Package 'LongDat'

June 21, 2022

Type Package

Title A Tool for 'Covariate'-Sensitive Longitudinal Analysis on 'omics' Data

Version 1.0.5

Description This tool takes longitudinal dataset as input and analyzes if there is significant change of the features over time (a proxy for treatments), while detects and controls for 'covariates' simultaneously. 'LongDat' is able to take in several data types as input, including count, proportion, binary, ordinal and continuous data. The output table contains p values, effect sizes and 'covariates' of each feature, making the downstream analysis easy.

License GPL-2

Encoding UTF-8

Language en-US

LazyData true

URL https://github.com/CCY-dev/LongDat

BugReports https://github.com/CCY-dev/LongDat/issues

RoxygenNote 7.1.2

Depends R (>= 4.0.0)

Imports lme4, reshape2, glmmTMB, emmeans, bestNormalize, MASS, ggplot2, stringr, magrittr, tibble, dplyr, graphics, utils, stats, rlang, car, rstatix, effsize, tidyr, patchwork

Suggests rmarkdown, knitr, tidyverse, kableExtra

VignetteBuilder knitr

NeedsCompilation no

Author Chia-Yu Chen [aut, cre] (<https://orcid.org/0000-0003-1765-7132>), Sofia Forslund [ctb] (<https://orcid.org/0000-0003-4285-6993>)

Maintainer Chia-Yu Chen <Chia-Yu.Chen@mdc-berlin.de>

Repository CRAN

Date/Publication 2022-06-21 16:20:02 UTC

R topics documented:

`cliff_cal` *Effect size (Cliff's delta) calculation in longdat_disc() pipeline*

Description

Effect size (Cliff's delta) calculation in longdat_disc() pipeline

Arguments

`melt_data`	Internal function argument.
`Ps_poho_fdr`	Internal function argument.
`variables`	Internal function argument.
`test_var`	Internal function argument.
`data`	Internal function argument.

verbose	Internal function argument.

`ConModelTest_cont` *Covariate model test in longdat_cont() pipeline*

Description

Covariate model test in longdat_cont() pipeline

Arguments

N	Internal function argument.
variables	Internal function argument.
melt_data	Internal function argument.
sel_fac	Internal function argument.
data_type	Internal function argument.
test_var	Internal function argument.
verbose	Internal function argument.

`ConModelTest_disc` *Covariate model test in longdat_disc() pipeline*

Description

Covariate model test in longdat_disc() pipeline

Arguments

N	Internal function argument.
variables	Internal function argument.
melt_data	Internal function argument.
sel_fac	Internal function argument.
data_type	Internal function argument.
test_var	Internal function argument.
verbose	Internal function argument.

correlation_posthoc *Post-hoc test based on correlation test for longdat_cont().*

Description

Post-hoc test based on correlation test for longdat_cont().

Usage

```
correlation_posthoc(variables, verbose, melt_data, test_var, N)
```

Arguments

variables	Internal function argument.
verbose	Internal function argument.
melt_data	Internal function argument.
test_var	Internal function argument.
N	Internal function argument.

cuneiform_plot *Create cuneiform plots of result table from longdat_disc() or longdat_cont()*

Description

Create cuneiform plots of result table from longdat_disc() or longdat_cont()

Arguments

result_table	The result table from longdat_disc() or longdat_cont() output, or any data frame that has the same format.
x_axis_order	The plotting order of the x axis. It should be a character vector (e.g. c("Effect_1_2", "Effect_2_3", "Effect_1_3")).
covariate_panel	A boolean vector indicating whether to plot covariate status alongside the effect panel. The default is TRUE.
pos_color	The color for a positive effect size. It should be a hex color code (e.g. "#b3e6ff") or the colors recognized by R. The default is "red".
neg_color	The color for a negative effect size. It should be a hex color code (e.g. "#b3e6ff") or the colors recognized by R. The default is "blue".
panel_width	The width of the effect size panel on the left relative to the covariate status panel on the right (width set to 1). It should be a numerical vector. The default is 4.

title	The name of the plot title. The default is "LongDat result cuneiform plot".
title_size	The size of the plot title. The default is 20.
covariate_text_size	The size of the text in the covariate status panel. The default is 4.
x_label_size	The size of the x label. The default is 10.
y_label_size	The size of the y label. The default is 10.
legend_title_size	The size of the legend title. The default is 12.
legend_text_size	The size of the legend text The default is 10.

