Package: SiFINeT (via r-universe)

January 16, 2025

Type Package

Title Single Cell Feature Identification with Network Topology

Version 1.13

Date 2025-01-01

Author Qi Gao [aut, cre]

Maintainer Qi Gao <gqi@med.umich.edu>

Description Cluster-independent method based on topology structure of gene co-expression network for identifying feature gene sets, extracting cellular subpopulations, and elucidating intrinsic relationships among these subpopulations. Without prior cell clustering, SifiNet circumvents potential inaccuracies in clustering that may influence subsequent analyses. This method is introduced in Qi Gao, Zhicheng Ji, Liuyang Wang, Kouros Owzar, Qi-Jing Li, Cliburn Chan, Jichun Xie ``SifiNet: a robust and accurate method to identify feature gene sets and annotate cells'' (2024) <doi:10.1093/nar/gkae307>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation yes

BuildVignettes yes

VignetteBuilder knitr

Depends R (>= 3.6.0), methods, utils, stats

Imports Rcpp (>= 1.0.9), quantreg (>= 5.94), igraph (>= 1.3.5), Matrix (>= 1.5-1), ggraph (>= 2.0.6), ggplot2 (>= 3.3.6),

Suggests rmarkdown (>= 2.20), knitr (>= 1.42)

LinkingTo Rcpp, RcppArmadillo

Repository CRAN

Date/Publication 2025-01-16 15:10:05 UTC

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libglpk-dev libicu-dev libxml2-dev

Contents

`assign_shared_feature` *assign_shared_feature*

Description

The function assigns non-unique candidate feature genes as shared feature genes into unique feature gene sets

Usage

```
assign_shared_feature(so, min_edge_prop = 0.4)
```

Arguments

`so`	a SiFINeT object
`min_edge_prop`	minimum proportion of edges between a gene and a unique feature gene set for the new gene to be assigned to the set

Details

Candidate feature genes that are not chosen as unique feature genes would be reconsidered as shared feature genes. A non-unique candidate feature gene would be assigned to a unique feature gene group if it is connected to more than min_edge_prop of the unique genes in the group.

Value

SiFINeT object with shared feature genes in featureset updated.

`cal_coexp` *cal_coexp This function calculates the coexpression patterns between genes and returns the coexpression matrix.*

Description

cal_coexp This function calculates the coexpression patterns between genes and returns the coexpression matrix.

Usage

```
cal_coexp(X)
```

Arguments

`X`	Input binarized cell (row) by gene (column) matrix

Value

Coexpression matrix

Author(s)

Qi Gao

`cal_coexp_sp` *cal_coexp_sp This function calculates the coexpression patterns between genes in sparse matrix and returns the coexpression matrix.*

Description

cal_coexp_sp This function calculates the coexpression patterns between genes in sparse matrix and returns the coexpression matrix.

Usage

```
cal_coexp_sp(X)
```

Arguments

`X`	Input binarized cell (row) by gene (column) sparse matrix

Value

Coexpression matrix

Author(s)

Qi Gao

`cal_conn` *cal_conn This function calculates the first 3 order connectivities for each gene and returns the list of vectors of connectivities.*

Description

cal_conn This function calculates the first 3 order connectivities for each gene and returns the list of vectors of connectivities.

Usage

```
cal_conn(data, thres = 3, m = 10L, abso = 1L, niter = 100L)
```

Arguments

`data` Input gene by gene coexpression matrix

`thres` Gene pairs with coexpression exceed thres would be assigned an edge between them in the coexpression network

`m` Sample size used for the calculation of 3rd order connectivities

`abso` Whether to calculate connectivities in absolute network (TRUE) or positive network (FALSE)

`niter` Number of sample used for the calculation of 3rd order connectivities

Value

List of connectivities C1, C2, and C3 ' @export

Author(s)

Qi Gao

`cal_connectivity` *cal_connectivity*

Description

The function calculates the 1st, 2nd and 3rd order connectivities for all genes

Usage

```
cal_connectivity(so, m = 10, niter = 100)
```

Arguments

`so`	a SiFINeT object
`m`	number of neighbors sampled each time for the calculation of 3rd order connectivity
`niter`	number of samples created for the calculation of 3rd order connectivity

Details

For gene i, First order connectivity is defined as the number of edges connected to gene i (degree of the gene node i in the network). Second order connectivity is defined as the proportion of edges between the neighbors of gene i, calculated as number of observed edges between the neighbors of gene i divided by the number of possible edges between the neighbors. Third order connectivity is defined as a weighted proportion of edges between neighbors and neighbors of neighbors of gene i. Third order connectivity is calculated as the mean of edge proportions across weighted samples. Each gene is weighted by the number of edges it has with the neighbors of gene i. Then SiFINeT repeatedly samples m genes for niter times. For each sample, the edge proportion (number of observed edges / number of possible edges) is calculated. And the mean edge proportion across the sample is the 3rd order connectivity for gene i.

