Package 'EMTscore'

August 21, 2026

Type Package

Title Calculate EMT Scores Based on Omics Data

Version 0.99.10

Date 2026-08-04

Description Epithelial-mesenchymal transition (EMT) is an important form of cellular plasticity that is fully or partially activated in several biological scenarios including development and disease progression. EMT involves altered expression of hundreds of protein-coding and non-protein-coding genes. Recent studies showed the prevalence of partial EMT in multiple processes such as various cancers and organ fibrosis, which necessitates rigorous quantification of the degree of EMT. While traditional gene set scoring methods such as gene set variation analysis have been used to generate EMT scores from omics data, multiple EMT scoring algorithms and EMT gene sets have been used by different groups without standardization. Furthermore, comparisons of EMT scores computed from different methods and/or different EMT gene sets are generally difficult due to both the context dependent nature of EMT and the lack of tools that comprehensively integrate varying components for EMT scoring. To address this problem, EMTscore enables users to select scoring methods from a list of previously used algorithms and EMT gene sets from a list of gene sets produced from different experiments. Several visualization methods are provided for making publication-quality plots of EMT scores from omics data. The package also implements a principal-component-analysis based method for scoring divergent EMT processes from a single dataset. Overall, EMTscore provides an integrated solution for assessing the degree and complexity of EMT from omics data, and paves the way for standardizing the comparison of EMT programs across multiple contexts.

License GPL-3

URL https://github.com/wenmm/EMTscore

BugReports https://github.com/wenmm/EMTscore/issues

Encoding UTF-8

biocViews Software, GeneExpression, GeneSetEnrichment, SingleCell, Transcriptomics, Visualization, RNASeq, DimensionReduction, PrincipalComponent, MultipleComparison

Imports AUCell, BiocParallel, circlize, ComplexHeatmap, dplyr, GSA, GSEABase, ggplot2, ggpubr, GSVA, magrittr, Matrix, mclust, nsprcomp, rlang, Seurat, stats, utils

Suggests testthat, knitr, rmarkdown, BiocStyle, BiocFileCache, EMTscoreData, ExperimentHub, SingleCellExperiment, SummarizedExperiment, Cairo, RColorBrewer, curl, ggalluvial, ggtext, ggthemes, gridExtra, paletteer, pheatmap

VignetteBuilder knitr

RoxygenNote 7.3.3

LazyData false

NeedsCompilation no

Depends R (>= 4.6.0)

git_url https://git.bioconductor.org/packages/EMTscore

git_branch devel

git_last_commit 66ca8ae

git_last_commit_date 2026-08-04

Repository Bioconductor 3.24

Date/Publication 2026-08-20

Author Haimei Wen [aut, cre] (ORCID: <https://orcid.org/0009-0006-1600-1210>), Daniel Lopez [aut], Tian Hong [aut], National Institutes of Health [fnd] (grant: R35GM149531), National Science Foundation [fnd] (grant: 2243562)

Maintainer Haimei Wen <hudie.luoluo@gmail.com>

Contents

`EMTscore-package` *EMTscore: Calculate 'EMT' Scores Based on 'Omics' Data*

Description

The EMTscore package provides a toolbox to compute epithelial-mesenchymal transition (EMT) scores from bulk and single-cell omics data using a range of gene-set scoring algorithms and curated EMT gene sets, together with visualization helpers for publication-quality figures.

Author(s)

Maintainer: Haimei Wen <hudie.luoluo@gmail.com> (ORCID)

Authors:

- Daniel Lopez
- Tian Hong

Other contributors:

- National Institutes of Health (R35GM149531) [funder]
- National Science Foundation (2243562) [funder]

See Also

Useful links:

- https://github.com/wenmm/EMTscore
- Report bugs at https://github.com/wenmm/EMTscore/issues

`add_EMT_score` *Add EMT Scores to Seurat or SCE Objects*

Description

This function calculates epithelial-mesenchymal transition (EMT) scores for one or more single-cell objects using multiple scoring methods. Supported methods include `"Seurat"`, `"nnPCA"`, `"AUCell"`, `"GSVA"`, `"ssGSEA"`, `"SCSE"`, and `"JASMINE"`.

Usage

```
add_EMT_score(
  objects,
  gmt_file,
  emt_name = "EMT_Score",
  method = c("Seurat", "nnPCA", "AUCell", "GSVA", "ssGSEA", "SCSE", "JASMINE"),
  nnPCA_dim = 1
)
```

Arguments

`objects`	A named list of Seurat or SingleCellExperiment objects that have already been loaded in the R session.
`gmt_file`	Character string. Path to a GMT file containing EMT gene sets.
`emt_name`	Character string. Name of the EMT score column to add.
`method`	Character string. Scoring method to use. One of: `"Seurat"`, `"nnPCA"`, `"AUCell"`,`"GSVA"`, `"ssGSEA"`, `"SCSE"`, or `"JASMINE"`.
`nnPCA_dim`	Integer. Dimension for nnPCA (default = 1).

Value

A named list of Seurat objects with a new metadata column containing EMT scores.