Details

This function creates a cuneiform plot which displays the result of longdat_disc() or longdat_cont(). It plots the effect sizes within each time interval for each feature, and also shows the covariate status. Only the features with non-NS signals will be included in the plot. The output is a ggplot object in patchwork structure. For further customization of the plot, please refer to the vignette.

Value

a 'ggplot' object

Examples

```
test_disc <- longdat_disc(input = LongDat_disc_master_table,
data_type = "count", test_var = "Time_point",
variable_col = 7, fac_var = c(1:3))
test_plot <- cuneiform_plot(result_table = test_disc[[1]],
x_axis_order = c("Effect_1_2", "Effect_2_3", "Effect_1_3"))
```

---

data_preprocess *Data preprocessing*

Description

Data preprocessing

Usage

```
data_preprocess(input, test_var, variable_col, fac_var, not_used)
```

Arguments

input	Internal function argument.
test_var	Internal function argument.
variable_col	Internal function argument.
fac_var	Internal function argument.
not_used	Internal function argument.

`factor_p_cal` *Calculate the p values for every factor (used for selecting factors later)*

Description

Calculate the p values for every factor (used for selecting factors later)

Usage

```
factor_p_cal(melt_data, variables, factor_columns, factors, data, N, verbose)
```

Arguments

`melt_data`	Internal function argument.
`variables`	Internal function argument.
`factor_columns`	Internal function argument.
`factors`	Internal function argument.
`data`	Internal function argument.
`N`	Internal function argument.
`verbose`	Internal function argument.

`final_result_summarize_cont` *Generate result table as output in longdat_cont()*

Description

Generate result table as output in longdat_cont()

Usage

```
final_result_summarize_cont(
  variable_col,
  N,
  Ps_conf_inv_model_unlist,
  variables,
  sel_fac,
  Ps_conf_model_unlist,
  model_q,
  posthoc_q,
  Ps_null_model_fdr,
  Ps_null_model,
  assoc,
```

```
  prevalence,
  mean_abundance,
  p_poho,
  not_used,
  Ps_effectsize,
  data_type,
  false_pos_count
)
```

Arguments

variable_col	Internal function argument.
N	Internal function argument.
Ps_conf_inv_model_unlist	Internal function argument.
variables	Internal function argument.
sel_fac	Internal function argument.
Ps_conf_model_unlist	Internal function argument.
model_q	Internal function argument.
posthoc_q	Internal function argument.
Ps_null_model_fdr	Internal function argument.
Ps_null_model	Internal function argument.
assoc	Internal function argument.
prevalence	Internal function argument.
mean_abundance	Internal function argument.
p_poho	Internal function argument.
not_used	Internal function argument.
Ps_effectsize	Internal function argument.
data_type	Internal function argument.
false_pos_count	Internal function argument.

`final_result_summarize_disc` *Generate result table as output in longdat_disc()*

Description

Generate result table as output in longdat_disc()

Usage

```
final_result_summarize_disc(
  variable_col,
  N,
  Ps_conf_inv_model_unlist,
  variables,
  sel_fac,
  Ps_conf_model_unlist,
  model_q,
  posthoc_q,
  Ps_null_model_fdr,
  Ps_null_model,
  delta,
  case_pairs,
  prevalence,
  mean_abundance,
  Ps_poho_fdr,
  not_used,
  Ps_effectsize,
  case_pairs_name,
  data_type,
  false_pos_count,
  p_wilcox_final
)
```

Arguments

variable_col	Internal function argument.
N	Internal function argument.
Ps_conf_inv_model_unlist	Internal function argument.
variables	Internal function argument.
sel_fac	Internal function argument.
Ps_conf_model_unlist	Internal function argument.
model_q	Internal function argument.
posthoc_q	Internal function argument.
Ps_null_model_fdr	Internal function argument.
Ps_null_model	Internal function argument.
delta	Internal function argument.
case_pairs	Internal function argument.
prevalence	Internal function argument.
mean_abundance	Internal function argument.

