Package

March 2, 2019

Type Package

Title Identify large-scale CNV events from single cell or bulk RNA-Seq data

Version 0.1.0

Author Akdes Serin Harmanci, Arif O. Harmanci

Maintainer Akdes Serin Harmanci <akdes.harmanci@uth.tmc.edu>

Description Identification, visualization and integrative analysis of CNV events in multiscale resolution using single-cell or bulk RNA sequencing data

Encoding UTF-8

LazyData true

LinkingTo Rcpp

Depends Rcpp, signal, pheatmap, RColorBrewer, HMMcopy, IRanges, grid, GenomeGraphs, ggplot2, reshape, mclust, ggpubr, scales, gridExtra, igraph, intergraph, ggnetwork, philentropy, ape, biomaRt, limma, GO.db, org.Hs.eg.db, GOstats

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`CaSpER-package` *CaSpER: Identify large-scale CNV events from single cell or bulk RNA-Seq data*

Description

Identification, visualization and integrative analysis of CNV events in multiscale resolution using single-cell or bulk RNA sequencing data

Details

The main functions you will need to use are CreateCasperObject() and runCaSpER(casper_object). For additional details on running the analysis step by step, please refer to the example vignette.

`assignStates` *assignStates()*

Description

calculates baf shift threshold using gaussian mixture models and assigns deletion or amplification to a segment when the HMM state is 1 or 5 without looking at the BAF signal. When the segment state is 2 or 4, an accompanying BAF shift on the segment is required.

Usage

```
assignStates(object)
```

Arguments

`object`	casper object

Value

object

`AverageReference` *AverageReference()*

Description

the mean the expression level for each gene across all the reference cells (samples) are computed.

Usage

```
AverageReference(data, ref_ids)
```

Arguments

`object`	casper object

Value

object

calcROC *calcROC()*

Description

Calculates tpr and fpr values using genotyping array as gold standard

Usage

```
calcROC(chrMat, chrMat2)
```

Arguments

`chrMat`	large scale event matrix generated using CaSpER
`chrMat2`	large scale event matrix generated using genotyping array

Value

accuracy measures

calculateLOHShiftsForEachSegment *calculateLOHShiftsForEachSegment()*

Description

calculate the median value of the BAF shift signal on the segments

Usage

```
calculateLOHShiftsForEachSegment(object)
```

Arguments

`object`	casper object

Value

object

`casper` *The CaSpER Class*

Description

The CaSpER Class The casper object is required for performing CNV analysis on single-cell and bulk RNA-Seq. It stores all information associated with the dataset, including data, smoothed data, baf values, annotations, scale specific segments, scale specific large scale events etc.

Slots

`raw.data` raw project data

`data` lowly expressed genes are filtered from the data

`loh` original baf signal

`median.filtered.data` median filtered expression signal

`loh.median.filtered.data` median filtered baf signal

`centered.data` gene expression levels are centered around the mid-point. For each gene, the mid-point of expression level is computed among all the cells (or samples in bulk RNA-seq), then the mid-point expression level is subtracted from the expression levels

`center.smoothed.data` cell centric expression centering is performed. For each cell (or sample), we compute the mid-point of the expression level then we subtract the mid-point expression from the expression levels of all the genes for the corresponding cel

`control.normalized` control normalization is performed by subtracting reference expression values from the tumor expression values.

`control.normalized.visbound` control normalized data is thresholded in order to perform better visualization.

`control.normalized.visbound.noiseRemoved` noise is removed from control normalized and thresholded data.