Examples

```
library(ExperimentHub)
eh <- ExperimentHub()
A549_TGFB1 <- eh[["EH10293"]]
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
objects <- list(A549_TGFB1 = A549_TGFB1)
res <- add_EMT_score(objects, gmt_file,
  emt_name = "EMT_Score",
  method = "AUCell"
)
```

`add_EMT_score_Bulk` *Compute EMT scores for one or multiple bulk expression matrices*

Description

Calculate EMT (epithelial-mesenchymal transition) scores for bulk RNA-seq expression data. The function accepts a single expression matrix (rows = genes, columns = samples) or a named list of such matrices, and computes EMT scores for each sample using one of the supported scoring methods.

Usage

```
add_EMT_score_Bulk(
  expr_mat_list,
  gmt_file = NULL,
  emt_name,
  method = c("nnPCA", "AUCell", "ssGSEA", "SCSE", "GSVA", "JASMINE"),
  dimension
)
```

Arguments

`expr_mat_list`	A matrix/data.frame (rows = genes, cols = samples) **or** a named list of such matrices. If a single matrix is provided it will be processed and a data.frame of scores returned. If a list is provided, each element will be processed and a named list of data.frames will be returned.
`gmt_file`	Character. Path to a GMT file (gene sets). The GMT should contain EMT-related genes used by the scoring functions.
`emt_name`	Character. Column name to use for the EMT score in the returned data.frames. Default: `"EMT_Score"`.
`method`	Character. One of `"nnPCA"`, `"AUCell"`,`"GSVA"`, `"ssGSEA"`, `"SCSE"`, or `"JASMINE"`. Default selects the first value.
`dimension`	Integer. Number of components for nnPCA (only used when `method = "nnPCA"`). Default: 1.

Details

Supported methods:

- `"nnPCA"` - non-negative sparse PCA (via `nsprcomp`)
- `"AUCell"` - AUCell AUC scoring
- `"ssGSEA"` - single-sample GSEA
- `"GSVA"`
- `"SCSE"` - Single-Cell Signature Explorer (adapted)
- `"JASMINE"` - JASMINE scoring

- The function expects gene names as `rownames(mat)` and sample names as `colnames(mat)`. It will coerce inputs to numeric matrices internally. - If none of the GMT genes are present in a matrix a warning is issued and that matrix returns `NULL` in the result list. - The scoring helpers (`Execute_nnPCA`, `Execute_AUCell`, `Execute_ssGSEA`, `Execute_SCSE`, `Execute_JAS`) are assumed available in the package namespace (or the environment) and must return a data.frame with sample rows and the score column name given in `score_names`.

Value

If a single matrix is provided, a `data.frame` with rownames = sample names and one column named after `emt_name` containing EMT scores. If a list of matrices is provided, a named `list` of such `data.frames` (one per input matrix) is returned.

Examples

```
# single matrix
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
res <- add_EMT_score_Bulk(
  expr_mat_list = list(s = geneExp), gmt_file,
  emt_name = "EMT", method = "AUCell", dimension = 1
)

# multiple matrices (named list)
exprs <- list(TCGA = geneExp, GTEX = geneExp)
res_list <- add_EMT_score_Bulk(
  expr_mat_list = exprs, gmt_file,
  emt_name = "EMT", method = "nnPCA", dimension = 1
)
```

---

add_EMT_score_multiple

Add Multiple EMT Scores to Seurat or SCE Objects

Description

Computes EMT-related scores for multiple gene sets defined in a GMT file.

Usage

```
add_EMT_score_multiple(
  objects,
  gmt_file,
  emt_names = NULL,
  method = c("Seurat", "nnPCA", "AUCell", "ssGSEA", "GSVA", "SCSE", "JASMINE"),
  nnPCA_dim = 1,
  cores = 1
)
```

Arguments

objects	A named list of Seurat or SingleCellExperiment objects already in memory.
gmt_file	Character string. Path to GMT file with multiple gene sets.
emt_names	Character vector. Column names to assign to the resulting EMT scores.
method	Character string. Scoring method.
nnPCA_dim	Integer. Dimension used in nnPCA.
cores	Integer. Number of CPU cores for parallel computation.

Value

A named list of Seurat objects with multiple EMT score columns added.

Examples

```
library(ExperimentHub)
eh <- ExperimentHub()
A549_TNF <- eh[["EH10291"]]
A549_EGF <- eh[["EH10292"]]
A549_TGFB1 <- eh[["EH10293"]]
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
objects <- list(A549_TGFB1 = A549_TGFB1, A549_EGF = A549_EGF, A549_TNF = A549_TNF)
EMscore_result <- add_EMT_score_multiple(objects, gmt_file,
  emt_names = c("Escore", "Mscore"),
  method = "nnPCA", nnPCA_dim = 1, cores = 2
)
```

Arrange_plots *Arrange multiple ggplot objects with subtitles and common legend*

Description

Arrange multiple ggplot objects with subtitles and common legend

Usage

```
Arrange_plots(
  plots_list,
  ncol_per_row = 2,
  subtitles = NULL,
  fig_title = NULL,
  common_legend = TRUE,
  legend_position = "bottom"
)
```

Arguments

plots_list	List of ggplot objects.
ncol_per_row	Number of plots per row.
subtitles	Optional vector of subtitles (bold).
fig_title	Optional main figure title.
common_legend	Logical, whether to use a common legend.
legend_position	Position of the legend ("bottom", "right", etc.).