Ps_poho_fdr	Internal function argument.
not_used	Internal function argument.
Ps_effectsize	Internal function argument.
case_pairs_name	Internal function argument.
data_type	Internal function argument.
false_pos_count	Internal function argument.
p_wilcox_final	Internal function argument.

`fix_name_fun` *Replace the symbols in variable and covariate names in raw input*

Description

Replace the symbols in variable and covariate names in raw input

Usage

```
fix_name_fun(z)
```

Arguments

z	A character vector. This is the character vector that needs to be changed.

`longdat_cont` *Longitudinal analysis with time as continuous variable*

Description

longdat_cont calculates the p values, effect sizes and discover covariate effects of time variables from longitudinal data.

Usage

```
longdat_cont(
  input,
  data_type,
  test_var,
  variable_col,
  fac_var,
  not_used = NULL,
  adjustMethod = "fdr",
  model_q = 0.1,
```

```
  posthoc_q = 0.05,
  theta_cutoff = 2^20,
  nonzero_count_cutoff1 = 9,
  nonzero_count_cutoff2 = 5,
  verbose = TRUE
)
```

Arguments

input	A data frame with the first column as "Individual" and all the columns of dependent variables (features, e.g. bacteria) at the end of the table. The time variable here should be continuous, if time is discrete, please apply longdat_disc() instead. Please avoid using characters that don't belong to ASCII printable characters for potential covariates names (covariates are any column apart from individual, test_var and dependent variables).
data_type	The data type of the dependent variables (features). Can either be "proportion", "measurement", "count", "binary", "ordinal" or "others". Proportion (or ratio) data range from 0 to 1. Measurement data are continuous and can be measured at finer and finer scale (e.g. weight). Count data consist of discrete non-negative integers resulted from counting. Binary data are the data of sorting things into one of two mutually exclusive categories. Ordinal data consist of ranks. Any data that doesn't belong to the previous categories should be classified as "others".
test_var	The name of the independent variable you are testing for, should be a string (e.g. "Time") identical to its column name and make sure there is no space in it.
variable_col	The column number of the position where the dependent variable columns (features, e.g. bacteria) start in the table.
fac_var	The column numbers of the position where the columns that aren't numerical (e.g. characters, categorical numbers, ordinal numbers). This should be a numerical vector (e.g. c(1, 2, 5:7)).
not_used	The column position of the columns not are irrelevant and can be ignored when in the analysis. This should be a numerical vector, and the default is NULL.
adjustMethod	Multiple testing p value correction. Choices are the ones in p.adjust(), including 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY' and 'fdr.' The default is 'fdr'.
model_q	The threshold for significance of model test after multiple testing correction. The default is 0.1.
posthoc_q	The threshold for significance of post-hoc test after multiple testing correction. The default is 0.05.
theta_cutoff	Required when the data type is set as "count". Variable with theta value from negative binomial regression larger than or equal to the cutoff will be filtered out if it also doesn't meet the non-zero count threshold. Users can use the function "theta_plot()" to help with specifying the value for theta_cutoff. The default is 2^20.
nonzero_count_cutoff1	Required when the data type is set as "count". Variable with non-zero counts lower than or equal to this value will be filtered out if it doesn't meet the theta

threshold either. Users can use the function "theta_plot()" to help with specifying the value for nonzero_count_cutoff1. The default is 9.

`nonzero_count_cutoff2`
: Required when the data type is set as "count". Variable with non-zero counts lower than or equal to this value will be filtered out. Users can use the function "theta_plot()" to help with specifying the value for nonzero_count_cutoff2. The default is 5.

