Package 'magical'

July 29, 2024

Title What the Package Does (One Line, Title Case)

Version 1.1.0

Description What the package does (one paragraph).

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends dplyr, Matrix

Imports GenomicRanges, graphics, karyoploteR, regioneR

Contents

Candidate_circuits_construction_without_TAD
Build candidate circuits

Description

This function constructs candidate circuits based on the input data WITHOUT TAD prior.

Usage

```
Candidate_circuits_construction_without_TAD(loaded_data, distance_control)
```

Arguments

loaded_data	The output from Data_loading
distance_control	Bp threshold for paring peaks and genes

Value

A list named "Candidate_circuits" containing the following elements:

- TFs: candidate TFs
- TF_log2Count: log2 transformed TF expression
- Peaks: candidate peaks
- Peak_log2Count: log2 transformed peak accessibility
- Genes: candidate genes
- Gene_log2Count: log2 transformed gene expression
- TF_peak_binding: TF-peak binding matrix
- Peak_Gene_looping: peak-gene looping matrix

See Also

Candidate_circuits_construction_with_TAD() for the version WITH TAD.

Candidate_circuits_construction_with_TAD
Build candidate circuits

Description

This function constructs candidate circuits based on the input data WITH TAD prior.

Usage

```
Candidate_circuits_construction_with_TAD(loaded_data, TAD_file_path)
```

Arguments

`loaded_data`	The output from `Data_loading`
`TAD_file_path`	The path to the TAD prior file

Value

A list named "Candidate_circuits" containing the following elements:

- TFs: candidate TFs
- TF_log2Count: log2 transformed TF expression
- Peaks: candidate peaks
- Peak_log2Count: log2 transformed peak accessibility
- Genes: candidate genes
- Gene_log2Count: log2 transformed gene expression
- TF_peak_binding: TF-peak binding matrix
- Peak_Gene_looping: peak-gene looping matrix

See Also

`Candidate_circuits_construction_without_TAD()` for the version WITHOUT TAD.

`Data_loading` *Loading Data*

Description

Load data from files.

Usage

```
Data_loading(
  Candidate_Gene_file_path,
  Candidate_Peak_file_path,
  scRNA_readcount_file_path,
  scRNA_Gene_file_path,
  scRNA_cellmeta_file_path,
  scATAC_readcount_file_path,
  scATAC_Peak_file_path,
  scATAC_cellmeta_file_path,
  Motif_mapping_file_path,
  Motif_name_file_path,
  Ref_seq_file_path
)
```

Arguments

`scRNA_readcount_file_path`, `scRNA_gene_file_path`, `scRNA_cellmeta_file_path`
: paths to filtered scRNA data of the cell type

`scATAC_readcount_file_path`, `scATAC_peak_file_path`, `scATAC_cellmeta_file_path`
: paths to filtered scATAC data of the cell type

`Motif_mapping_file_path`, `Motif_name_file_path`
: paths to TF motif prior on all ATAC peaks

`Ref_seq_file_path`
: path to the Refseq file for transcription starting site extraction

`Candidate_gene_file_path`, `Candidate_peak_file_path`
: paths to pre-selected candidate genes and peaks for the cell type

Value

A list named "loaded_data" containing the following elements:

- Common_samples: a vector of common samples between scRNA and scATAC data
- Candidate_Genes: a data frame of candidate genes
- Candidate_Peaks: a data frame of candidate peaks
- scRNA_Genes: a data frame of scRNA genes
- scRNA_cells: a data frame of scRNA cells
- scRNA_read_count_matrix: a sparse matrix of scRNA read counts
- scATAC_Peaks: a data frame of scATAC peaks
- scATAC_cells: a data frame of scATAC cells
- scATAC_read_count_matrix: a sparse matrix of scATAC read counts
- Motifs: a data frame of TF motifs
- TF_peak_binding_matrix: a sparse matrix of TF binding on ATAC peaks
- Refseq: a data frame of Refseq

See Also

`Data_loading_from_workspace()`

`Data_loading_from_workspace` *Loading Data*

Description

Load data from workspace.

Usage

```
Data_loading_from_workspace(
  Common_samples,
  Candidate_Genes,
  Candidate_Peaks,
  scRNA_Genes,
  scRNA_cells,
  scRNA_read_count_matrix,
  scATAC_Peaks,
  scATAC_cells,
  scATAC_read_count_matrix,
  Motifs,
  TF_Peak_binding_matrix,
  Refseq
)
```

Value

A list named "loaded_data" containing the following elements:

- Common_samples: a vector of common samples between scRNA and scATAC data
- Candidate_Genes: a data frame of candidate genes
- Candidate_Peaks: a data frame of candidate peaks
- scRNA_Genes: a data frame of scRNA genes
- scRNA_cells: a data frame of scRNA cells
- scRNA_read_count_matrix: a sparse matrix of scRNA read counts
- scATAC_Peaks: a data frame of scATAC peaks
- scATAC_cells: a data frame of scATAC cells
- scATAC_read_count_matrix: a sparse matrix of scATAC read counts
- Motifs: a data frame of TF motifs
- TF_peak_binding_matrix: a sparse matrix of TF binding on ATAC peaks
- Refseq: a data frame of Refseq

See Also

`Data_loading()` for loading data from files.