Value

Combined ggplot object.

---

cell_annotation_file *Cell-line annotation table*

Description

Sample-level annotation for the cell lines in geneExp, mapping each sample name to its data source and epithelial/mesenchymal cell-type label. Used to demonstrate the plotting helpers.

Usage

```
data(cell_annotation_file)
```

Format

A data.frame with 118 rows and 3 columns: name (sample identifier, matching the columns of geneExp), source (data source, e.g. CCLE), and celltype_annotation (E/M state label).

Source

Sample-level annotation accompanying the cell-line expression matrix deposited at Zenodo, doi:10.5281/zenodo.19487376, subset to the samples retained in geneExp. Creation and provenance code: system.file("script", "make-data.R", package = "EMTscore").

Examples

```
data(cell_annotation_file)
```

compute_Signature_score

Compute Signature Score from Bulk Matrix or Seurat Object

Description

This function computes signature scores using Spearman correlation between a predefined gene signature and expression data. It supports both bulk expression matrices and Seurat objects.

Usage

```
compute_Signature_score(data, signature_file, score_name)
```

Arguments

data	A bulk gene expression matrix (row names are gene names, and the column names are sample) or a Seurat object.
signature_file	Path to the signature weight file (two-column .tsv file).
score_name	Character string. The name to assign to the resulting score column.

Value

A numeric matrix (for bulk input) or a Seurat object with a new metadata column (for Seurat input).

Examples

```
data(geneExp)
signature_file <- system.file("extdata", "stemsig.tsv", package = "EMTscore")
compute_Signature_score(geneExp, signature_file, "Stemness_Score")
```

compute_Signature_score_SingleCell

Compute Multiple Signature Scores for Single-Cell Objects

Description

This function calculates one or more gene signature scores for single-cell datasets stored as Seurat or SingleCellExperiment (SCE) objects. Each score is computed using Spearman correlation between gene expression profiles and signature weights, followed by min-max normalization. The resulting scores are added to the Seurat object metadata.

Usage

```
compute_Signature_score_SingleCell(objects, signature_files, score_names)
```

Arguments

objects	A named list of Seurat or SingleCellExperiment objects Each file will be read, updated, and annotated with computed signature scores.
signature_files	A character vector of paths to signature weight files (two-column '.tsv' or '.txt' files with gene symbols and weights). Multiple files can be provided.
score_names	A character vector of names for the resulting score columns. Must have the same length as 'signature_files'.

Details

Each signature score is calculated by computing the Spearman correlation between the normalized expression of genes and their corresponding signature weights. The resulting scores are then scaled to a 0-1 range and stored in the Seurat object metadata under the specified names.

Value

A named list of Seurat objects, each containing new metadata columns corresponding to the computed signature scores.

Examples

```
obj <- system.file("extdata", "example_seurat_obj.rds", package = "EMTscore")
obj1 <- readRDS(obj)
files <- c(test = obj1)
signature_file1 <- system.file("extdata", "stemsig.tsv", package = "EMTscore")
signature_file2 <- system.file("extdata", "cellular_senescence_sig.tsv", package = "EMTscore")
signature_files <- c(signature_file1, signature_file2)
result <- compute_Signature_score_SingleCell(files, signature_files,
  score_names = c("Stemness_Score", "Senescence_Score")
)
```

---

correlate_sample_scores

Calculate correlations between pathway or signature scores across samples

---

Description

This function calculates pairwise correlations between all numeric columns of two input dataframes that contain sample-level scores (e.g., pathway, signature, or module scores). The correlation is computed only for samples (rows) shared by both dataframes.

Usage

```
correlate_sample_scores(score_mat1, score_mat2, method = "pearson")
```

Arguments

score_mat1	A dataframe or matrix of sample-level scores. Each row represents a sample, and each column represents a pathway or signature.
score_mat2	Another dataframe or matrix of sample-level scores with matching sample row names.
method	Correlation method, one of "pearson" (default), "spearman", or "kendall".

Details

This function is designed for comparing pathway, EMT, or signature scores across samples. It finds the intersection of sample names (row names) between the two dataframes and performs correlation analysis between every pair of numeric score columns.

Value

A dataframe containing:

Pathway_in_score_mat1 Column name from score_mat1

Pathway_in_score_mat2 Column name from score_mat2

Correlation Correlation coefficient

P_value P-value of the correlation test

Valid_n Number of non-missing paired samples used in the test

Examples

```
set.seed(1)
ids <- paste0("sample", seq_len(20))
score_mat1 <- data.frame(PathwayA = rnorm(20), PathwayB = rnorm(20),
  row.names = ids)
score_mat2 <- data.frame(EMT = rnorm(20), row.names = ids)
correlate_sample_scores(score_mat1, score_mat2, method = "pearson")
```

---

data_for_plot *Example data prepared for plotting*

Description

A data frame combining cell annotations with computed scores, ready to be passed to the plotting helpers.