`verbose`
: A boolean vector indicating whether to print detailed message. The default is TRUE.

Details

The brief workflow of longdat_cont() is as below:

When there's no potential covariates in the input data (covariates are anything apart from individual, test_var and dependent variables): First, the model test tests the significance of test_var on dependent variables. Different generalized linear mixed effect models are implemented for different types of dependent variable. Negative binomial mixed model for "count", linear mixed model (dependent variables normalized first) for "measurement", beta mixed model for "proportion", binary logistic mixed model for "binary", and proportional odds logistic mixed model for "ordinal". Then, post-hoc test (Spearman's correlation test) on the model is done. When the data type is "count" mode, a control model test will be run on randomized data (the rows are shuffled). If there are false positive signals in this control model test, users will get a warning at the end of the run.

When there are potential covariates in the input data: After the model test and post-hoc test described above, a covariate model test will be added to the work flow. The potential covariates will be added to the model one by one and test for its significance on each dependent variable. The rest are the same as the description above.

Also, when your data type is count data, please use set.seed() before running longdat_cont() so that you can get reproducible randomized negative check.

Value

longdat_cont() returns a list which contains a "Result_table", and if there are covariates in the input data frame, there will be another table called "Covariate_table". For count mode, if there is any false positive in the randomized control result, then another table named "Randomized_control_table" will also be generated. The detailed description is as below.

Result_table

1. The first column: The dependent variables in the input data. This can be used as row name when being imported into R.

2. Prevalence_percentage: The percentage of each dependent variable present across individuals and time points

3. Mean_abundance: The mean value of each dependent variable across individuals and time points

4. Signal: The final decision of the significance of the test_var (independent variable) on each dependent variable. NS: This represents "Non-significant", which means that there's no effect of time.

OK_nc: This represents "OK and no covariate". There's an effect of time and there's no potential covariate.

OK_d: This represents "OK but doubtful". There's an effect of time and there's no potential covariate, however the confidence interval of the test_var estimate in the model test covers zero, and thus it is doubtful of this signal.

OK_nrc: This represents "OK and not reducible to covariate". There are potential covariates, however there's an effect of time and it is independent of those of covariates.

EC: This represents "Entangled with covariate". There are potential covariates, and it isn't possible to conclude whether the effect is resulted from time or covariates.

RC: This represents "Effect reducible to covariate". There's an effect of time, but it can be reduced to the covariate effects.

5. Effect: This column contains the value of each dependent variable decreases/increases/NS(non-significant) along the time. A positive correlation between with time dependent variable value yields "increase", while a negative correlation yields "decrease". NS means no significant correlation.

6. 'EffectSize': This column reports the correlation coefficient (Spearman's rho) between each dependent variable value and time.

7. Null_time_model_q: This column shows the multiple-comparison-adjusted p values (Wald test) of the significance of test_var in the models.

8. Post-hoc_q: These are the multiple-comparison-adjusted p values from the post-hoc test (Spearman's correlation test) of the model.

Covariate_table

The first column contains the dependent variables in the input data. This can be used as row name when being imported into R. Then every 3 columns are a group. Covariate column shows the covariate's name; Covariate column shows the covariate's name; Covariate_type column shows how effect is affected by covariate ; Effect_size column shows the effect size of dependent variable value between different values of covariate. Due to the different number of covariates for each dependent variable, there may be NAs in the table and they can simply be ignored. If the covariate table is totally empty, this means that there are no covariates detected.

Randomized_control_table (for user's reference)

We assume that there shouldn't be positive results in the randomized control test, because all the rows in the original dataset are shuffled randomly. Therefore, any signal that showed significance here will be regarded as false positive. And if there's false positive in this randomized control result, longdat_disc will warn the user at the end of the run. This Randomized_control table is only generated when there is false positive in the randomized control test. It is intended to be a reference for users to see the effect size of false positive features.