`large.scale.cnv.events` large scale CNV events identified by CaSpER

`segments` CNV segments identified by CaSpER

`cytoband` cytoband information downloaded from UCSC hg19: http://hgdownload.cse.ucsc.edu/goldenpath/hg19/databa
hg38:http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz

`annotation` positions of each gene along each chromosome in the genome

`annotation.filt` lowly expressed genes are filtered from gene annotation data.frame

`control.sample.ids` vector containing the reference (normal) cell (sample) names

`project.name` project name

`genomeVersion` genomeVersion: hg19 or hg38

`hmmparam` initial hmm parameters estimated from data

`plotorder` cell (sample) ordering for heatmap plots

`vis.bound` threshold for control normalized data for better visualization

`noise.thr` noise threshold for better visualization

`loh.name.mapping` containing the cell (sample) name and the matching baf signal sample name

`sequencing.type` sequencing type: bulk or single-cell `cnv.scale` maximum expression scale

`loh.scale` maximum baf scale

`loh.shift.thr` baf shift threshold estimated from baf signal using gaussian mixture models

`window.length` window length used for median filtering

`length.iterations` increase in window length at each scale iteration

`CenterSmooth` *CenterSmooth()*

Description

Cell centric expression centering is performed. For each cell (or sample), we compute the mid-point of the expression level then we subtract the mid-point expression from the expression levels of all the genes for the corresponding cell

Usage

```
CenterSmooth(object)
```

Arguments

`object`	casper object

Value

object

`ControlNormalize` *ControlNormalize()*

Description

The control normalization is performed by subtracting reference expression values from the tumor expression values.

Usage

```
ControlNormalize(object, vis.bound, noise.thr)
```

Arguments

`object`	casper object

Value

object

`CreateCasperObject` *CreateCasperObject*

Description

Creation of a casper object.

Usage

```
CreateCasperObject(raw.data, annotation, control.sample.ids, cytoband,
  loh.name.mapping, cnv.scale, loh.scale, method, loh,
  project = "casperProject", sequencing.type, expr.cutoff = 4.5,
  display.progress = TRUE, log.transformed = TRUE,
  centered.threshold = 3, window.length = 50, length.iterations = 50,
  vis.bound = 2, noise.thr = 0.3, genomeVersion = "hg19", ...)
```

Arguments

`raw.data` the matrix of genes (rows) vs. cells (columns) containing the raw counts

`annotation` data.frame containing positions of each gene along each chromosome in the genome

`control.sample.ids`
vector containing the reference (normal) cell (sample) names

`cytoband` cytoband information downloaded from UCSC hg19: http://hgdownload.cse.ucsc.edu/goldenpath/hg1
hg38:http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz

`loh.name.mapping`
contains the cell (sample) name and the matching baf signal sample name

`cnv.scale` maximum expression scale

`loh.scale` maximum baf scale

`method` analysis type: itereative or fixed (default: iterative)

`loh` The original baf signal

`sequencing.type`
sequencing.type sequencing type: bulk or single-cell

`expr.cutoff` expression cutoff for lowly expressed genes

`log.transformed`
indicates if the data log2 transformed or not. (default:TRUE)

`centered.threshold`

`window.length` window length used for median filtering (default: 50)

`length.iterations`
increase in window length at each scale iteration (default: 50)

`vis.bound` threshold for control normalized data for better visualization (default: 2)

`genomeVersion` genomeVersion: hg19 or hg38 (default: hg19)

Value

casper

`extractEvents` *extractEvents()*

Description

formats large scale events as a matrix. Rows represent samples (cells) whereas columns represent chromosome arms (1: amplification, 0: neutral, -1: deletion) helper function for generateLargeScaleEvents()

Usage

```
extractEvents(segments, cytoband, type)
```

Arguments

`cytoband` cytoband information downloaded from UCSC hg19: http://hgdownload.cse.ucsc.edu/goldenpath/hg1
hg38:http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz

`type` event type amp (amplification) or del (deletion)/

`object` casper object

Value

combined large scale events in data.frame

`extractLargeScaleEvents` *extractLargeScaleEvents()*

Description

generates coherent set of large scale CNV events using the pairwise comparison of all scales from BAF and expression signals

Usage

```
extractLargeScaleEvents(final.objects, thr = 0.5)
```

Arguments

`final.objects` casper object

`thr` gamma threshold determining the least number of scales required to support

Value

final large scale event summary reported as a matrix

`extractMUAndCooccurence` *extractMUAndCooccurence()*

Description

calculates significant mutually exclusive and co-occurent events

Usage

```
extractMUAndCooccurence(finalChrMat, loh, loh.name.mapping)
```

Arguments

`finalChrMat`	large scale event matrix generated using CaSpER
`loh`	original baf signal
`loh.name.mapping`	contains the cell (sample) name and the matching baf signal sample name