Usage

```
data(data_for_plot)
```

Format

A data frame (rows = cells/samples) of annotations and scores.

Source

Precomputed example object built with `data_prepare` by merging `cell_annotation_file` with nnPCA scores computed from the bundled `geneExp` matrix. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(data_for_plot)
```

---

data_prepare *Prepare data for plotting*

Description

Merge cell annotation and score result, and set rownames based on a user-specified column.

Usage

```
data_prepare(cell_annotation_file, score_result, merge_colname)
```

Arguments

`cell_annotation_file`	A data.frame containing cell annotation.
`score_result`	A data.frame or matrix of computed scores (nnPCA, AUCell, etc.).
`merge_colname`	Character, the column of cell_annotation_file to use as row names.

Value

A data.frame ready for plotting.

Examples

```
data(cell_annotation_file)
data(nnPCA_Result_multiple)
data_for_plot <- data_prepare(cell_annotation_file, nnPCA_Result_multiple, merge_colname = "name")
```

Execute_AUCell *Execute AUCell Scoring from GMT File*

Description

Read a GMT file and compute EMT scores for a gene expression matrix.

Usage

```
Execute_AUCell(exprMatrix, Genesets, score_names)
```

Arguments

exprMatrix	Numeric matrix of gene expression, row names are genes, column names are samples.
Genesets	Character string. Path to GMT file.
score_names	Character string. Column name for the scores.

Value

Data frame of samples with AUCell scores.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
result <- Execute_AUCell(geneExp, gmt_file, score_names = "score")
```

Execute_AUCell_parallel

Compute AUCell Scores for Multiple Gene Sets

Description

This function computes AUCell scores for each gene set in a GMT file using a given gene expression matrix (genes as rows, samples as columns). It runs in parallel across multiple cores and returns a combined data frame of AUCell scores for all pathways.

Usage

```
Execute_AUCell_parallel(exprMatrix, gmt_file, cores)
```

Arguments

exprMatrix	Numeric matrix or data frame of gene expression, where row names are genes and column names are samples.
gmt_file	Character string. Path to the GMT file containing gene sets.
cores	Integer. Number of CPU cores to use for parallel processing.

Value

A data frame containing AUCell scores for each pathway across samples.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
result <- Execute_AUCell_parallel(geneExp, gmt_file, cores = 1)
```

`Execute_E_M_plot` *Plot E vs M scores*

Description

Scatter plot of E vs M scores with mean +/- SD per cell type.

Usage

```
Execute_E_M_plot(
  data_for_plot,
  E_colname,
  M_colname,
  celltype_colname,
 colors = c("#F87189", "#CE9031", "#A48CF5", "#97A430", "#39A7D0", "#E57D5F", "#84C7B9",
    "#E1AF64", "#C26CCF", "#B0BF43", "#57C3E8", "#F29D9E", "#92AAE6")
)
```

Arguments

`data_for_plot`	A data.frame prepared by plot_data_prepare.
`E_colname`	Character, column name for E score.
`M_colname`	Character, column name for M score.
`celltype_colname`	Character, column name for cell type annotation.
`colors`	Vector of colors for cell types.

Value

A ggplot object.

Examples

```
data(data_for_plot)
plot <- Execute_E_M_plot(
  data_for_plot,
  E_colname = "Panchy_et_al_E_signature",
  M_colname = "Panchy_et_al_M_signature",
  celltype_colname = "celltype_annotation",
  colors = c(
    "#F87189", "#CE9031", "#A48CF5", "#97A430", "#39A7D0", "#E57D5F",
    "#84C7B9", "#E1AF64", "#C26CCF", "#B0BF43", "#57C3E8", "#F29D9E", "#92AAE6"
  )
)
```

Execute_GSVA *Execute GSVA Scoring from GMT File*

Description

Read a GMT file and compute EMT scores for a gene expression matrix.

Usage

```
Execute_GSVA(exprMatrix, Genesets, score_names)
```

Arguments

exprMatrix	Numeric matrix of gene expression, row names are genes, column names are samples.
Genesets	Character string. Path to GMT file.
score_names	Character string. Column name for the scores.

Value

Data frame of samples with GSVA scores.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
result <- Execute_GSVA(geneExp, gmt_file, score_names = "score")
```

Execute_GSVA_parallel *Compute GSVA Scores for Multiple Gene Sets in Parallel*

Description

This function calculates GSVA scores for gene sets defined in a GMT file using the GSVA package. Parallel processing is used to speed up computation across pathways.

Usage

```
Execute_GSVA_parallel(exprMatrix, gmt_file, cores)
```

Arguments

exprMatrix	Numeric matrix of gene expression, rows are genes, columns are samples.
gmt_file	Character string. Path to the GMT file containing gene sets.
cores	Integer. Number of CPU cores to use for parallel processing.