1. The first column "Model_q" shows the multiple-comparison-adjusted p values (Wald test) of the significance of test_var in the negative- binomial models in the randomized dataset. Only the features with Model_q lower than the defined model_q (default = 0.1) will be listed in this table.

2. Signal: This column describes if test_var is significant on each dependent variable based on the post-hoc test p values (Spearman's correlation test). "False positive" indicates that test_var is significant, while "Negative" indicates non-significance.

3. 'Posthoc_q': This column describes the multiple-comparison-adjusted p values from the post-hoc test (Spearman's correlation test) of the model in the randomized control dataset.

4. Effect_size: This column describes the correlation coefficient (Spearman's rho) of each dependent variable between each dependent variable value and time.

Examples

```
test_cont <- suppressWarnings(longdat_cont(input = LongDat_cont_master_table,
data_type = "count", test_var = "Day",
variable_col = 7, fac_var = c(1, 3)))
```

LongDat_cont_feature_table *data/LongDat_cont_feature_table.RData documentation*

Description

Example feature data frame for longdat_cont(). This is a dummy data which contains features (dependent variables).

Usage

```
data(LongDat_cont_feature_table)
```

Format

An object of class `data.frame` with 20 rows and 4 columns.

Examples

```
## Not run:
data(LongDat_cont_feature_table)

## End(Not run)
```

LongDat_cont_master_table *data/LongDat_cont_master_table.RData documentation*

Description

Example master data frame for longdat_cont(). This is a dummy data which contains metadata and features.

Usage

```
data(LongDat_cont_master_table)
```

Format

An object of class `data.frame` with 20 rows and 9 columns.

Examples

```
## Not run:
data(LongDat_cont_master_table)

## End(Not run)
```

LongDat_cont_metadata_table *data/LongDat_cont_metadata_table.RData documentation*

Description

Example metadata data frame for longdat_cont(). This is a dummy data which contains metadata.

Usage

```
data(LongDat_cont_metadata_table)
```

Format

An object of class `data.frame` with 20 rows and 7 columns.

Examples

```
## Not run:
data(LongDat_cont_metadata_table)

## End(Not run)
```

---

longdat_disc *Longitudinal analysis with time as discrete variable*

Description

longdat_disc calculates the p values, effect sizes and discover covariate effects of time variables from longitudinal data.

Usage

```
longdat_disc(
  input,
  data_type,
  test_var,
  variable_col,
  fac_var,
  not_used = NULL,
```

```
  adjustMethod = "fdr",
  model_q = 0.1,
  posthoc_q = 0.05,
  theta_cutoff = 2^20,
  nonzero_count_cutoff1 = 9,
  nonzero_count_cutoff2 = 5,
  verbose = TRUE
)
```

Arguments

input	A data frame with the first column as "Individual" and all the columns of dependent variables (features, e.g. bacteria) at the end of the table. The time variable here should be discrete, if time is continuous, please apply longdat_cont() instead. Please avoid using characters that don't belong to ASCII printable characters for potential covariates names (covariates are any column apart from individual, test_var and dependent variables).
data_type	The data type of the dependent variables (features). Can either be "proportion", "measurement", "count", "binary", "ordinal" or "others". Proportion (or ratio) data range from 0 to 1. Measurement data are continuous and can be measured at finer and finer scale (e.g. weight). Count data consist of discrete non-negative integers resulted from counting. Binary data are the data of sorting things into one of two mutually exclusive categories. Ordinal data consist of ranks. Any data that doesn't belong to the previous categories should be classified as "others".
test_var	The name of the independent variable you are testing for, should be a string (e.g. "Time") identical to its column name and make sure there is no space in it.
variable_col	The column number of the position where the dependent variable columns (features, e.g. bacteria) start in the table.
fac_var	The column numbers of the position where the columns that aren't numerical (e.g. characters, categorical numbers, ordinal numbers). This should be a numerical vector (e.g. c(1, 2, 5:7)).
not_used	The column position of the columns not are irrelevant and can be ignored when in the analysis. This should be a numerical vector, and the default is NULL.
adjustMethod	Multiple testing p value correction. Choices are the ones in p.adjust(), including 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY' and 'fdr'. The default is 'fdr'.
model_q	The threshold for significance of model test after multiple testing correction. The default is 0.1.
posthoc_q	The threshold for significance of post-hoc test of the model after multiple testing correction. The default is 0.05.
theta_cutoff	Required when the data type is set as "count". Variable with theta value from negative binomial regression larger than or equal to the cutoff will be filtered out if it also doesn't meet the non-zero count threshold. Users can use the function "theta_plot()" to help with specifying the value for theta_cutoff. The default is 2^20.

