Package 'SCclust'

September 20, 2018

Type Package

Title Clustering of Single Cell Sequencing Copy Number Profiles to Identify Clones

Date 2018-05-13

Version 1.0.0

Author Alex Krasnitz, Jude Kendall, Junyan Song, Lubomir Chorbadjiev

Maintainer Lubomir Chorbadjiev <lubomir.chorbadjiev@gmail.com>

Description The SCclust package implements feature selection based on breakpoints, permutations for FDRs for Fisher test p-values and identification of the clone structure in single cell copy number profiles.

License MIT | file LICENSE

LazyData TRUE

RoxygenNote 6.0.1

Suggests knitr,
rmarkdown,
testthat

VignetteBuilder knitr

Imports DNAcopy,
futile.logger,
tools,
parallel,
assertthat

URL https://github.com/KrasnitzLab/SCclust

BugReports https://github.com/KrasnitzLab/SCclust/issues

R topics documented:

calc_bins2regions 2
calc_centroareas 2
calc_pinmat 2
calc_ploidies 3
calc_regions2bins 3
case_filenames 4
chrom_numeric 4

find_clones 4
find_subclones 5
fisher_dist 6
fisher_fdr 7
hclust_tree 7
segment_varbin_files 8
sgains_pipeline 8
sim_fisher_wrapper 9
tree_py 9
varbin_input_files 10

Index **11**

`calc_bins2regions` *Converts list of bins from binning scheme to regions.*

Description

Converts list of bins from binning scheme to regions.

Usage

```
calc_bins2regions(gc_df, bins)
```

`calc_centroareas` *Calculates centromere regions (areas).*

Description

Calculates centromere regions (areas).

Usage

```
calc_centroareas(cyto, centromere = c("p11", "q11"))
```

`calc_pinmat` *Select features and generate the incidence table.*

Description

Select features (called as pins), generate the binary matrix with rows as pins and columns as cells.

Usage

```
calc_pinmat(gc_df, segment_df, homoloss = 0, dropareas = NULL, smear = 1,
  chromrange = 1:24, keepboundaries = F)
```

Arguments

gc_df	bining schema used for the analysis
segment_df	the breakpoint table generated by segment_varbin_files.
homoloss	drop out boundary
dropareas	areas of the chromosomes that should be excluded from further analysis (e.g. centromeres)

Value

a list of pinmat and pins objects. pinmat is the incidence table; pins is the bin location

---

calc_ploidies *Calculates ploidies for samples passed in* segment_df.

Description

Calculates ploidies for samples passed in segment_df.

Usage

```
calc_ploidies(gc_df, segment_df, chromrange = 1:24)
```

Arguments

gc_df	bining schema used for the analysis
segment_df	the breakpoint table generated by segment_varbin_files.

Value

ploidies calculated for samples in segment_df.

---

calc_regions2bins *Converts regions to list of bins from binning scheme.*

Description

Converts regions to list of bins from binning scheme.

Usage

```
calc_regions2bins(gc_df, regions)
```

`case_filenames` *Constructs names for various output files based on ‘output_dir‘ and ‘casename‘*

Description

Constructs names for various output files based on ‘output_dir‘ and ‘casename‘

Usage

```
case_filenames(output_dir, casename)
```

`chrom_numeric` *Converts chrom name to numeric and adds ‘chrom.numeric‘ column to the dataframe.*

Description

Converts chrom name to numeric and adds ‘chrom.numeric‘ column to the dataframe.

Usage

```
chrom_numeric(chrom)
```

`find_clones` *Identify nodes in a hierarchical tree which qualify as clones. Identify ’hard’ clones first, then expand them to ’soft’ clones. Expansion may result in clone mergers. Based on hierarchical clustering, identify the hard/soft clones.*

Description

Identify nodes in a hierarchical tree which qualify as clones. Identify ’hard’ clones first, then expand them to ’soft’ clones. Expansion may result in clone mergers. Based on hierarchical clustering, identify the hard/soft clones.

