Package: repli (via r-universe)

November 12, 2026

Title Replicate-based Imputation for Randomly Missing Data in Quantitative Proteomics

Version 0.99.0

Description Implements replicate-based imputation for missing at random (MAR) values in proteomics data. The method leverages existing measurements from other replicates to determine imputation parameters (mean of observed values and global standard deviation).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports dplyr, magrittr, matrixStats, SummarizedExperiment, tibble, tidyr

Suggests DEP, knitr, rmarkdown, BiocStyle, testthat (>= 3.0.0)

biocViews Proteomics, MassSpectrometry, Software

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev zlib1g-dev

Repository https://biocstaging.r-universe.dev

Date/Publication 2026-11-12 15:38:16 UTC

RemoteUrl https://github.com/BiocStaging/repli

RemoteRef HEAD

RemoteSha 65bb55f5328fb28d4678508625d84f22f66ca6bf

Contents

example_dataset 2
example_se 2
impute_replicate 3

impute_replicate_mixed_se 4
impute_replicate_se 5
prepare_imputation_data 6
prepare_imputation_data_se 7
reconstruct_se 8

Index **9**

`example_dataset` *Example Proteomics Dataset*

Description

A small subset of a proteomics dataset used to demonstrate replicate-based imputation.

Usage

```
data(example_dataset)
```

Format

A data frame containing protein abundances and experimental conditions in columns. Protein names are set as row names.

Source

Data derived from a subset of the dataset reported in doi:10.1016/j.celrep.2025.115362.

Examples

```
data(example_dataset)
head(example_dataset)
```

`example_se` *Example Proteomics SummarizedExperiment*

Description

A SummarizedExperiment object containing the example proteomics dataset and experimental design information.

Usage

```
data(example_se)
```

Format

A `SummarizedExperiment` object containing protein abundance measurements and experimental metadata.

Source

Data derived from a subset of the dataset reported in doi:10.1016/j.celrep.2025.115362.

Examples

```
data(example_se)
example_se
```

`impute_replicate` *Perform replicate-based imputation*

Description

Performs replicate-based imputation of Missing At Random (MAR) values. Missing values are imputed by sampling from a normal distribution using the mean of the observed replicates for the corresponding protein-condition pair and a global standard deviation derived from replicates of all protein-condition pairs across the dataset.

Usage

```
impute_replicate(prepared_data)
```

Arguments

`prepared_data`	An object returned by `prepare_imputation_data()` or `prepare_imputation_data_se()`. Contains a list consisting of data in a format prepared for imputation (including missingness classification and protein-condition means), a vector specifying the columns containing replicate measurements and the global standard deviation.

Value

A data frame containing imputed data. MAR values are imputed, while values classified as MNAR or observed are left unchanged.

Examples

```
#in this instance data is a data frame with proteins as rows and
#experimental conditions as columns. Protein names are set as rownames.
data(example_dataset)
prepared <- prepare_imputation_data(example_dataset)
imputed <- impute_replicate(prepared)
```

`impute_replicate_mixed_se` *Perform mixed MAR/MNAR imputation on a SummarizedExperiment object*

Description

Performs mixed imputation of missing values in a SummarizedExperiment object. Missing At Random (MAR) values are imputed using replicate-based imputation, while Missing Not At Random (MNAR) values are imputed using a specified method from the DEP R package. Both methods are applied independently to the original dataset, after which their results are combined.

Usage

```
impute_replicate_mixed_se(
  se,
  p = 0.05,
  MNAR_method = "MinProb",
  MNAR_args = list(q = 0.01)
)
```

Arguments

`se`	A `SummarizedExperiment` object containing protein abundance measurements. Sample names must contain a condition and replicate identifier separated by an underscore or period, such as `"control_1"` or `"treatment.2"`.
`p`	A numeric value between 0 and 1 specifying the proportion of proteins with the highest standard deviation across replicates, used to estimate the global standard deviation. Default is `0.05`. This means the global standard deviation will be calculated as the median of the top 5% highest standard deviations across all protein-condition pairs in the dataset.
`MNAR_method`	A character string specifying the DEP imputation method used for MNAR values, passed to `DEP::impute()`. Default is `"MinProb"`.
`MNAR_args`	A named list of additional arguments passed to `DEP::impute()`. Default is `list(q = 0.01)`.

Details

Protein-condition pairs are classified as MAR if there are missing values for some but not all replicates. Protein-condition pairs are classified as MNAR if there are missing values for all of the replicates of a condition.

#' MAR and MNAR imputation are performed independently on the original dataset. Missing MAR values are imputed using replicate-based imputation, while missing MNAR values are imputed using the specified DEP method. The results are subsequently combined, retaining all originally observed values.

When `MNAR_method = "none"`, MNAR values are left missing.

Value

A SummarizedExperiment object containing the imputed assay. MAR values are imputed using replicate-based imputation and MNAR values are imputed using the specified DEP method.

Examples

```
data(example_se)
imputed_se <- impute_replicate_mixed_se(
  example_se,
  p = 0.05,
  MNAR_method = "MinProb",
  MNAR_args = list(q = 0.01)
)
```

`impute_replicate_se` *Perform replicate-based imputation on a SummarizedExperiment object*

Description

Performs replicate-based imputation of Missing At Random (MAR) values in a SummarizedExperiment object. Missing values are imputed by sampling from a normal distribution using the mean of the observed replicates for the corresponding protein-condition pair and a global standard deviation derived from replicates of all protein-condition pairs across the dataset.The assay is prepared using `prepare_imputation_data_se()`, imputation is performed using `impute_replicate()`, and the imputed values are returned in the original SummarizedExperiment structure using `reconstruct_se()`.