Value

list of mutually exclusive and co-occurent events

`extractSegmentSummary` *extractSegmentSummary()*

Description

generates coherent set of CNV segments using the pairwise comparison of all scales from BAF and expression signals

Usage

```
extractSegmentSummary(final.objects)
```

Arguments

`final.objects`	list of casper object

Value

list of loss and gain segments identified in all scales

gene.matrix *gene.matrix()*

Description

Gene level CNV events represented as matrix where rows represent samples and columns represent samples

Usage

```
gene.matrix(segment, all.genes, all.samples, genes.ann)
```

Arguments

`segment`	CNV segments
`all.genes`	gene names
`all.samples`	samp names
`genes.ann`	gene symbols within each segments

Value

matrix of gene level CNV events

generateAnnotation *generateAnnotation()*

Description

retrieves gene chromosomal locations from biomart

Usage

```
generateAnnotation(id_type = "ensembl_gene_id", genes, ishg19,
  centromere)
```

Arguments

`id_type`	gene list identifier, ensembl_gene_id or hgnc_symbol
`genes`	list of genes
`ishg19`	boolean values determining the genome version
`centromere`	centromer regions

Value

list of mutually exclusive and co-occurent events

`generateEnrichmentSummary` *generateEnrichmentSummary()*

Description

generate GO Term enrichment summary

Usage

```
generateEnrichmentSummary(results)
```

Arguments

`results`	output of getDiffExprGenes() function

Value

significantly enriched GO Terms

`generateLargeScaleEvents` *generateLargeScaleEvents()*

Description

generates large scale CNV events

Usage

```
generateLargeScaleEvents(object)
```

Arguments

`object`	casper object

Value

object

generateParam *generateParam()*

Description

Initial HMM parameters estimated from the data.

Usage

```
generateParam(object, cnv.scale = 3)
```

Arguments

`object`	casper object
`cnv.scale`	expression.scale for the expression signal

Value

object

getDiffExprGenes *getDiffExprGenes()*

Description

get differentially expressed genes between samples having selected specified CNV events

Usage

```
getDiffExprGenes(final.objects, sampleName, chrs, event.type)
```

Arguments

`final.objects`	list of objects
`sampleName`	sample name
`chrs`	selected chromosomes
`event.type`	cnv event type

Value

differentially expressed genes

goEnrichmentBP *goEnrichmentBP()*

Description

GO Term enrichment

Usage

```
goEnrichmentBP(genes, ontology, universe = character(0), pvalue = 0.05,
  annotation = "org.Hs.eg.db", conditionalSearch = TRUE, genes2)
```

Arguments

`genes`	list of genes
`ontology`	ontology (BP, CC or MF)
`universe`	universe of genes
`pvalue`	pvalue cutoff
`annotation`	ontology annotation default:org.Hs.eg.db

Value

significantly enriched GO Terms

lohCallMedianFilter *lohCallMedianFilter()*

Description

Reads BAFExtract output files

Usage

```
lohCallMedianFilter(object, loh.scale, n = 50, scale.iteration = 50)
```

Arguments

`path`	path for the folder that contains BAFExtract output files

Value

baf signal in data.frame format

`lohCallMedianFilterByChr` *readBAFExtractOutput()*

Description

Reads BAFExtract output files

Usage

```
lohCallMedianFilterByChr(object, loh.scale, n = 50,
  scale.iteration = 50)
```

Arguments

`path`	path for the folder that contains BAFExtract output files

Value

baf signal in data.frame format

`mergeScalesAndGenerateFinalEventSummary` *mergeScalesAndGenerateFinalEventSummary()*

Description

helper function for extractLargeScaleEvents()