Value

SiFINeT object with conn (absolute network connectivities) updated.

`create_network` *create_network*

Description

The function estimates the null distribution of coexpression patterns and generates coexpression network

Usage

```
create_network(so, alpha = 0.05, manual = FALSE, least_edge_prop = 0.01)
```

Arguments

`so` a SiFINeT object

`alpha` the Type I error rate used for FDR control procedure

`manual` whether to manually set threshold for edge assignment

`least_edge_prop` the minimum proportion of edges. Only used when manual = TRUE

Details

Theoretically the distribution of coexpression patterns would converge to standard Gaussian if either one of the gene pair is not feature gene. However in genomics analysis, empirical null could be much more variable than theoretical null. SiFINeT uses estimated null mean and standard deviation to find the threshold for network edges. An edge is assigned to a pair of gene if the absolute value of coexpression pattern between the 2 genes is greater than the threshold Assuming the distribution to be Gaussian, with the estimated null mean and standard deviation, SiFINeT uses SQUAC to control the false discovery rate (FDR) for coexpression patterns. In case the signal is not strong enough and the coexpression network is too sparse, SiFINeT also accept user-defined lower bound for the least proportion of edges. Usually a coexpression network with edge proportion between 0.5% - 10% would have better performance for the detection of feature gene sets.

Value

SiFINeT object with est_ms (estimated mean and sd) and thres (network edge threshold) updated.

References

Jiashun Jin and Tony T. Cai. “Estimating the Null and the Proportion of Non-Null Effects in Large-Scale Multiple Comparisons”. In: Journal of the American Statistical Association 102 (478 2004), pp. 495–506. doi: 10.1198/016214507000000167.

Jichun Xie and Ruosha Li. “False discovery rate control for high dimensional networks of quantile associations conditioning on covariates”. In: J R Stat Soc Series B Stat Methodol (2018). doi: 10.1111/rssb.12288.

`create_SiFINeT_object` *create_SiFINeT_object*

Description

The function classifies count data based on thresholds defined by quantile regression

Usage

```
create_SiFINeT_object(
  counts,
  gene.name = NULL,
  meta.data = NULL,
  data.name = NULL,
  sparse = FALSE,
  rowfeature = TRUE
)
```

Arguments

`counts`	count matrix
`gene.name`	name of the features
`meta.data`	data.frame of meta data
`data.name`	name of dataset
`sparse`	whether the count matrix should be analyzed as sparse matrix
`rowfeature`	whether the count matrix is feature (row) by cell (column)

Value

a SiFINeT object

`enrich_feature_set` *enrich_feature_set*

Description

The function chooses genes that are not found to be feature genes as enriched feature genes and assigns them into unique+shared feature gene sets

Usage

```
enrich_feature_set(so, min_edge_prop = 0.9)
```

Arguments

`so`	a SiFINeT object
`min_edge_prop`	minimum proportion of edges between a gene and a unique+shared feature gene set for the new feature to be assigned to the set

Details

Genes that are not selected as feature genes would be added in the enriched section of the feature gene set if they are connected with more than min_edge_prop of the unique and shared feature genes in each of the feature gene group.

Value

SiFINeT object with enriched feature genes in geneset updated.

EstNull *EstNull This function is a Rcpp version of Wenguang Sun and Tony T. Cai's EstNull.func R function, estimating null distribution from data. Sun, W., & Cai, T. T. (2007). Oracle and Adaptive Compound Decision Rules for False Discovery Rate Control. Journal of the American Statistical Association, 102(479), 901–912.*

Description

EstNull This function is a Rcpp version of Wenguang Sun and Tony T. Cai's EstNull.func R function, estimating null distribution from data. Sun, W., & Cai, T. T. (2007). Oracle and Adaptive Compound Decision Rules for False Discovery Rate Control. Journal of the American Statistical Association, 102(479), 901–912.

