Package 'SUMO'

May 29, 2025

Title Generating Multi-Omics Datasets for Testing and Benchmarking

Version 1.2.0

Description
Provides tools to simulate multi-omics datasets with predefined signal structures. The generated data can be used for testing, validating, and benchmarking integrative analysis methods such as factor models and clustering approaches. This version includes enhanced signal customization, visualization tools (scatter, histogram, 3D), MOFA-based analysis pipelines, PowerPoint export, and statistical profiling of datasets. Designed for both method development and teaching, SUMO supports real and synthetic data pipelines with interpretable outputs. Tini, Giulia, et al (2019) <doi:10.1093/bib/bbx167>.

License CC BY 4.0

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), MOFAdata, MOFA2, rvg, fabia, tidyverse, grid, basilisk, systemfonts,

Config/testthat/edition 3

Imports ggplot2, gridExtra, rlang, stats, graphics, utils, dplyr, readr, readxl, stringr, data.table, magrittr, officer

Collate 'SUMO.R' 'compute_means_vars.R' 'demo_multiomics_analysis.R' 'divide_vector.R' 'divide_features_one.R' 'divide_features_two.R' 'divide_samples.R' 'divide_samples_alternative.R' 'feature_selection_one.R' 'feature_selection_two.R' 'globals.R' 'plot_factor.R' 'plot_simData.R' 'plot_weights.R' 'simulateMultiOmics.R' 'simulate_twoOmicsData.R'

NeedsCompilation no

Author Bernard Isekah Osang'ir [aut, cre] (ORCID: <https://orcid.org/0000-0002-5557-3602>), Ziv Shkedy [ctb], Surya Gupta [ctb], Jürgen Claesen [ctb]

Maintainer Bernard Isekah Osang'ir <Bernard.Osangir@sckcen.be>

Repository CRAN
Date/Publication 2025-05-29 11:10:02 UTC

Contents

`compute_means_vars` *Compute Summary Statistics for a List of Datasets*

Description

Computes overall, row-wise, and column-wise means and standard deviations for each dataset in a list. Also provides average statistics across datasets.

Usage

```
compute_means_vars(data_list)
```

Arguments

`data_list`	A list of numeric matrices or data frames. Each entry should be a matrix or data frame with numeric values.

Value

A named list containing:

- Overall mean and SD for each dataset.
- Average row-wise mean and SD.
- Average column-wise mean and SD.
- `mean_smp`: Average row-wise mean across all datasets.
- `sd_smp`: Average row-wise SD across all datasets.

Examples

```
# Example using simulated matrices
set.seed(123)
dataset1 <- matrix(rnorm(100, mean = 5, sd = 2), nrow = 10, ncol = 10)
dataset2 <- matrix(rnorm(100, mean = 10, sd = 3), nrow = 10, ncol = 10)
data_list <- list(dataset1, dataset2)
results <- compute_means_vars(data_list)
print(results)

## Not run:
# Example using real experimental data (requires MOFAdata)
if (requireNamespace("MOFAdata", quietly = TRUE)) {
  utils::data("CLL_data", package = "MOFAdata")
  CLL_data2 <- CLL_data[c(2, 3)]
  results <- compute_means_vars(CLL_data2)
  print(results)
}

## End(Not run)
```

`demo_multiomics_analysis` *Demonstration of SUMO Utility in Multi-Omics Analysis using MOFA2*

Description

Run a complete MOFA2-based analysis pipeline using either SUMO-generated or real-world CLL multi-omics data. This function includes preprocessing, MOFA model training, variance decomposition visualization, and optional PowerPoint report generation.

Usage

```
demo_multiomics_analysis(
  data_type = c("SUMO", "real_world"),
  export_pptx = TRUE,
  verbose = TRUE
)
```

Arguments

`data_type`	Character. Options are `"SUMO"` for synthetic data or `"real_world"` for the CLL dataset.
`export_pptx`	Logical. If `TRUE`, saves a PowerPoint summary of the analysis. Default is `TRUE`.
`verbose`	Logical. If `TRUE`, prints progress messages. Default is `TRUE`.

Details

PowerPoint generation is skipped if required packages (`officer`, `rvg`, and `systemfonts >= 1.1.0`) are not available.

Value

Invisibly returns the trained MOFA model object.