Value

A data frame containing GSVA scores for each pathway across samples. Each row corresponds to a pathway with an additional ‘Pathway‘ column.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
result <- Execute_GSVA_parallel(geneExp, gmt_file, cores = 1)
```

Execute_JAS *Execute JASMINE Scoring from GMT File*

Description

Read a GMT file and compute JASMINE scores for a gene expression matrix.

Usage

```
Execute_JAS(exprMatrix, Genesets, score_names)
```

Arguments

exprMatrix	Numeric matrix of gene expression, row names are genes, column names are samples.
Genesets	Character string. Path to GMT file.
score_names	Character string. Column name for the scores.

Value

Data frame of samples with JASMINE scores.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
result <- Execute_JAS(geneExp, gmt_file, score_names = "score")
```

Execute_JASMINE_parallel *Compute JASMINE Scores for Multiple Gene Sets in Parallel*

Description

This function calculates JASMINE scores for gene sets defined in a GMT file. Each pathway score combines normalized rank mean and odds ratio of gene expression across samples. Parallel processing is used to speed up computation across gene sets.

Usage

```
Execute_JASMINE_parallel(exprMatrix, gmt_file, cores)
```

Arguments

exprMatrix	Numeric matrix of gene expression, rows are genes, columns are samples.
gmt_file	Character string. Path to the GMT file containing gene sets.
cores	Integer. Number of CPU cores to use for parallel processing.

Value

A data frame containing JASMINE scores for each pathway across samples.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
result <- Execute_JASMINE_parallel(geneExp, gmt_file, cores = 1)
```

Execute_M_dimension_plot

Plot M dimension scores

Description

Scatter plot of two M scores with mean +/- SD per cell type.

Usage

```
Execute_M_dimension_plot(
  data_for_plot,
  M1_colname,
  M2_colname,
  celltype_colname,
 colors = c("#F87189", "#CE9031", "#A48CF5", "#97A430", "#39A7D0", "#E57D5F", "#84C7B9",
    "#E1AF64", "#C26CCF", "#B0BF43", "#57C3E8", "#F29D9E", "#92AAE6")
)
```

Arguments

data_for_plot	A data.frame prepared by plot_data_prepare.
M1_colname	Column name for M1 score.
M2_colname	Column name for M2 score.
celltype_colname	Column name for cell type annotation.
colors	Vector of colors for cell types.

Value

A ggplot object.

Examples

```
data(cell_annotation_file)
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
nnPCA_Mscore <- Execute_nnPCA(geneExp, gmt_file,
  dimension = 2, score_names = c("M1_score", "M2_score"))
data_for_plot <- data_prepare(cell_annotation_file, nnPCA_Mscore, merge_colname = "name")
plot2 <- Execute_M_dimension_plot(
  data_for_plot,
  M1_colname = "M1_score",
  M2_colname = "M2_score",
  celltype_colname = "celltype_annotation",
  colors = c(
    "#F87189", "#CE9031", "#A48CF5", "#97A430", "#39A7D0", "#E57D5F",
    "#84C7B9", "#E1AF64", "#C26CCF", "#B0BF43", "#57C3E8", "#F29D9E", "#92AAE6"
  )
)
```

`Execute_nnPCA` *Execute Non-Negative Sparse PCA (nnPCA) Scoring*

Description

Execute nnPCA Scoring from GMT File

Usage

```
Execute_nnPCA(exprMatrix, Genesets, dimension, score_names)
```

Arguments

`exprMatrix`	Numeric matrix of gene expression, row names are genes, column names are samples.
`Genesets`	Character string. Path to GMT file.
`dimension`	An integer specifying the number of principal components to compute (default is 1).
`score_names`	Character string. Column name for the scores.

Details

Read a GMT file and compute EMT scores for a gene expression matrix.

Value

Data frame of samples with nnPCA scores.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
result <- Execute_nnPCA(geneExp, gmt_file, dimension = 1, score_names = "score")
```

Execute_nnPCA_parallel

Compute nnPCA Scores for Multiple Gene Sets in Parallel

Description

This function reads a GMT gene set file and performs non-negative sparse PCA (nnPCA) for each gene set in parallel, returning a combined data frame of pathway scores.

Usage

```
Execute_nnPCA_parallel(exprMatrix, Genesets, dimension, cores)
```

Arguments

exprMatrix	Numeric matrix of gene expression, where rows are genes and columns are samples.
Genesets	Character string. Path to the GMT file containing gene sets.
dimension	Integer. Number of nnPCA components to compute.
cores	Integer. Number of cores to use for parallel processing.

Value

A data frame containing the first nnPCA component scores for each pathway.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
result <- Execute_nnPCA_parallel(geneExp, gmt_file, dimension = 1, cores = 1)
```

Execute_SCSE

Execute SCSE Scoring from GMT File

Description

Read a GMT file and compute EMT scores for a gene expression matrix.

Usage

```
Execute_SCSE(exprMatrix, Genesets, score_names)
```

Arguments

exprMatrix	Numeric matrix of gene expression, row names are genes, column names are samples.
Genesets	Character string. Path to GMT file.
score_names	Character string. Column name for the scores.

Value

Data frame of samples with SCSE scores.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
result <- Execute_SCSE(geneExp, gmt_file, score_names = "score")
```

`Execute_SCSE_parallel` *Compute SCSE Scores for Multiple Gene Sets*

Description

This function computes SCSE (Single Cell Signature Explorer) scores for each gene set in a GMT file using a given gene expression matrix (genes as rows, samples as columns). It runs in parallel across multiple cores and returns a combined data frame of SCSE scores for all pathways.