Usage

```
EstNull(x, gamma = 0.1)
```

Arguments

`x`	Input vector of all coexpression values
`gamma`	Parameter setting the stopping threshold

Value

List of mean and std

Author(s)

Qi Gao

---

extract_subnetwork *extract_subnetwork*

Description

The function extract a subnetwork from the co-expression network

Usage

```
extract_subnetwork(
  so,
  target_gene_name = NULL,
  target_gene_id = NULL,
  positive = TRUE
)
```

Arguments

`so` a SiFINeT object

`target_gene_name` the names of the target genes in the output network

`target_gene_id` the indices of the target genes in the output network, not used when target_gene_name is not Null

`positive` whether only positive (default) co-expressions or all co-expressions are considered in assigning edges

Value

an adjacency matrix of the output subnetwork

`feature_coexp` *feature_coexp*

Description

The function calculates coexpression patterns between genes

Usage

```
feature_coexp(so)
```

Arguments

`so` a SiFINeT object

Details

The coexpression pattern of a pair of genes is a normalized co-occurrence of high (or equivalently low) expression level of the 2 genes in the classified count matrix. The normalization is based on the estimated quantiles of the low-high separation instead of the quantiles used for quantile regressions. Theoretically, the distribution of coexpression patterns should asymptotically follow standard Gaussian distribution if at least one of the 2 genes is not differentially expressed feature gene.

Value

SiFINeT object with coexp (gene coexpression matrix) updated.

`filter_lowexp` *filter_lowexp*

Description

The function filters out genes with low expression rate and high positive coexpression with genes of same expression level

Usage

```
filter_lowexp(so, t1 = 10, t2 = 0.9, t3 = 0.9)
```

Arguments

`so`	a SiFINeT object
`t1`	threshold for number of total edges connecting the feature node. Lower t1 leads to stricter filtering.
`t2`	threshold for the proportion of positive edges. Lower t2 leads to stricter filtering.
`t3`	threshold for the proportion of edges with features of same expression level. Lower t3 leads to stricter filtering.

Details

When using only mean expression level as independent variable in quantile regression, it is observed that genes with low expression level tend to have large positive S_{ij} with genes that have same median expression level. To reduce the coexpression noise caused by low expression level, it is preferred to filter out genes which have large amount and high proportion of positive coexpressions with genes sharing same median expression level.

Value

SiFINeT object with kset (kept index set) updated.

`find_unique_feature` *find_unique_feature*

Description

The function finds the clustered unique feature genes

Usage

```
find_unique_feature(
  so,
  t1 = 5,
  t2 = 0.4,
  t3 = 0.3,
  t1p = 5,
  t2p = 0.7,
  t3p = 0.5,
  resolution = 1,
  min_set_size = 5
)
```

Arguments

`so` a SiFINeT object

`t1` feature gene selection parameter, lower threshold for 1st order connectivity in absolute network

`t2` feature gene selection parameter, lower threshold for 2nd order connectivity in absolute network

`t3` feature gene selection parameter, lower threshold for 3rd order connectivity in absolute network

`t1p` unique feature gene selection parameter, lower threshold for 1st order connectivity in positive sub-network

`t2p` unique feature gene selection parameter, lower threshold for 2nd order connectivity in positive sub-network

`t3p` unique feature gene selection parameter, lower threshold for 3rd order connectivity in positive sub-network

`resolution` resolution for louvain clustering of unique feature genes

`min_set_size` minimum size for a unique feature gene cluster to be a separate unique feature gene set

Details

SiFINeT first find genes with high 1st (>= t1), 2nd (>= t2) and 3rd (>= t3) order connectivities in absolute network (conn) to be candidate feature genes. Then a positive sub-network is created where only candidate feature gene nodes (fg_id) and edges with positive coexpression patterns (coexp >=

thres) are included. Feature genes genes with high 1st (>= t1p), 2nd (>= t2p) and at least moderate 3rd (>= t3p) order connectivities in positive sub-network (conn2) are chosen to be candidate unique feature genes. Note that when the network is not too sparse, t3p should usually be smaller than t2p for the detection of unique feature genes in transition cell types. The candidate unique feature genes are then separated into groups by louvain clustering (with resolution defined by the resolution parameter), and among them large groups (number of genes greater than min_set_size) are chosen to be unique feature gene sets that represent different cell types.

Value

SiFINeT object with fg_id (candidate feature gene index), uni_fg_id (candidate unique feature gene index), conn2 (connectivities in positive sub-network), uni_cluster (cluster of candidate unique feature genes), selected_cluster (selected unique feature gene clusters), and unique feature genes in featureset updated.

geneset_topology *geneset_topology*

Description

The function plots the topology network of the feature gene sets found by SiFINeT.