See Also

`simulate_twoOmicsData()`, `plot_factor()`, `plot_weights()`

Examples

```
if (
  requireNamespace("MOFA2", quietly = TRUE) &&
  requireNamespace("MOFAdata", quietly = TRUE) &&
  requireNamespace("systemfonts", quietly = TRUE) &&
  utils::packageVersion("systemfonts") >= "1.1.0" &&
  identical(Sys.getenv("NOT_CRAN"), "true")
) {
  demo_multiomics_analysis("SUMO", export_pptx = FALSE)
  demo_multiomics_analysis("real_world", export_pptx = FALSE)
}
```

---

divide_features_one *Dividing features to create vectors with signal in the first omic for single data*

Description

Dividing features to create vectors with signal in the first omic for single data

Usage

```
divide_features_one(n_features_one, num.factor)
```

Arguments

`n_features_one`	number of features of first omic
`num.factor`	number of factor = '1'

divide_features_two *Dividing features to create vectors with signal in the second omic for single data*

Description

Dividing features to create vectors with signal in the second omic for single data

Usage

```
divide_features_two(n_features_two, num.factor)
```

Arguments

n_features_two	number of features of second omic
num.factor	type of factors - single or multiple

divide_samples *Global Variable*

Description

A global variable used in multiple functions.

This utility function divides a sequence of sample indices into num segments ensuring that each segment meets a specified minimum size. It optionally extracts a subset of each segment based on predefined selection logic:

- For a single group (num = 1): selects a random contiguous sub-vector comprising between 10% and 55% of the total samples.
- For multiple groups (num > 1): selects a contiguous sub-vector comprising approximately 75% of each segment.

Usage

```
divide_samples(n_samples, num, min_size)

divide_samples(n_samples, num, min_size)
```

Arguments

n_samples	Integer. Total number of samples to divide.
num	Integer. Number of desired segments or latent factors.
min_size	Integer. Minimum size (length) allowed for each segment.

Details

This function is primarily used for randomized simulation of sample blocks, useful in bootstrapping, subsampling, or simulating latent factor scores across multi-omics datasets.

Value

A list of integer vectors. Each vector contains a sequence of indices representing a subsample of the corresponding segment.

Examples

```
divide_samples(n_samples = 100, num = 3, min_size = 10)
divide_samples(n_samples = 50, num = 1, min_size = 5)
```

`divide_vector` *#' Global Variable #' #' A global variable used in multiple functions. #' #'*

Description

#' Global Variable #' #' A global variable used in multiple functions. #' #'

Usage

```
divide_vector(n_samples, num, min_size)
```

Arguments

`n_samples`	number of samples
`num`	number of factors
`min_size`	Minimum length of any samples scores
	#' ## ~ Updated IN USE (IN USE): Simulate the samples scores (IN USE) ~

feature_selection_one *Dividing features to create vectors with signal in the first omic*

Description

Dividing features to create vectors with signal in the first omic

Usage

```
feature_selection_one(n_features_one, num.factor, no_factor)
```

Arguments

n_features_one	number of features of first omic
num.factor	type of factors - single or multiple
no_factor	number of factors

feature_selection_two *Dividing features to create vectors with signal in the second omic*

Description

Dividing features to create vectors with signal in the second omic

Usage

```
feature_selection_two(n_features_two, num.factor, no_factor)
```

Arguments

n_features_two	number of features of second omic
num.factor	type of factors - single or multiple
no_factor	number of factors

plot_factor *Visualization of factor scores (ground truth)*

Description

Scatter or histogram plots of sample-level factor scores from simulated multi-omics data, using scores from list_alphas and list_gammas.

Usage

```
plot_factor(
  sim_object = NULL,
  factor_num = NULL,
  type = "scatter",
  show.legend = TRUE
)
```

Arguments

sim_object	R object containing simulated data output from simulate_twoOmicsData and simulateMultiOmics.
factor_num	Integer or "all". Which factor(s) to plot.
type	Character. Either "scatter" (default) or "histogram" for plot type.
show.legend	Logical. Whether to show legend in plots. Default is TRUE.

Examples

```
output_obj <- simulate_twoOmicsData(
  vector_features = c(4000,3000),
  n_samples = 100,
  n_factors = 2,
  snr = 2.5,
  num.factor = 'multiple',
  advanced_dist = 'mixed')

plot_factor(sim_object = output_obj, factor_num = 1)
plot_factor(sim_object = output_obj, factor_num = 'all', type = 'histogram')
```

plot_simData *Visualizing the simulated data using heatmap or 3D surface plot*

Description

Generates a visual representation of the simulated omics data either as a heatmap or a 3D surface plot. You can select which dataset to visualize: the merged/concatenated matrix, or any individual omic (e.g., "omic1", "omic2", etc.).