Usage

```
Execute_SCSE_parallel(exprMatrix, gmt_file, cores)
```

Arguments

`exprMatrix`	Numeric matrix of gene expression, where rows are genes and columns are samples.
`gmt_file`	Character string. Path to the GMT file containing gene sets.
`cores`	Integer. Number of CPU cores to use for parallel processing.

Value

A data frame containing SCSE scores for each pathway across samples.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
result <- Execute_SCSE_parallel(geneExp, gmt_file, cores = 1)
```

Execute_ssGSEA *Execute ssGSEA Scoring from GMT File*

Description

Read a GMT file and compute EMT scores for a gene expression matrix.

Usage

```
Execute_ssGSEA(exprMatrix, Genesets, score_names)
```

Arguments

exprMatrix	Numeric matrix of gene expression, row names are genes, column names are samples.
Genesets	Character string. Path to GMT file.
score_names	Character string. Column name for the scores.

Value

Data frame of samples with ssGSEA scores.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
result <- Execute_ssGSEA(geneExp, gmt_file, score_names = "score")
```

Execute_ssGSEA_parallel *Compute ssGSEA Scores for Multiple Gene Sets in Parallel*

Description

This function calculates ssGSEA scores for gene sets defined in a GMT file using the ssGSEA package. Parallel processing is used to speed up computation across pathways.

Usage

```
Execute_ssGSEA_parallel(exprMatrix, gmt_file, cores)
```

Arguments

exprMatrix	Numeric matrix of gene expression, rows are genes, columns are samples.
gmt_file	Character string. Path to the GMT file containing gene sets.
cores	Integer. Number of CPU cores to use for parallel processing.

Value

A data frame containing ssGSEA scores for each pathway across samples. Each row corresponds to a pathway with an additional 'Pathway' column.

Examples

```
data(geneExp)
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
result <- Execute_ssGSEA_parallel(geneExp, gmt_file, cores = 1)
```

`filter_gmt_by_reference` *Filter a GMT file by overlap with a reference GMT*

Description

Removes (or keeps) gene sets from 'target_gmt' according to the proportion of their genes that appear anywhere in 'ref_gmt'. By default, only novel and non-redundant gene sets are kept (overlap < 'cutoff').

Usage

```
filter_gmt_by_reference(
  ref_gmt,
  target_gmt,
  output_gmt,
  cutoff = 0.5,
  keep_low_overlap = TRUE,
  min_genes = 5
)
```

Arguments

`ref_gmt`	Path to the reference GMT file containing the gene universe to compare against.
`target_gmt`	Path to the GMT file that should be filtered.
`output_gmt`	Path where the filtered GMT will be written.
`cutoff`	Numeric threshold between 0 and 1 (default '0.5'). Fraction of overlapping genes.
`keep_low_overlap`	Logical (default 'TRUE'): • 'TRUE' keep gene sets with overlap < 'cutoff' (remove redundant sets) • 'FALSE' keep gene sets with overlap >= 'cutoff' (keep similar sets)
`min_genes`	Integer. Minimum number of genes required in a set (default '5'). Sets with fewer genes are ignored.

Value

Invisibly a ‘data.frame‘ with one row per processed gene set and the columns:

`name`	Gene set name
`size`	Total number of genes in the set
`overlap`	Number of genes present in the reference GMT
`fraction`	‘overlap / size‘

Examples

```
ref_gmt <- system.file("extdata", "TianLab_collected_EMT_signatures.gmt",
  package = "EMTscore")
target_gmt <- system.file("extdata", "h.all.v2025.1.Hs.symbols.gmt",
  package = "EMTscore")
output_gmt <- tempfile(fileext = ".gmt")
# Keep only gene sets that share < 50 % genes with the TianLab EMT collection
filter_gmt_by_reference(
  ref_gmt,
  target_gmt,
  output_gmt,
  cutoff = 0.50,
  keep_low_overlap = TRUE
)
```

`geneExp` *Example gene expression matrix*

Description

A small log2-scale bulk gene expression matrix used to demonstrate the scoring functions in runnable examples. It is a subset of the CCLE / cell line panel distributed with the EMTscore study, restricted to the genes that appear in the bundled EMT signature gene sets (`EM_signature.gmt` and `test.gmt`) so that the examples run quickly without a network download.

Usage

```
data(geneExp)
```

Format

A numeric matrix with 544 rows (genes, row names are HGNC symbols) and 120 columns (samples/cell lines, column names match the `name` column of `cell_annotation_file`). Values are log2-transformed normalized expression.

Source

Subset of the full expression matrix deposited at Zenodo, doi:10.5281/zenodo.19487376 (file `geneExp.rda`). The processing script that produced this subset is in `system.file("script", "make_geneExp.R", package = "EMTscore")`.

Examples

```
data(geneExp)
dim(geneExp)
```

GO *Gene Ontology EMT-related gene sets*

Description

A curated collection of Gene Ontology (GO) gene sets used as input for EMT scoring.