`nonzero_count_cutoff1`
: Required when the data type is set as "count". Variable with non-zero counts lower than or equal to this value will be filtered out if it doesn't meet the theta threshold either. Users can use the function "theta_plot()" to help with specifying the value for nonzero_count_cutoff1. The default is 9.

`nonzero_count_cutoff2`
: Required when the data type is set as "count". Variable with non-zero counts lower than or equal to this value will be filtered out. Users can use the function "theta_plot()" to help with specifying the value for nonzero_count_cutoff2. The default is 5.

`verbose`
: A boolean vector indicating whether to print detailed message. The default is TRUE.

Details

The brief workflow of longdat_disc() is as below:

When there's no potential covariates in the input data (covariates are anything apart from individual, test_var and dependent variables): First, the model test tests the significance of test_var on dependent variables. Different generalized linear mixed effect models are implemented for different types of dependent variable. Negative binomial mixed model for "count", linear mixed model (dependent variables normalized first) for "measurement", beta mixed model for "proportion", binary logistic mixed model for "binary", and proportional odds logistic mixed model for "ordinal". Then, post-hoc test ('emmeans') on the model is done. When the data type is "count" mode, a control model test will be run on randomized data (the rows are shuffled). If there are false positive signals in this control model test, then additional Wilcoxon post-hoc test will be done because it is more conservative.

When there are potential covariates in the input data: After the model test and post-hoc test described above, a covariate model test will be added to the work flow. The potential covariates will be added to the model one by one and test for its significance on each dependent variable. The rest are the same as the description above.

Also, when your data type is count data, please use set.seed() before running longdat_disc() so that you can get reproducible randomized negative check.

Value

longdat_disc() returns a list which contains a "Result_table", and if there are covariates in the input data frame, there will be another table called "Covariate_table". For count mode, if there is any false positive in the randomized control result, then another table named "Randomized_control_table" will also be generated. The detailed description is as below.

Result_table

1. The first column: The dependent variables in the input data. This can be used as row name when being imported into R.

2. Prevalence_percentage: The percentage of each dependent variable present across individuals and time points.

3. Mean_abundance: The mean value of each dependent variable across individuals and time points.

4. Signal: The final decision of the significance of the test_var (independent variable) on each dependent variable. NS: This represents "Non-significant", which means that there's no effect of time.

OK_nc: This represents "OK and no covariate". There's an effect of time and there's no potential covariate.

OK_d: This represents "OK but doubtful". There's an effect of time and there's no potential covariate, however the confidence interval of the test_var estimate in the model test covers zero, and thus it is doubtful of this signal.

OK_nrc: This represents "OK and not reducible to covariate". There are potential covariates, however there's an effect of time and it is independent of those of covariates.

EC: This represents "Entangled with covariate". There are potential covariates, and it isn't possible to conclude whether the effect is resulted from time or covariates.

RC: This represents "Effect reducible to covariate". There's an effect of time, but it can be reduced to the covariate effects.

5. 'Effect_a_b': The "a" and "b" here are the names of the time points. These columns describe the value of each dependent variable decreases/increases/NS(non-significant) at time point b comparing with time point a. The number of Effect columns depends on how many combinations of time points in the input data.

6. 'EffectSize_a_b': The "a" and "b" here are the names of the time points. These columns describe the effect size (Cliff's delta) of each dependent variable between time point b and a. The number of 'EffectSize' columns depends on how many combinations of time points in the input data.