MAGICAL_circuits_output *MAGICAL output*

Description

MAGICAL output

Usage

```
MAGICAL_circuits_output(
  Output_file_path,
  Candidate_circuits,
  Circuits_linkage_posterior,
  prob_threshold_TF_peak_binding = 0.8,
  prob_threshold_peak_gene_looping = 0.95
)
```

Arguments

Output_file_path	The output file path
Candidate_circuits	The output from "Candidate_circuits" (with or without TAD)
Circuits_linkage_posterior	The output from MAGICAL_estimation
prob_threshold_TF_peak_binding	The threshold of TF-peak binding probability. Default is 0.8
prob_threshold_peak_gene_looping	The threshold of peak-gene looping probability. Default is 0.95

Value

There is no return value for this function, but it will write a file with this format: Gene_symbol - Gene_chr - Gene_TSS - Peak_chr - Peak_start - Peak_end - Looping_prob - TFs(binding prob)

MAGICAL_estimation *MCMC sampling*

Description

This function estimates the parameters.

Usage

```
MAGICAL_estimation(
  loaded_data,
  Candidate_circuits,
  Initial_model,
  iteration_num
)
```

Arguments

loaded_data	The output from Data_loading
Candidate_circuits	The output from "Candidate_circuits" (with or without TAD)
Initial_model	The output from MAGICAL_initialization
iteration_num	The number of iterations

Value

A list named "Circuits_linkage_posterior" containing the following elements:

- TF_Peak_Binding_prob: The posterior probability of TF binding to peaks
- Peak_Gene_Looping_prob: The posterior probability of looping between peaks and genes
- Noise_parameters: The posterior noise parameters

MAGICAL_initialization *MCMC model initialization*

Description

This function initializes the MCMC model for MAGICAL.

Usage

```
MAGICAL_initialization(loaded_data, Candidate_circuits)
```

Arguments

loaded_data	The output from Data_loading
Candidate_circuits	The output from "Candidate_circuits" (with or without TAD)

Details

For MCMC method, the initial values of parameters are not deterministic for the results. Use pseudo bulk data to initialize the model as only at this dimension all data are matched.

Value

A list of initial values for the MCMC model

---

plot_circuits_with_gene *Plot circuit(s)*

Description

This function plots all circuits containing one certain gene.

Usage

```
plot_circuits_with_gene(
  data,
  gene,
  gene_track = T,
  TxDb,
  peak_track = T,
  peaks
)
```

Arguments

data	The data frame written by MAGICAL_circuits_output().
gene	The gene symbol.
gene_track	Whether to plot gene tracks around the region or not. Default is TRUE.
TxDb	The TxDb object corresponding to the reference genome. Required if gene_track is TRUE.
peak_track	Whether to plot peak tracks around the region or not. Default is TRUE.
peaks	The data frame containing the peaks. It should contain 3 columns: the chromosome, start, and end of the peaks. Required if peak_track is TRUE.

Value

A karyoploteR plot.

See Also

plot_circuits_with_idx() and plot_circuits_with_peak()

Examples

```
# data = read.table("MAGICAL_selected_regulatory_circuits.txt", header = T, sep = "\t")
#
# library("TxDb.Hsapiens.UCSC.hg38.knownGene")
#
# peaks = read.table('Demo input files/scATAC peaks.txt', header = F, sep = "\t")
#
# plot_circuits_with_gene(data, gene = "ACOT11", gene_track = T, TxDb.Hsapiens.UCSC.hg38.knownGene, peak_track
```

---

plot_circuits_with_idx

Plot circuit(s)

Description

This function plots one circuit by its index.

Usage

```
plot_circuits_with_idx(data, idx, gene_track = T, TxDb, peak_track = T, peaks)
```

Arguments

data	The data frame written by MAGICAL_circuits_output().
idx	The index of the circuit to be plotted.
gene_track	Whether to plot gene tracks around the region or not. Default is TRUE.
TxDb	The TxDb object corresponding to the reference genome. Required if gene_track is TRUE.
peak_track	Whether to plot peak tracks around the region or not. Default is TRUE.
peaks	The data frame containing the peaks. It should contain 3 columns: the chromosome, start, and end of the peaks. Required if peak_track is TRUE.

Value

A karyoploteR plot.

See Also

plot_circuits_with_gene() and plot_circuits_with_peak()

Examples

```
# data = read.table("MAGICAL_selected_regulatory_circuits.txt", header = T, sep = "\t")
#
# library("TxDb.Hsapiens.UCSC.hg38.knownGene")
#
# peaks = read.table('Demo input files/scATAC peaks.txt', header = F, sep = "\t")
#
# plot_circuits_with_idx(data, 1, gene_track = T, TxDb.Hsapiens.UCSC.hg38.knownGene, peak_track = T, peaks)
```

---

plot_circuits_with_peak

Plot circuit(s)

Description

This function plots all circuits containing one certain peak.

Usage

```
plot_circuits_with_peak(
  data,
  peak_chr,
  peak_start,
  peak_end,
  gene_track = T,
  TxDb,
  peak_track = T,
  peaks
)
```

Arguments

data	The data frame written by MAGICAL_circuits_output().
peak_chr	(Character) the chromosome of the peak.
peak_start	The start position of the peak.
peak_end	The end position of the peak.
gene_track	Whether to plot gene tracks around the region or not. Default is TRUE.
TxDb	The TxDb object corresponding to the reference genome. Required if gene_track is TRUE.
peak_track	Whether to plot peak tracks around the region or not. Default is TRUE.
peaks	The data frame containing the peaks. It should contain 3 columns: the chromosome, start, and end of the peaks. Required if peak_track is TRUE.

Value

A karyoploteR plot.

See Also

plot_circuits_with_idx() and plot_circuits_with_gene()

Examples

```
# data = read.table("MAGICAL_selected_regulatory_circuits.txt", header = T, sep = "\t")
#
# library("TxDb.Hsapiens.UCSC.hg38.knownGene")
#
# peaks = read.table('Demo input files/scATAC peaks.txt', header = F, sep = "\t")
#
# plot_circuits_with_peak(data, peak_chr = "chr4", peak_start = 184817550, peak_end = 184818412, gene_track = T
```

Index