Usage

```
impute_replicate_se(se, p = 0.05)
```

Arguments

`se`	A `SummarizedExperiment` object containing protein abundance measurements. Sample names must contain a condition and replicate identifier separated by an underscore or period, such as `"control_1"` or `"treatment.2"`.
`p`	A numeric value between 0 and 1 specifying the proportion of proteins with the highest standard deviation across replicates, used to estimate the global standard deviation. Default is `0.05`. This means the global standard deviation will be calculated as the median of the top 5% highest standard deviations across all protein-condition pairs in the dataset.

Value

A SummarizedExperiment object with MAR values imputed. Completely missing protein-condition pairs, classified as MNAR, are not imputed.

Examples

```
data(example_se)
imputed_se <- impute_replicate_se(example_se)
```

`prepare_imputation_data`

Prepare data for replicate-based imputation

Description

Prepares a data frame containing protein abundance values for replicate-based imputation. Data is transformed in long format, with protein-condition pairs as rows and replicates as columns. Missingness is classified as Missing At Random (MAR), Missing Not At Random (MNAR) or observed. For protein-condition pairs classified as MAR, the mean abundance is calculated. A global standard deviation is derived from the standard deviations of replicates of all protein-condition pairs across the dataset.

Usage

```
prepare_imputation_data(data, p = 0.05)
```

Arguments

`data`	A data frame containing protein abundance values, with protein names as row names and experimental conditions as column names. Column names should follow condition-replicate naming convention such as `"control_1"` or `"treatment.2"`. An underscore or period must separate the condition and replicate identifier.
`p`	A numeric value between 0 and 1 specifying the proportion of proteins with the highest standard deviation across replicates, used to estimate the global standard deviation. Default is `0.05`. This means the global standard deviation will be calculated as the median of the top 5% highest standard deviations across all protein-condition pairs in the dataset.

Details

Protein-condition pairs are classified as MAR if there are missing values for some but not all replicates. Protein-condition pairs are classified as MNAR if there are missing values for all of the replicates of a condition.

Value

A list containing:

data The data in a format prepared for imputation, including missingness classification and protein-condition means.

replicate_cols A vector specifing the replicate columns.

global_sd The global standard deviation among the proteins in the upper `p` proportion of replicate variability.

Examples

```
data(example_dataset)
prepared <- prepare_imputation_data(example_dataset)
```

`prepare_imputation_data_se` *Prepare a SummarizedExperiment for replicate-based imputation*

Description

Extracts abundance data from a SummarizedExperiment and prepares it for replicate-based imputation using `prepare_imputation_data()`. The experimental design is also extracted from the sample names for downstream reconstruction of the SummarizedExperiment.

Usage

```
prepare_imputation_data_se(se, p = 0.05)
```

Arguments

`se`	A `SummarizedExperiment` object containing protein abundance measurements. Sample names must contain a condition and replicate identifier separated by an underscore or period, such as `"control_1"` or `"treatment.2"`.
`p`	A numeric value between 0 and 1 specifying the proportion of proteins with the highest standard deviation across replicates, used to estimate the global standard deviation. Default is `0.05`. This means the global standard deviation will be calculated as the median of the top 5% highest standard deviations across all protein-condition pairs in the dataset.

Value

A list containing:

data The data in a format prepared for imputation, including missingness classification and protein-condition means.

replicate_cols A vector specifing the replicate columns.

global_sd The global standard deviation among the proteins in the upper `p` proportion of replicate variability.

experimental_design A data frame describing the experimental design, containing columns `label`, `condition`, and `replicate`

Examples

```
data(example_se)
prepared <- prepare_imputation_data_se(example_se)
imputed <- impute_replicate(prepared)
```

`reconstruct_se` *Reconstruct a SummarizedExperiment after replicate-based imputation*

Description

Places imputed data from `impute_replicate()` back into the original SummarizedExperiment object supplied in `prepare_imputation_data_se()`.

Usage

```
reconstruct_se(imputed_data, se, experimental_design)
```

Arguments

`imputed_data`	A data frame containing the imputed data returned by `impute_replicate()`.
`se`	The original SummarizedExperiment object supplied in `prepare_imputation_data_se()`.
`experimental_design`	A data frame describing the experimental design, containing columns `label`, `condition`, and `replicate`.

Value

A SummarizedExperiment object containing the imputed assay. The original row and column metadata are retained.

Examples

```
data(example_se)
prepared <- prepare_imputation_data_se(example_se)
imputed <- impute_replicate(prepared)
result <- reconstruct_se(
  imputed,
  example_se,
  prepared$experimental_design
)
```

Index

∗ **datasets**
example_dataset, 2
example_se, 2

DEP::impute(), *4*

example_dataset, 2
example_se, 2

impute_replicate, 3
impute_replicate(), *5*, *8*
impute_replicate_mixed_se, 4
impute_replicate_se, 5

prepare_imputation_data, 6
prepare_imputation_data(), *3*, *7*
prepare_imputation_data_se, 7
prepare_imputation_data_se(), *3*, *5*, *8*

reconstruct_se, 8
reconstruct_se(), *5*