Usage

```
data(GO)
```

Format

A list of gene sets, each a character vector of gene symbols.

Source

EMT-related gene sets derived from the Gene Ontology (GO) database (http://geneontology.org). Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(GO)
```

MSigDB_Hallmark *MSigDB Hallmark gene sets*

Description

Hallmark gene sets from the Molecular Signatures Database (MSigDB), provided as example pathway input.

Usage

```
data(MSigDB_Hallmark)
```

Format

A list of Hallmark gene sets, each a character vector of gene symbols.

Source

Hallmark (H) gene set collection from the Molecular Signatures Database (MSigDB, https://www.gsea-msigdb.org/gsea/msigdb). Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(MSigDB_Hallmark)
```

nnPCA_Result_EMT *nnPCA_Result_EMT*

Description

This dataset contains EMT score computed using the nnPCA method.

Usage

```
data(nnPCA_Result_EMT)
```

Format

A data frame or matrix of nnPCA-based EMT score.
EMT score results generated using the nnPCA method.

Source

Precomputed example output produced by running `Execute_nnPCA` on the bundled `geneExp` matrix with the `EM_signature.gmt` gene set shipped in `system.file("extdata", package = "EMTscore")`. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(nnPCA_Result_EMT)
```

nnPCA_Result_multiple *Example multiple nnPCA results*

Description

Example Escore and Mscore values computed with the nnPCA method across multiple gene sets.

Usage

```
data(nnPCA_Result_multiple)
```

Format

A data frame with nnPCA Escore and Mscore columns (rows = samples).

Source

Precomputed example output produced by running `Execute_nnPCA` (with `dimension = 2`) on the bundled `geneExp` matrix and the `EM_signature.gmt` gene set shipped in `system.file("extdata", package = "EMTscore")`. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(nnPCA_Result_multiple)
```

Panchy_et_al_E_signature

Panchy et al. epithelial signature

Description

Epithelial (E) marker gene set curated from Panchy et al., provided as an example EMT signature.

Usage

```
data(Panchy_et_al_E_signature)
```

Format

A character vector of epithelial-state gene symbols.

Source

Epithelial marker gene list curated from Panchy et al. Gene symbols were extracted from the published EMT signature; no further processing was applied. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(Panchy_et_al_E_signature)
```

Panchy_et_al_M_signature

Panchy et al. mesenchymal signature

Description

Mesenchymal (M) marker gene set curated from Panchy et al., provided as an example EMT signature.

Usage

```
data(Panchy_et_al_M_signature)
```

Format

A character vector of mesenchymal-state gene symbols.

Source

Mesenchymal marker gene list curated from Panchy et al. Gene symbols were extracted from the published EMT signature; no further processing was applied. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(Panchy_et_al_M_signature)
```

plot_EMT_from_objects *Plot EMT Scores Along Pseudotime from Seurat Objects*

Description

This function visualizes EMT (epithelial mesenchymal transition) scores across pseudotime for multiple Seurat objects. Each Seurat object is treated as a different condition or sample.

Usage

```
plot_EMT_from_objects(obj_list, col_name, emt_score_col)
```

Arguments

obj_list	A named list of Seurat objects. Each element should be a Seurat object with the EMT score already computed and stored in the metadata.
col_name	Character string. Name of the column in meta.data containing values to plot.
emt_score_col	Character string. Name of the column in meta.data containing EMT scores. Default: "EMT_Score".

Value

A ggplot object showing smoothed EMT score trajectories along pseudotime for each condition.

Examples

```
library(ExperimentHub)
eh <- ExperimentHub()
A549_TGFB1 <- eh[["EH10293"]]
gmt_file <- system.file("extdata", "EM_signature.gmt", package = "EMTscore")
objs <- add_EMT_score(list(A549_TGFB1 = A549_TGFB1), gmt_file,
  emt_name = "EMT_Score", method = "AUCell"
)
p <- plot_EMT_from_objects(objs,
  col_name = "nCount_RNA",
  emt_score_col = "EMT_Score"
)
```

plot_heatmap_function *present result of heat map of genes that contribute to M score of dimension 1 and disension 2*

Description

present result of heat map of genes that contribute to M score of dimension 1 and disension 2

Usage

```
plot_heatmap_function(geneExp, geneList_M)
```

Arguments

geneExp	gene expression matrix
geneList_M	M signature gene list in dataframe, we supply example list such as geneList_M, M_signature_for_cancer, M_signature_for_cell

Value

Figure represent EMT score for different methods and gene sets

Examples

```
data(geneExp)
data(Tan_et_al_cell_line_M_signature)
geneList_M <- Tan_et_al_cell_line_M_signature
p <- plot_heatmap_function(t(geneExp), geneList_M)
```

---

predict_cluster_labels

Predict Cluster Labels from PCA Results Using GMM or Kmeans

Description

This function clusters PCA results using Gaussian Mixture Model (GMM) or K-means, identifies the clusters with maximum Escore and Mscore, assigns labels "E", "M", and "EM1", "EM2", ... to other clusters, and returns a named vector of final cluster labels for each sample.