Usage

```
plot_simData(sim_object, data = "merged", type = "heatmap")
```

Arguments

sim_object	R object containing simulated data as returned by simulate_twoOmicsData and simulateMultiOmics.
data	Character. Specifies which data matrix to visualize. Options are "merged" (or "concatenated"), "omic.one", or "omic.two".
type	Character. Type of plot: either "heatmap" for a 2D image plot or "3D" for a 3D perspective surface plot.

Examples

```
output_obj <- simulateMultiOmics(
  vector_features = c(3000, 2500, 2000),
  n_samples = 100,
  n_factors = 3,
  snr = 3,
  signal.samples = c(5, 1),
  signal.features = list(
    c(3, 0.3),
    c(2.5, 0.25),
    c(2, 0.2)
  ),
  factor_structure = "mixed",
  num.factor = "multiple",
  seed = 123
)

# Visualize merged heatmap
plot_simData(sim_object = output_obj, data = "merged", type = "heatmap")

# Visualize omic2 in 3D
plot_simData(sim_object = output_obj, data = "omic2", type = "heatmap")
```

---

plot_weights *Visualizing the raw loading/weights of the features*

Description

Generates scatter or histogram plots of feature loadings (weights) from simulated multi-omics data. Supports plotting for omic.one, omic.two, or integrated views.

Usage

```
plot_weights(
  sim_object,
  factor_num = 1,
  data = "omic.one",
  type = "scatter",
  show.legend = TRUE
)
```

Arguments

sim_object	R object containing data to be plotted.
factor_num	Integer or "all". Specifies which factor(s) to visualize.
data	Character. Section of the data to visualize: "omic.one", "omic.two", or "integrated".
type	Character. Type of plot: "scatter" or "histogram".
show.legend	Logical. Whether to show the legend in the plot. Default is TRUE.

Value

A ggplot object or a combined grid of plots.

Examples

```
output_obj <- simulate_twoOmicsData(
  vector_features = c(4000, 3000),
  n_samples = 100,
  n_factors = 2,
  signal.samples = NULL,
  signal.features.one = NULL,
  signal.features.two = NULL,
  snr = 2.5,
  num.factor = 'multiple',
  advanced_dist = 'mixed'
  )

plot_weights(
   sim_object = output_obj,
   factor_num = 1,
   data = 'omic.one',
   type = 'scatter',
   show.legend = FALSE
   )

plot_weights(
  sim_object = output_obj,
  factor_num = 2,
  data = 'omic.two',
  type = 'histogram'
  )
```

`simulateMultiOmics` *Simulation of omics with predefined single or multiple latent factors in multi-omics*

Description

Simulate multiple omics (>2) datasets with non-overlapping sample and feature signal blocks.

Usage

```
simulateMultiOmics(
  vector_features,
  n_samples,
  n_factors,
  snr = 2,
  signal.samples = c(5, 1),
  signal.features = NULL,
  factor_structure = "mixed",
  num.factor = "multiple",
  seed = NULL
)
```

Arguments

`vector_features`	Integer vector indicating number of features per omic (length k for k omics).
`n_samples`	Total number of samples across all omics.
`n_factors`	Number of latent factors.
`snr`	Signal-to-noise ratio.
`signal.samples`	Mean and SD for generating sample signal values (e.g., c(mean, sd)).
`signal.features`	List of vectors with mean and SD for features per omic (e.g., list(c(3,0.2), c(2.5,0.15))).
`factor_structure`	Character. "shared", "exclusive", "mixed", "partial", or "custom" factor distribution
`num.factor`	Character. "multiple" (default) or "single"
`seed`	Optional. Set random seed for reproducibility.

Details

This function generates synthetic omics data where each omic layer has its own feature space and noise characteristics. The sample signal blocks for each latent factor are non-overlapping and sequential with random gaps. Feature signal blocks are generated per omic with sequential non-overlapping segments.

Value

A list containing:

- `omic.list`: List of simulated omic datasets.
- `signal_annotation`: Signal sample indices per factor.
- `list_alphas`, `list_betas`: Latent factor loading vectors.