7. 'Null_time_model_q': This column shows the multiple-comparison-adjusted p values (Wald test) of the significance of test_var in the models.

8. 'Post-hoc_q_a_b': The "a" and "b" here are the names of the time points. These are the multiple-comparison-adjusted p values from the post-hoc test of the model. The number of Post-hoc_q columns depends on how many combinations of time points in the input data.

9. 'Wilcox_p_a_b': The "a" and "b" here are the names of the time points. These columns only appear when data type is "count" and there exist false positives in the model test on randomized data. Wilcoxon test are more conservative than the default post-hoc test ('emmeans'), and thus it is a good reference for getting a more conservative result of the significant outcomes.

Covariate_table

The first column contains the dependent variables in the input data. This can be used as row name when being imported into R. Then every 3 columns are a group. Covariate column shows the covariate's name; Covariate_type column shows how effect is affected by covariate; Effect_size column shows the effect size of dependent variable value between different values of covariate. Due to the different number of covariates for each dependent variable, there may be NAs in the table and they can simply be ignored. If the covariate table is totally empty, this means that there are no covariates detected.

Randomized_control_table (for user's reference)

We assume that there shouldn't be positive results in the randomized control test, because all the rows in the original dataset are shuffled randomly. Therefore, any signal that showed significance here will be regarded as false positive. And if there's false positive in this randomized control result, longdat_disc() will warn the user at the end of the run. This Randomized_control table is

only generated when there is false positive in the randomized control test. It is intended to be a reference for users to see the effect size of false positive features.

1. "Model_q": It shows the multiple-comparison-adjusted p values (Wald test)of the significance of test_var in the negative-binomial models in the randomized dataset. Only the features with Model_q lower than the defined model_q (default = 0.1) will be listed in this table.

2. Final_signal: It show the overall signal being either false positive or negative. "False positive" indicates that test_var is significant, while "Negative" indicates non-significance.

3. 'Signal_a_b': The "a" and "b" here are the names of the time points. These columns describe if test_var is significant on each dependent variable between each time point based on the post-hoc test p values (listed right to Signal_a_b). "False positive" indicates that test_var is significant, while "Negative" indicates non-significance. The number of Signal_a_b columns depends on how many combinations of time points in the input data.

4. 'Posthoc_q_a_b': The "a" and "b" here are the names of the time points. These columns describe the multiple-comparison-adjusted p values from the post-hoc test of the model between time point b and a in the randomized control dataset. The number of 'Posthoc_q_a_b' columns depends on how many combinations of time points in the input data.

5. 'Effect_size_a_b': The "a" and "b" here are the names of the time points. These columns describe the effect size (Cliff's delta) of each dependent variable between time point b and a in the randomized control dataset. The number of Effect_size_a_b columns depends on how many combinations of time points in the input data.

Examples

```
test_disc <- longdat_disc(input = LongDat_disc_master_table,
data_type = "count", test_var = "Time_point",
variable_col = 7, fac_var = c(1:3))
```

`LongDat_disc_feature_table` *data/LongDat_disc_feature_table.RData documentation*

Description

Example feature data frame for longdat_disc(). This is a dummy data which contains features (dependent variables).

Usage

```
data(LongDat_disc_feature_table)
```

Format

An object of class `data.frame` with 30 rows and 4 columns.

Examples

```
## Not run:
data(LongDat_disc_feature_table)

## End(Not run)
```

LongDat_disc_master_table
data/LongDat_disc_master_table.RData documentation

Description

Example master data frame for longdat_disc(). This is a dummy data which contains metadata and features.

Usage

```
data(LongDat_disc_master_table)
```

Format

An object of class `data.frame` with 30 rows and 9 columns.

Examples

```
## Not run:
data(LongDat_disc_master_table)

## End(Not run)
```

LongDat_disc_metadata_table
data/LongDat_disc_metadata_table.RData documentation

Description

Example metadata data frame for longdat_disc(). This is a dummy data which contains metadata.