Usage

```
predict_cluster_labels(
  pca_result,
  method = "GMM",
  n_clusters = 3,
  PC_name = c("Escore", "Mscore")
)
```

Arguments

pca_result	A numeric matrix or data frame of PCA results (rows = samples, columns = PCs).
method	Character, either "GMM" (Gaussian Mixture Model) or "Kmeans" for clustering. Default is "GMM".
n_clusters	Numeric, number of clusters for Kmeans. Ignored if method = "GMM".
PC_name	Character vector of length 2 specifying the column names for Escore and Mscore. Default is c("Escore", "Mscore").

Value

A named character vector of cluster labels for each sample.

Examples

```
data(nnPCA_Result_multiple)
labels_gmm <- predict_cluster_labels(nnPCA_Result_multiple,
  method = "GMM", n_clusters = 3, PC_name = c("Escore", "Mscore")
)
labels_km <- predict_cluster_labels(nnPCA_Result_multiple,
  method = "Kmeans", n_clusters = 3, PC_name = c("Escore", "Mscore")
)
```

---

read_gmt *Read GMT File and Extract Genes*

Description

This function reads a GMT (Gene Matrix Transposed) file with `GSEABase::getGmt()` and extracts all unique genes across all gene sets.

Usage

```
read_gmt(fname)
```

Arguments

fname	Character string. Path to the GMT file.

Value

A data frame with a single column gene, containing all unique genes.

Examples

```
gmt_file <- system.file("extdata", "test.gmt", package = "EMTscore")
genes <- read_gmt(gmt_file)
```

SCSE_Result_EMT *Example SCSE EMT scores*

Description

Example EMT scores computed with the Single-Cell Signature Explorer (SCSE) method.

Usage

```
data(SCSE_Result_EMT)
```

Format

A data frame of SCSE-based EMT scores (rows = samples).

Source

Precomputed example output produced by running `Execute_SCSE` on the bundled `geneExp` matrix with the `EM_signature.gmt` gene set shipped in `system.file("extdata", package = "EMTscore")`. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(SCSE_Result_EMT)
```

Tan_et_al_cell_line_E_signature
Tan et al. cell-line epithelial signature

Description

Epithelial (E) signature gene set derived from cell-line data in Tan et al.

Usage

```
data(Tan_et_al_cell_line_E_signature)
```

Format

A character vector of epithelial signature gene symbols.

Source

Cell-line epithelial EMT signature curated from Tan et al. Gene symbols were extracted from the published signature; no further processing was applied. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(Tan_et_al_cell_line_E_signature)
```

Tan_et_al_cell_line_M_signature
Tan et al. cell-line mesenchymal signature

Description

Mesenchymal (M) signature gene set derived from cell-line data in Tan et al.

Usage

```
data(Tan_et_al_cell_line_M_signature)
```

Format

A character vector of mesenchymal signature gene symbols.

Source

Cell-line mesenchymal EMT signature curated from Tan et al. Gene symbols were extracted from the published signature; no further processing was applied. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(Tan_et_al_cell_line_M_signature)
```

Tan_et_al_tumor_E_signature

Tan et al. tumor epithelial signature

Description

Epithelial (E) signature gene set derived from tumor data in Tan et al.

Usage

```
data(Tan_et_al_tumor_E_signature)
```

Format

A character vector of epithelial signature gene symbols.

Source

Tumor epithelial EMT signature curated from Tan et al. Gene symbols were extracted from the published signature; no further processing was applied. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(Tan_et_al_tumor_E_signature)
```

Tan_et_al_tumor_M_signature

Tan et al. tumor mesenchymal signature

Description

Mesenchymal (M) signature gene set derived from tumor data in Tan et al.

Usage

```
data(Tan_et_al_tumor_M_signature)
```

Format

A character vector of mesenchymal signature gene symbols.

Source

Tumor mesenchymal EMT signature curated from Tan et al. Gene symbols were extracted from the published signature; no further processing was applied. Creation and provenance code: `system.file("script", "make-data.R", package = "EMTscore")`.

Examples

```
data(Tan_et_al_tumor_M_signature)
```

`write_gmt` *Write Gene Sets to a GMT File*

Description

Convert multiple gene lists into a single GMT (Gene Matrix Transposed) formatted file. Each element of the input list represents one gene set, where the element name is the gene set name and the element value is a vector of gene symbols.

Usage

```
write_gmt(gene_sets, file)
```

Arguments

`gene_sets`	A named list of gene sets. Each element should be a character vector containing gene symbols, and each list name will be used as the gene set name. Example: `list(Pathway1 = c("TP53", "BRCA1"), Pathway2 = c("EGFR", "MYC"))`
`file`	A character string specifying the output GMT file path (e.g. "output.gmt").

Value

No return value. The function writes a GMT file to the specified path.

Examples

```
gene_sets <- list(
  EMT_High = c("VIM", "ZEB1", "SNAI1"),
  EMT_Low  = c("CDH1", "OCLN", "DSP")
)
write_gmt(gene_sets, file = tempfile(fileext = ".gmt"))
```

Index