Package ‘ProtExp’

May 2, 2017

Type Package

Title An Example Package for Proteomics Experiments

Version 1.0

Date 2017-05-02

Author NEU May Institute Advanced R Course

Description ProtExp is an example package illustrating
several R concepts for building packages
for analysis of proteomics experiments.

Maintainer Nobody `<nobody@no.where>`

License Artistic-2.0

LazyData true

Depends methods

Suggests tidyverse, knitr

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

`is_log_trans` *Is a dataset log-transformed?*

Description

Returns `TRUE` if the data has been log-transformed and `FALSE` otherwise.

Usage

```
is_log_trans(object)

is_log_trans(object) <- value

## Default S3 method:
is_log_trans(object)

## Default S3 replacement method:
is_log_trans(object) <- value
```

Arguments

`object`	An object with data that may have been log-transformed.

Details

This is an S3 generic function: methods can be defined for it by writing functions with the naming convention `is_log_trans.classname`. Functions should return `TRUE` or `FALSE`. The default method gets and sets the `is_log_trans` attribute.

Value

`TRUE` if the data has been log-transformed and `FALSE` otherwise.

Examples

```
x <- structure("test", is_log_trans = TRUE)
is_log_trans(x)
is_log_trans(x) <- FALSE
is_log_trans(x)
```

`normalize` *Normalize a dataset*

Description

Performs normalization on an experimental dataset.

Usage

```
normalize(object, ...)

## S3 method for class 'ProtExp'
normalize(object, ..., by)
```

Arguments

`object`	An object with data to be normalized.

Details

This is an S3 generic function: methods can be defined for it by writing functions with the naming convention `normalize.classname`. Functions should be appropriate for the experiment type.

`normalize.ProtExp` performs normalization on a `ProtExp` proteomics experiment dataset object using the median-of-medians method. It also performs log2-transformation on the intensities if the data has not already been log2-transformed.

Value

Another object of the same class with the normalized data.

`ProtExp` *Proteomics Experiment*

Description

Create a `ProtExp` object.

Usage

```
ProtExp(protein, feature, run, intensity, label, ..., is_log_trans = FALSE)
```

Arguments

`protein`	A `character` vector of protein names.
`feature`	A `character` vector of MS features.
`run`	A `character` or `numeric` vector of MS runs.
`intensity`	A `numeric` vector of MS intensities.
`label`	A `character` or `factor` of MS labels.
`...`	Additional columns.
`is_log_trans`	`TRUE` or `FALSE` indicating whether the intensities has been log2-transformed or not.

Details

`ProtExp` is a simple S3 class inheriting from `data.frame` for storing proteomics experiments. It expects columns named ‘protein’, ‘feature’, ‘run’, ‘intensity’, and ‘label’. Additional columns are also allowed. A `ProtExp` object also has a `is_log_exp` attribute for tracking whether the intensities have been log2-transformed or not.

Value

object A `ProtExp` object.

twin *Twin DIA and SRM experiments*

Description

DIA and SRM datasets from the same experiment. These datasets are subsets of the original datasets and have 33 proteins by DIA, and 39 proteins by SRM. The dataset includes 58 pairs of monozygotic (MZ) and dizygotic (DZ) twins, each measured at 2 time points, yielding 58 x 2 x 2 = 232 MS runs. The variables are as follows:

Format

Two data frames with 103936 rows and 9 variables for `twin_dia` and 63104 rows and 9 variables for `twin_srm`.

Details

- protein. (chr) protein name.
- feature. (chr) combination of peptide, precursor charge state, fragment ion, and product charge state, separated by _.
- run. (chr) MS run identifier (R001-R232).
- pair. (int) pair identifier number (1-58).
- zygosity. (factor) zygosity (MZ, DZ).
- subject. (int) subject identifier number (1-116).
- visit. (int) time of visit (1, 2).
- intensity_h. (num) integrated feature intensity from light (L) channel.
- intensity_l. (num) integrated feature intensity from heavy (H, aka reference) channel.

Examples

```
data(twin)
head(twin_dia)
head(twin_srm)
```

Index