Usage

```
mergeScalesAndGenerateFinalEventSummary(final.objects)
```

Arguments

`final.objects`	list of casper objects

Value

list of objects

`PerformMedianFilter` *PerformMedianFilter()*

Description

Recusive iterative median filtering is applied to whole genome

Usage

```
PerformMedianFilter(object, window.length = 50, length.iterations = 50)
```

Arguments

`object`	casper object
`window.length`	window length used for median filtering
`length.iterations`	increase in window length at each scale iteration

Value

object

`PerformMedianFilterByChr` *PerformMedianFilterByChr()*

Description

Recusive iterative median filtering is applied for each chromosome

Usage

```
PerformMedianFilterByChr(object, window.length = 50,
  length.iterations = 50)
```

Arguments

`object`	casper object
`window.length`	window length used for median filtering
`length.iterations`	increase in window length at each scale iteration

Value

object

`PerformSegmentationWithHMM` *PerformSegmentationWithHMM()*

Description

HMM segmentation applied for each scale of expression signal

Usage

```
PerformSegmentationWithHMM(object, cnv.scale, removeCentromere = T,
  cytoband)
```

Arguments

`object` casper object

`cnv.scale` expression signal scale number

`removeCentromere` boolean values determining if centromere regions should be removed from the analysis

`cytoband` cytoband information downloaded from UCSC hg19: http://hgdownload.cse.ucsc.edu/goldenpath/hg1 hg38:http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz

Value

object

`plotBAFAllSamples` *plotBAFAllSamples()*

Description

Visualization of BAF shift signal for all samples together

Usage

```
plotBAFAllSamples(loh, fileName)
```

Arguments

`loh` baf signal, user can either give smoothed baf signal or original baf signal as an input.

`fileName` fileName of the putput image

`plotBAFInSeperatePages` *plotBAFInSeperatePages()*

Description

Visualization of BAF deviation for each sample in separate pages

Usage

```
plotBAFInSeperatePages(loh, folderName)
```

Arguments

`loh`	baf signal, user can either give smoothed baf signal or original baf signal as an input.
`folderName`	folder name for the output images

Value

object

`plotBAFOneSample` *plotBAFOneSample()*

Description

Visualization of BAF shift signal in different scales for one sample

Usage

```
plotBAFOneSample(object, fileName)
```

Arguments

`object`	casper object
`fileName`	fileName of the output image

`plotGEAllSamples` *plotGEAllSamples()*

Description

plot gene expression signal for each sample seperately

Usage

```
plotGEAllSamples(object, fileName = fileName, cnv.scale)
```

Arguments

`object`	casper object
`fileName`	fileName of the putput image
`cnv.scale`	expression.scale for the expression signal

`plotGEAndBAFOneSample` *plotGEAndBAFOneSample()*

Description

Gene expression and BAF signal for one sample in one plot

Usage

```
plotGEAndBAFOneSample(object, cnv.scale, loh.scale, sample, n = 50,
  scale.iteration = 50)
```

Arguments

`object`	casper object
`cnv.scale`	expression.scale for the expression signal
`sample`	sample name
`n`	window length used for median filtering
`length.iterations`	increase in window length at each scale iteration

plotGEAndGT *plotGEAndGT()*

Description

Heatmap plot for large scale event calls identified by CaSpER and genotyping array.

Usage

```
plotGEAndGT(chrMat, genoMat, fileName)
```

Arguments

`chrMat`	large scale events identified from CaSpER represented as matrix. Rows indicates samples (cells) whereas columns indicates chromosome arms
`genoMat`	large scale events identified from genotyping array represented as matrix. Rows indicates samples (cells) whereas columns indicates chromosome arms
`fileName`	fileName of the putput image

plotHeatmap *plotHeatmap()*

Description

Visualization of the genomewide gene expression signal plot at different smoothing scales

Usage

```
plotHeatmap(object, fileName, cnv.scale = 3, cluster_cols = F,
  cluster_rows = T, show_rownames = T, only_soi = T)
```