Usage

```
find_clones(hc, fdrthresh = -2, sharemin = 0.85, nshare = 3, bymax = T,
  climbfromsize = 2, climbtoshare = 3)
```

Arguments

hc	An hclust object with additional items generated by hclust_tree.
fdrthresh	maximal allowed value for log10(FDR) for any pair of leaves in a clone node. Default: -2.
sharemin	A feature is considered 'widely shared' if present in sharemin fraction of leaves in a node.Default: 0.90.
nshare	Minimal number of 'widely shared' features in a hard clone. Default: 3.
bymax	Logical. If TRUE (default), use maximal, and otherwise mean, FDR for the node as a criterion for a hard clone.
climbfromsize	An integer: minimal size of a hard clone allowed to be expanded
climbtoshare	An integer: expand the clone as long as the number of widely shared features is at least this value

Value

An hclust object, with hard/soft clones indicated

---

find_subclones *Identify subclones in a clonal branch of a hierarchical tree.*

Description

Iterate the procedure for clone identification for a subset of cells forming a clone.

Usage

```
find_subclones(hc, pinmat, pins, nmin = 6, nsim = 500, njobs = NULL,
  lmmax = 0.001, hcmethod = "average", baseshare = 3, fdrthresh = -2,
  sharemin = 0.85, bymax = T, climbfromsize = 2, climbtoshare = 3,
  clonetype = "soft")
```

Arguments

hc	The hclust object with clones identified.
pinmat	The feature incidence matrix: columns are cells, rows are features, 1 if a feature is present, 0 if not.
nmin	An integer. Default: 6. The minimal allowed size of a clone to be examined for subclones.
nsim	The number of permutation simulations for subclone identification. Default: 500.
lmmax	Numeric value. Default: 0.001. The threshold parameter for a linear fit, passed to fisherfdr function.
hcmethod	Default: average
baseshare	An integer. Default: 3. A balance parameter for controlling minimal number of shared features in a subclone node.
fdrthresh	FDR criterion for subclone nodes. Default: -2.

sharemin	A feature is considered shared if present in sharemin fraction of leaves in a node.Default: 0.85.
bymax	Logical. If TRUE (Default), use maximal pairwise FDR for the node to find subclones, otherwise use mean over all pairs.
climbfromsize	An integer specifying the minimal size of a hard subclone allowed to be expanded. Default: 2.
climbtoshare	An integer the minimal number of widely shared features in a soft subclone Default: 3.
A	two-column matrix, one row per feature, providing the bin number and thepy (sign) of the feature.
clonetype.	A character string specifying whether hard or soft subclones are to be determined. Default: 'soft'.

Value

A list of hclust objects for clones.

`fisher_dist` *Calculates a distance matrix given Fisher FDR true p-values.*

Description

Calculates a distance matrix given Fisher FDR true p-values.

Usage

```
fisher_dist(true_pv, cell_names)
```

Arguments

true_pv	The Fisher's test p-values for the observation.
cell_names	A character vector. The names of cells.

Value

distance matrix based on Fisher's test p-values (mat_dist).

fisher_fdr *Compute FDRs for Fisher's test p-values.*

Description

Linear fit to the tail of empirical null distribution of Fisher p-values; FDR computation: compare true to simulated CDF(empirical null).

Usage

```
fisher_fdr(true_pv, sim_pv, cell_names, lmmax = 0.001)
```

Arguments

true_pv	The Fisher's test p-values for the observation.
sim_pv	The Fisher's test p-values for the permutations.
cell_names	A character vector. The names of cells.
lmmax	Numeric value. Default: 0.001. The threshold parameter for the linear fit.

Value

A list containing the matrix of the FDR values (mat_fdr)

hclust_tree *Build the hierarchical clustering tree.*

Description

Hierarchical clustering with Fisher's test p-values as distance matrix. Also add feature coverage information for each node in the tree.

