (1990). Maximum likelihood estimation for the negative binomial dispersion parameter. Biometrics, 863-867.

Robinson MD and Smyth GK (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics, 9, 321-332

See Also

`regionSegCost`

Examples

```
x<-matrix(rnbinom(120, size=0.05, mu=20), ncol=3)
Aa<-regionSegAlphaNB(x, maxk=20)
dim(Aa) # [1] 20 40
Ab<-regionSegAlphaNB(x, segs=as.integer(c(3, 6, 12, 30)))
dim(Ab) # [1] 5 5
```

`regionSegCost` *Regional segmentation cost matrix*

Description

To calculate regional cost matrix for the initial stage and second merging stage of the segmentation model.

Usage

```
regionSegCost(x, maxk = NULL, segs = NULL, family = NULL, alpha = NULL, useSum = TRUE, useMC = FALSE,
```

Arguments

`x`	The input data matrix or vector
`maxk`	Maximum number of index to search forward
`segs`	Starting indices (excluding 1) for the candidate segments, for the second stage model, `maxk` will be overridden with `length(segs)+1`.
`family`	which exponential family the data belongs to, possible values are 'norm', 'pois' and 'nbinom'
`alpha`	alpha matrix for negative binomial cost calculation, estimated from `regionSegAlphaNB`
`useSum`	TRUE if using grand sum across sample / x columns, like in the `tilingArray` solution
`useMC`	TRUE if `mclapply` should be used to speed up
`comVar`	TRUE if assuming common variance across samples (x columns)

Details

Preparing the cost matrix for the follow-up segmentation. Using residual sum of squares for 'norm' data, and negative log-likelihood for 'pois' and 'nbinom' data. Extension of the `costMatrix` function in `tilingArray`.

Value

Matrix with `maxk` rows and `nrow(x)` columns, or a `length(segs)+1` square matrix for the second stage model.

References

Piegorsch, W. W. (1990). Maximum likelihood estimation for the negative binomial dispersion parameter. Biometrics, 863-867.

Picard,F. et al. (2005) A statistical approach for array CGH data analysis. BMC Bioinformatics, 6, 27.

Huber,W. et al. (2006) Transcript mapping with high density oligonucleotide tiling arrays. Bioinformatics, 22, 1963-1970. .

Robinson MD and Smyth GK (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics, 9, 321-332

See Also

regionSegAlphaNB

Examples

```
x<-matrix(rnorm(120), ncol=3)
Ca<-regionSegCost(x, maxk=20, family='norm')
dim(Ca) # [1] 20 40
Cb<-regionSegCost(x, segs=as.integer(c(3, 6, 12, 30)), family='norm')
dim(Cb) # [1] 5 5
```

---

simSegData *Simulate exemplary segmentation data.*

Description

Simulate exemplary segmentation data.

Usage

```
simSegData(nseg=10, J=3, soj, emis, seed=1234, toPlot=FALSE)
```

Arguments

nseg	size of initial segments pool
J	states number
soj	a list object containing sojourn settings
emis	a list object containing emission settings
seed	seed for simulation
toPlot	whether to output a pdf image of the simulated series

Value

a list object containing the simulated data and the segment info

E	a numeric vector of the simulated data serie
L	a vector of the length for each continuous segment
S	a vector of state assignment for each segment
pdf	the name of the output pdf file if any

Examples

```
soj<-list(type='pois', lambda=c(200, 100, 10))
emis<-list(type='pois', lambda=1:3)
simSegData(soj=soj, emis=emis)
```

`sojournAnno` *Estimate sojourn distribution parameters from posterior information like annotation data*

Description

Using prior information from previous studies or annotation data to determine sojourn distribution parameters

Usage

```
sojournAnno(xAnno, soj.type = "gamma", pbdist = NULL)
```

Arguments

`xAnno`	a `GRanges` / `GRangesList` / `TxDb` object, with its first meta column to represent the possible type of the range; Or a `list` object with named initial value vectors matching required parameters for a specific `soj.type`
`soj.type`	type of the sojourn distribution, following types are supported: 'gamma', 'pois', 'nbinom'
`pbdist`	average distance between neighbouring features, in this case in the `link{biomvRhsmm}` call one should only use the rank rather than the position.

Details

Be default, the hidden-semi Markov model implemented in this package uses a uniform prior for the initial sojourn distribution. However user can provide custom data from related studies to learn the prior of the sojourn distribution. The number of possible state will also be estimated from the unique level of feature type in the first meta column of `xAnno` if it is not a `TxDb` object.

Value

a list object containing the sojourn distribution parameter

`type`	type of the sojourn distribution
`fttypes`	unique levels of the types stored in the first meta column of `xAnno`, alphabetically sorted
`J`	number of possible states
`\code{...}`	distribution parameters, 'lambda' and 'shift' for 'pois'; 'size', 'mu' and 'shift' for 'nbinom'; 'scale' and 'shape' for 'gamma'

Author(s)

Yang Du

Examples

```
data(encodeTP53)
encodeTP53$gmgr # a GRanges object
soj<-sojournAnno(encodeTP53$gmgr, soj.type='gamma')
```

`splitFarNeighbour` *Split segments if long gaps exist between feature positions*

Description

Split segments if long gaps exist between feature positions, due to low coverage or resolution.

Usage

```
splitFarNeighbour(intStart = NULL, intEnd = NULL, xPos = NULL, xRange = NULL, maxgap = Inf, minrun = 1
```

Arguments

`intStart`	indices of start for each segment
`intEnd`	indices of end for each segment
`xPos`	position vector, the distance of neighbouring features will be counted as point to point
`xRange`	IRanges / GRanges object for the positions, the the distance of neighbouring features will be counted as end to start.
`maxgap`	maximum distance between neighbouring features
`minrun`	when splitting, the minimum length of the features spanning, which half will be ignored if shorter.

Value

a list object containing the start and end indices for new segments

`IS`	the start indices for new segments
`IE`	the end indices for new segments

Author(s)

Yang Du

Examples

```
set.seed(123)
pos<-cumsum(rnbinom(20, size=10, prob=0.01))
splitFarNeighbour(intStart=c(1, 10), intEnd=c(6, 18), xPos=pos, maxgap=1000)
```

variosm *Differential methylation data from sequencing*

Description

Extracted from package BiSeq, which is a small subset of a published study using targeted bisulfite sequencing data to detect differentially methylated regions (DMRs).

Format

Containing one GRanges object

References

http://dx.doi.org/10.1182 Schoofs et al. DNA methylation changes are a late event in acute upromyelocytic leukemia and coincide with loss of transcription factor binding. Blood, Nov 2012.

Index

∗ **classes**
biomvRCNS-class, 2
∗ **datasets**
coriell, 10
encodeTP53, 10
variosm, 19
∗ **data**
coriell, 10
encodeTP53, 10
variosm, 19
∗ **hsmm**
biomvRhsmm, 4
hsmmRun, 11

biomvRCNS-class, 2
biomvRGviz, 3
biomvRhsmm, *2*, 4, *7*, *9*, *12*, *13*
biomvRmgmr, *2*, 6, *13*
biomvRseg, *2*, *6*, 8, *13*

coriell, 10

encodeTP53, 10

hsmmRun, 11

maxGapminRun, *7*, 12
mclapply, *5*, *8*, *14*, *15*

plot,biomvRCNS,ANY-method
(biomvRCNS-class), 2

regionSegAlphaNB, 14, *15*, *16*
regionSegCost, *14*, 15

show,biomvRCNS-method
(biomvRCNS-class), 2
simSegData, 16
sojournAnno, *4*, 17
splitFarNeighbour, 18

variosm, 19