Examples

```
sim_object1 <- simulateMultiOmics(
  vector_features = c(3000, 2500, 2000),
  n_samples = 100,
  n_factors = 3,
  snr = 3,
  signal.samples = c(5, 1),
  signal.features = list(
    c(3, 0.3),   # omic1 signal mean/sd
    c(2.5, 0.25),# omic2 signal mean/sd
    c(2, 0.2)    # omic3 signal mean/sd
  ),
  factor_structure = "mixed",
  num.factor = "multiple",
  seed = 123
)

# View available elements
names(sim_object1)

# Visualize the simulated data
plot_simData(sim_object = sim_object1, data = "merged", type = "heatmap")

sim_object2 <- simulateMultiOmics(
  vector_features = c(3000, 2500),
  n_samples = 100,
  n_factors = 1,
  snr = 0.5,
  signal.samples = c(3, 1),
  signal.features = list(
    c(3.5, 0.3),   # omic1 signal mean/sd
    c(4, 0.2)    # omic3 signal mean/sd
  ),
  factor_structure = "shared",
  num.factor = "single",
  seed = NULL
)

# Visualize the simulated data
plot_simData(sim_object = sim_object2, data = "merged", type = "heatmap")
```

`simulate_twoOmicsData` *Simulation of omics with predefined single or multiple latent factors in multi-omics*

Description

Simulates two high-dimensional omics datasets with customizable latent factor structures. Users can control the number and type of factors (shared, unique, mixed), the signal-to-noise ratio, and the distribution of signal-carrying samples and features. The function is flexible for benchmarking multi-omics integration methods under various controlled scenarios.

Usage

```
simulate_twoOmicsData(
  vector_features = c(2000, 2000),
  n_samples = 50,
  n_factors = 3,
  signal.samples = NULL,
  signal.features.one = NULL,
  signal.features.two = NULL,
  num.factor = "multiple",
  snr = 1,
  advanced_dist = NULL,
  ...
)
```

Arguments

`vector_features`	A numeric vector of length two, specifying the number of features in the first and second omics datasets, respectively.
`n_samples`	Integer. The number of samples shared between both omics datasets.
`n_factors`	Integer. Number of latent factors to simulate.
`signal.samples`	Optional numeric vector of length two: the first element is the mean, and the second is the variance of the number of signal-carrying samples per factor. If NULL, signal assignment is inferred from `snr`.
`signal.features.one`	Optional numeric vector of length two: the first element is the mean, and the second is the variance of the number of signal-carrying features per factor in the first omic.
`signal.features.two`	Optional numeric vector of length two: the first element is the mean, and the second is the variance of the number of signal-carrying features per factor in the second omic.
`num.factor`	Character string. Either 'single' or 'multiple'. Determines whether to simulate a single latent factor or multiple factors.

snr	Numeric. Signal-to-noise ratio used to estimate the background noise. The function uses this value to infer the proportion of signal versus noise in the simulated datasets.
advanced_dist	Character string. Specifies how latent factors are distributed when num.factor = 'multiple'. Options include: '', NULL, 'mixed', 'omic.one', 'omic.two', or 'exclusive'.
...	Additional arguments (not currently used).

SUMO *SUMO: Simulation Utilities for Multi-Omics Data*

Description

It provides tools for simulating complex multi-omics datasets, enabling researchers to generate data that mirrors the biological intricacies observed in real-world omics studies. This package addresses a critical gap in current bioinformatics by offering flexible and customizable methods for synthetic multi-omics data generation, supporting method development, validation, and benchmarking.

Details

Key Features:

- **Multi-Omics Simulation**: Generate multi-layered datasets with shared and modality-specific structures.
- **Flexible Generation Engine**: Fine control over samples, noise levels, signal distributions, and latent factor structures.
- **Pipeline-Friendly Design**: Seamlessly integrates with existing multi-omics analysis workflows and packages (e.g., SummarizedExperiment, MultiAssayExperiment).
- **Visualization Tools**: Built-in plotting functions for exploring synthetic signals, factor structures, and noise.

Main Functions:

- simulateMultiOmics(): Simulates multiple (> two) high-dimensional multi-omics datasets.
- simulate_twoOmicsData(): Simulates two high-dimensional multi-omics datasets.
- plot_simData(): Visualizes generated data at different levels.
- plot_factor(): Displays factor scores across samples for signal inspection.
- plot_weights(): Visualizes feature loadings to assess signal versus noise.
- demo_multiomics_analysis(): Full demo function for applying MOFA on SUMO-generated or real-world data.
- compute_means_vars(): Estimate parameters from the real experimental dataset.

Author(s)

Maintainer: Bernard Isekah Osang'ir <Bernard.Osangir@sckcen.be> (ORCID)

Other contributors:

- Ziv Shkedy [contributor]
- Surya Gupta [contributor]
- Jürgen Claesen [contributor]

Index