Usage

```
hclust_tree(pinmat, mat_fdr, mat_dist, hcmethod = "average")
```

Arguments

pinmat	The incidence table generated by calc_pinmat.
mat_fdr	The FDR matrix generated by fisher_fdr
mat_dist	The dissmilarity based on Fisher's test p-values for hierarchical clustering.
hcmethod	Default: average

Value

A hclust objects with new items added.

segment_varbin_files *Generate the segmented profile for each cell.*

Description

Generate the segmented profile for each cell in the input directory using CBS.

Usage

```
segment_varbin_files(varbin_files, gc_df, badbins = NULL)
```

Arguments

varbin_files	list of bin count files for all cells produced by 'varbin' step of 'sgains' package.
gc_df	binning scheme used for the analysis.
badbins	list of bins that should be excluded from the analysis.

Value

The list containing seg quantal and ratio quantal matrix for all cells.

sgains_pipeline *Integration with 'sGAINS' tool.*

Description

This function is called by sGAINS tools to perform the final step in preparation of results: phylogenetic analysis of single-cell genomes represented by their copy-number profiles

Usage

```
sgains_pipeline(scgv_dir, case_name, varbin_dir, varbin_suffix,
  bins_boundaries_filename, cytoband, badbins = NULL, nsim = 150,
  sharemin = 0.85)
```

Arguments

scgv_dir	directory where the results of the analysis should be stored
case_name	name of the case to be used for storing results of the analysis
varbin_dir	directory where output of 'varbin' step of sGAINS(the binning scheme) is located
varbin_suffix	common suffix for files produced by 'varbin' step of sGAINS
cytoband	file name for a cytoband coordinate table for the version of the genome being used
badbins	a file name for a table of bad bins (bins with outlying read counts) for the specified binning scheme
nsim	number of simulations to run for calculating simulated FDR distribution

sharemin	a feature is considered 'widely shared' by leaves of a tree node if present in sharemin fraction of leaves
bins_boundaries	file name for binning scheme to use in the analysis

---

`sim_fisher_wrapper` *Simulate the Fisher's test p-values.*

Description

Given the incidence table for selected features (i.e. pinmat generated by `calc_pins`), computes the Fisher's test p-values for pairwise comparisons. Also perform permutations on the incidence table and compute a set of Fisher's test p-values for each permutation.

Usage

```
sim_fisher_wrapper(pinmat_df, pins_df, njobs = NULL, nsim = 150,
  nsweep = 200, seedme = 123)
```

Arguments

pinmat_df	The incidence table generated by `findpins`.
pins_df	The pin generated by `findpins`. The bin information for the selected feature set.
nsim	the number of permutations/simulations. Default value: 150.

Value

a list of two numeric vector objects. The Fisher's test p-values for the observation (true) and for the permutations (sim).

`tree_py` *Builds HC tree representation based on the distance matrix computed by* `fisher_dist`

Description

Builds HC tree representation based on the distance matrix computed by `fisher_dist`

Usage

```
tree_py(mdist, method, metric = "euclidean")
```

`varbin_input_files` *Collects all bin count files from given directory*

Description

Collects all bin count files from given directory

Usage

```
varbin_input_files(input_file_dir, suffix_pattern = "")
```

Arguments

`input_file_dir`	directory to scan for bin count files
`suffix_pattern`	suffix to select files from input directory

Value

data frame with filenames, cell names and file basenames.

Index

calc_bins2regions, 2
calc_centroareas, 2
calc_pinmat, 2
calc_ploidies, 3
calc_regions2bins, 3
case_filenames, 4
chrom_numeric, 4

find_clones, 4
find_subclones, 5
fisher_dist, 6
fisher_fdr, 7

hclust_tree, 7

segment_varbin_files, 8
sgains_pipeline, 8
sim_fisher_wrapper, 9

tree_py, 9

varbin_input_files, 10