Package 'COMET'

April 8, 2024

Title COMET (Context-specific Optimization of EMT Trajectories)

Version 0.0.0.9000

Author Annice Najafi

Maintainer [Annice Najafi] <annicenajafi@tamu.edu>

Description COMET is a computational framework for inferring EMT trajectories from time-course single cell RNA sequencing data.

License MIT

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports Rmagic,
Seurat,
phateR,
umap,
dtw,
pracma,
dplyr,
ggplot2,
ggpubr,
readxl,
devtools,
tidyr,
tidyverse,
reshape2,
grid,
gridExtra,
diagram,
plotly,
data.table

R topics documented:

`calculate_conf_intervals` *This function calculates confidence intervals for every sample over 10 runs*

Description

This function calculates confidence intervals for every sample over 10 runs

Usage

```
calculate_conf_intervals(data.inputs)
```

Arguments

`data.inputs`	the input datasheet stored in a csv file in the tables dir

Value

nothing, saves results within the Confidence_Interval_Calculations dir

---

`DTW_calculate` *This function calculates the DTW distance bewteen the inferred trajectories for every cutoff and the flow cytometry data*

Description

This function calculates the DTW distance bewteen the inferred trajectories for every cutoff and the flow cytometry data

Usage

```
DTW_calculate(data.inputs, MET.range)
```

Arguments

`data.inputs`	the input datasheet stored in a csv file in the tables dir
`MET.range`	range of time for MET to take place

Value

nothing, saves the matrix in the DTW_Matrix dir

`find.min.alpha` *This function finds the mse_total given a certain alpha*

Description

This function finds the mse_total given a certain alpha

Usage

```
find.min.alpha(
  alpha,
  E_cad,
  hybrid,
  ZEB,
  M_sc,
  Mu_sc,
  eq,
  ref_eq_day,
  timepoints
)
```

Arguments

`alpha`	parameter alpha
`E_cad`	Epithelial percentage, just named E_cad
`hybrid`	hybrid percentage
`ZEB`	Mesenchymal percentage, just named ZEB
`M_sc`	ratio of M/E at steady state
`Mu_sc`	ratio of H/E at steady state
`eq`	what timepoint to start
`ref_eq_day`	steady state timepoint
`timepoints`	total timepoints

Value

mse_total

`find.optimal.cutoff` *Find the optimal cutoff Finds the best number of highly variable EMT genes*

Description

Find the optimal cutoff Finds the best number of highly variable EMT genes

Usage

```
find.optimal.cutoff(data.input)
```

Arguments

data.input	input data to use

Value

optimal cutoff of highly variable genes

---

fit.all.data *This function fits optimal CTMC trajectories to timecourse data*

Description

This function fits optimal CTMC trajectories to timecourse data

Usage

```
fit.all.data(data.inputs, MET.range)
```

Arguments

data.inputs	the input datasheet stored in a csv file in the tables dir
MET.range	range of time for MET to take place

Value

the final dataframe with the optimal trajectories fitted to data

---

fit.CTMC *This function optimally fits 3 CTMC models to data (1st phase, 2nd phase, MET range)*

Description

This function optimally fits 3 CTMC models to data (1st phase, 2nd phase, MET range)

Usage

```
fit.CTMC(data.input, MET.range, opt.cutoff)
```

Arguments

data.input	input data to use
MET.range	range where MET takes place
opt.cutoff	optimal cutoff of highly variable genes

Value

final trajectories, lambda_E, mu, and lambda_M respectively

`generate_pipeline_files` *This function is to be ran for all cutoffs, purpose is to find the optimal number of EMT genes to minimize the DTW distance between the flow cytometry trajectories and data*

Description

This function is to be ran for all cutoffs, purpose is to find the optimal number of EMT genes to minimize the DTW distance between the flow cytometry trajectories and data

Usage

```
generate_pipeline_files(data.inputs, tables.dir, input.data.dir, parallelize)
```

Arguments

`data.inputs`	the input datasheet stored in a csv file in the tables dir
`tables.dir`	directory with a csv file that had the address for the data and metadata
`input.data.dir`	this is the directory where the input file should be saved in
`parallelize`	this boolean variable indicates whether code is parallelized over cores or not, set parallelize to 'TRUE' for parallelizing code over cores

Value

does not return, saves the files in the COMET_populated_files dir

`KS.label.me` *This function perform Kolmogorov Smirnov scoring on data, credit given to Priyanka Chakraborty as the code was adapted from her work and modified*

Description

This function perform Kolmogorov Smirnov scoring on data, credit given to Priyanka Chakraborty as the code was adapted from her work and modified

Usage

```
KS.label.me(exp.mat, genes, topgenes)
```

Arguments

`exp.mat`	gene expression matrix
`genes`	genes
`top200`	receives top 200 highly variable genes for scoring

Value

KS score

run.CTMC — *This function generates trajectories for a CTMC model given parameters*

Description

This function generates trajectories for a CTMC model given parameters

Usage

```
run.CTMC(alph_fun, time.range, M_sc_fun, Mu_sc_fun, p0_fun)
```

Arguments

alph_fun	alpha parameter used in function
time.range	specific timerange to generate trajectory
M_sc_fun	ratio of M/E at steady state
Mu_sc_fun	ratio of H/E at steady state
p0_fun	initial state vector

Value

trajectories along with the resulting p vector

run_pipeline — *This function runs the data driven pipeline for inferring trajectories*

Description

This function runs the data driven pipeline for inferring trajectories

Usage

```
run_pipeline(data.inputs, tables.dir, input.data.dir, cutoff)
```

Arguments

data.inputs	the input datasheet stored in a csv file in the tables dir
tables.dir	directory with a csv file that had the address for the data and metadata
input.data.dir	this is the directory where the input file should be saved in
cutoff	of highly variable EMT genes to be considered

Value

the inferred trajectories, also saves them within the COMET_populated_files directory

`start_pipeline` *Read necessary files and parameters for pipeline to run This function reads all the necessary files to run this pipeline*

Description

Read necessary files and parameters for pipeline to run This function reads all the necessary files to run this pipeline

Usage

```
start_pipeline(tables.dir, input.data.dir)
```

Arguments

`tables.dir`	directory with a csv file that had the address for the data and metadata
`input.data.dir`	this is the directory where the input file should be saved in

Value

nothing, stores necessary parameters for the model in global variables

`transition_matrix` *This function finds the transition matrix given the parameters for the generator matrix*

Description

This function finds the transition matrix given the parameters for the generator matrix

Usage

```
transition_matrix(lambda_E, lambda_M, mu_E, mu_M, t)
```

Arguments

`lambda_E`	Rate of transition from H to E
`lambda_M`	Rate of transition from H to M
`mu_E`	Rate of transition from E to H
`mu_M`	Rate of transition from M to H
`t`	time

Value

the probability transition matrix

Index