Arguments

`object`	casper object
`fileName`	fileName of the putput image
`cnv.scale`	expression.scale for the expression signal
`cluster_cols`	boolean values determining if columns should be clustered
`cluster_rows`	boolean values determining if rows should be clustered
`show_rownames`	boolean values determining if rownames should be plotted
`only_soi`	boolean values determining if only samples of interest without control samples should be plotted

plotLargeScaleEvent *plotLargeScaleEvent()*

Description

Visualization of the large-scale CNV events among all the samples/cells

Usage

```
plotLargeScaleEvent(object, fileName)
```

Arguments

`object`	casper object
`fileName`	fileName of the output image

plotLargeScaleEvent2 *plotLargeScaleEvent2()*

Description

Visualization of the large-scale CNV events among all the samples/cells

Usage

```
plotLargeScaleEvent2(chrMat, fileName)
```

Arguments

`chrMat`	large scale events identified from CaSpER represented as matrix. Rows indicates samples (cells) whereas columns indicates chromosome arms
`fileName`	fileName of the output image

plotMUAndCooccurence *plotMUAndCooccurence()*

Description

Visualization of mutually exclusive and co-occuring events

Usage

```
plotMUAndCooccurence(results)
```

Arguments

`results`	output of extractMUAndCooccurence() function

plotSCellCNVTree *plotSCellCNVTree()*

Description

Pyhlogenetic tree-based clustering and visualization of the cells based on the CNV events from single cell RNA-seq Data.

Usage

```
plotSCellCNVTree(finalChrMat, sampleName,
  path = "C:\\Users\\aharmanci\\Downloads\\phylip-3.695\\phylip-3.695\\exe",
  fileName)
```

Arguments

`finalChrMat`	large scale events identified from CaSpER represented as matrix. Rows indicates samples (cells) whereas columns indicates chromosome arms
`sampleName`	sample name
`path`	path to the executable containing fitch. If path = NULL, the R will search several commonly used directories for the correct executable file. More information about installing PHYLIP can be found on the PHYLIP webpage: http://evolution.genetics.washington.edu/phylip.html.

plotSingleCellLargeScaleEventHeatmap *plotSingleCellLargeScaleEventHeatmap()*

Description

Visualization of large scale event summary for selected samples and chromosomes

Usage

```
plotSingleCellLargeScaleEventHeatmap(finalChrMat, sampleName, chrs)
```

Arguments

`finalChrMat`	large scale events identified from CaSpER represented as matrix. Rows indicates samples (cells) whereas columns indicates chromosome arms
`sampleName`	sample name
`chrs`	chromosome names

Value

object

ProcessData *ProcessData()*

Description

Processing expression signal. Step 1. Recursively iterative median filtering Step 2. Center Normalization Step 3. Control Normalization

Usage

```
ProcessData(object)
```

Arguments

`object`	casper object

Value

object

readBAFExtractOutput *readBAFExtractOutput()*

Description

Reads BAFExtract output files

Usage

```
readBAFExtractOutput(path, sequencing.type = "bulk")
```

Arguments

`path`	path for the folder that contains BAFExtract output files

Value

baf signal in data.frame format

`runCaSpER` *runCaSpER()*

Description

Main casper function that performs a pairwise comparison of all scales from BAF and expression signals to ensure a coherent set of CNV calls.

Usage

```
runCaSpER(object, removeCentromere = T, cytoband = object@cytoband,
  method = "iterative")
```

Arguments

`object` casper object

`removeCentromere`
boolean values determining if centromere regions should be removed from the analysis

`cytoband` cytoband information downloaded from UCSC hg19: http://hgdownload.cse.ucsc.edu/goldenpath/hg1
hg38:http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz

`method` iterative or fixed method. Fixed performs CNV calls on desired baf and expression scale whereas iterative performs pairwise comparison of all expression and baf scale pairs. Iterative method is recommendend. (default: iterative)

Value

list of objects

`splitByOverlap` *splitByOverlap()*

Description

helper function for segment summary. Acknowledgements to https://support.bioconductor.org/p/67118/

Usage

```
splitByOverlap(query, subject, column = "ENTREZID", ...)
```

Index