Usage

```
geneset_topology(
  so,
  weightthres = 0.3,
  edge_method = 2,
  node_color = "black",
  shiftsize = 0.05,
  boundsize = 0.3,
  prefix = "",
  set_name = NULL
)
```

Arguments

so	a SiFINeT object
weightthres	edges between nodes (feature gene sets) with weight greater than weightthres would be shown in the plot
edge_method	SiFINeT provides 2 methods of calculating edge weight. The number of shared feature genes between feature gene sets would be used when edge_method = 1; while the edge proportion between feature gene sets would be applied if edge_method = 2.
node_color	color of nodes. Should have either length 1 or same length as the number of feature gene sets.

`shiftsize`	set the distance between center of label and the corresponding feature gene sets node.
`boundsize`	set the size of the boundary region.
`prefix`	the prefix of the labels
`set_name`	name of the gene sets

Details

This function visualizes the output feature gene sets of SiFINeT in the form of network. Number of shared feature genes or proportion of edges between feature gene sets could be used to weight the edges. The layout of the nodes is created by create_layout function in ggraph package.

Value

A ggraph (ggplot) object

References

Thomas Lin Pedersen (2022). ggraph: An Implementation of Grammar of Graphics for Graphs and Networks. R package version 2.0.6. https://CRAN.R-project.org/package=ggraph

`norm_FDR_SQAUC` *norm_FDR_SQAUC The function controls the false discovery rate (FDR) of coexpression patterns using SQAUC method Jichun Xie and Ruosha Li. "False discovery rate control for high dimensional networks of quantile associations conditioning on covariates". In: J R Stat Soc Series B Stat Methodol (2018). doi: 10.1111/rssb.12288.*

Description

norm_FDR_SQAUC The function controls the false discovery rate (FDR) of coexpression patterns using SQAUC method Jichun Xie and Ruosha Li. "False discovery rate control for high dimensional networks of quantile associations conditioning on covariates". In: J R Stat Soc Series B Stat Methodol (2018). doi: 10.1111/rssb.12288.

Usage

```
norm_FDR_SQAUC(value, sam_mean, sam_sd, alpha, n, p)
```

Arguments

`value`	the vector of coexpression patterns
`sam_mean`	the estimated sample mean
`sam_sd`	the estimated sample sd
`alpha`	the type I error rate
`n`	the number of cells
`p`	the number of genes

Value

lower bound threshold for genes to be significantly coexpressed

`quantile_thres` *quantile_thres*

Description

The function classifies count data into binary low (0) - high (1) data, based on whether the count number is greater than a threshold.

Usage

```
quantile_thres(so)
```

Arguments

`so`	a SiFINeT object

Details

The threshold used for classification is defined by quantile regression on each gene using Frisch–Newton interior point method ("fn" option for method variable in quantreg package, rq function). By default if no meta data is provided, the quantile regression would be applied on the mean expression level of each cell. The quantile to be estimated in the quantile regression is set to be the estimated 50% quantile of the non-zero part of the expression level for each gene. If the expression level of a gene is low (with median 0), then the threshold is set to be 0.

Value

SiFINeT object with data.thres (categorized count matrix) updated.

References

Koenker, R. and S. Portnoy (1997) The Gaussian Hare and the Laplacean Tortoise: Computability of Squared-error vs Absolute Error Estimators, (with discussion). Statistical Science, 12, 279-300.

Roger Koenker (2022). quantreg: Quantile Regression. R package version 5.94. https://CRAN.R-project.org/package=quantreg

`SiFINeT-class` *The SiFINeT Class*

Description

The SiFINeT Class

Slots

`data` a list of cell (row) by gene (column) count matrix, either regular or sparse matrix

`sparse` whether the count matrix should be analyzed as sparse matrix

`meta.data` matrix of meta data, the number of rows should equal to the number of cells

`gene.name` a vector of names of genes with length equal to the number of genes

`data.name` name of the dataset

`n` number of cells in the dataset

`p` number of genes in the dataset

`data.thres` binarized count matrix

`coexp` matrix of genes coexpression

`est_ms` estimated mean and sd of coexpression values

`thres` lower bound of coexpression (or absolute value of coexpression) for network edge assignment

`q5` 50% quantile for each gene

`kset` index of kept genes after the filtering step

`conn` list of connectivities in absolute network

`conn2` list of connectivities in positive sub-network

`fg_id` index of the candidate feature genes

`uni_fg_id` index of the candidate unique feature genes

`uni_cluster` cluster result of the candidate unique feature genes

`selected_cluster` selected unique feature gene clusters

`featureset` detected set of feature genes

Index