Usage

```
data(LongDat_disc_metadata_table)
```

Format

An object of class `data.frame` with 30 rows and 7 columns.

Examples

```
## Not run:
data(LongDat_disc_metadata_table)

## End(Not run)
```

`make_master_table` *Create input master table from metadata and feature tables for longdat_disc() and longdat_cont()*

Description

Create input master table from metadata and feature tables for longdat_disc() and longdat_cont()

Usage

```
make_master_table(
  metadata_table,
  feature_table,
  sample_ID,
  individual,
  keep_id = FALSE
)
```

Arguments

`metadata_table`	A data frame whose columns consist of sample identifiers (sample_ID), individual, time point and other meta data. Each row corresponds to one sample_ID. Metadata table should have the same number of rows as feature table does. Please avoid using characters that don't belong to ASCII printable characters for the column names.
`feature_table`	A data frame whose columns only consist of sample identifiers (sample_ID) and features (dependent variables, e.g. microbiome). Each row corresponds to one sample_ID. Please do not include any columns other than sample_ID and features. Please avoid using characters that don't belong to ASCII printable characters for the column names. Also, feature table should have the same number of rows as metadata table does.
`sample_ID`	The name of the column which stores sample identifiers. Please make sure that sample_IDs are unique for each sample, and that metadata and feature tables have the same sample_IDs. If sample_IDs don't match between the two tables, it will fail to join them together. This should be a string, e.g. "Sample_ID"
`individual`	The name of the column which stores individual information in the metadata table. This should be a string, e.g. "Individual"
`keep_id`	A boolean vector indicating whether keep sample_ID column in the output master table. The default is FALSE.

Details

This function joins metadata and feature tables by the sample_ID column. Users can create master tables compatible with the format of longdat_disc() and longdat_cont() input easily.This function outputs a master table with individual as the first column, followed by time point and other metadata, and then by feature columns.

Value

a data frame which complies with the required format of an input data frame for longdat_disc() and longdat_cont().

Examples

```
test_master <- make_master_table(
metadata_table = LongDat_disc_metadata_table,
feature_table = LongDat_disc_feature_table,
sample_ID = "Sample_ID",
individual = "Individual")
```

NuModelTest_cont *Null Model Test and post-hoc Test in longdat_cont() pipeline*

Description

Null Model Test and post-hoc Test in longdat_cont() pipeline

Arguments

N	Internal function argument.
data_type	Internal function argument.
test_var	Internal function argument.
melt_data	Internal function argument.
variables	Internal function argument.
verbose	Internal function argument.

NuModelTest_disc *Null Model Test and post-hoc Test in longdat_disc() pipeline*

Description

Null Model Test and post-hoc Test in longdat_disc() pipeline

Arguments

N	Internal function argument.
data_type	Internal function argument.
test_var	Internal function argument.
melt_data	Internal function argument.
variables	Internal function argument.
verbose	Internal function argument.

random_neg_ctrl_cont *Randomized negative control for count data in longdat_cont()*

Description

Randomized negative control for count data in longdat_cont()

Arguments

test_var	Internal function argument.
variable_col	Internal function argument.
fac_var	Internal function argument.
not_used	Internal function argument.
factors	Internal function argument.
data	Internal function argument.
N	Internal function argument.
data_type	Internal function argument.
variables	Internal function argument.
adjustMethod	Internal function argument.
model_q	Internal function argument.
posthoc_q	Internal function argument.
theta_cutoff	Internal function argument.
nonzero_count_cutoff1	Internal function argument.
nonzero_count_cutoff2	Internal function argument.
verbose	Internal function argument.

`random_neg_ctrl_disc` *Randomized negative control for count data in longdat_disc()*

Description

Randomized negative control for count data in longdat_disc()

Arguments

`test_var`	Internal function argument.
`variable_col`	Internal function argument.
`fac_var`	Internal function argument.
`not_used`	Internal function argument.
`factors`	Internal function argument.
`data`	Internal function argument.
`N`	Internal function argument.
`data_type`	Internal function argument.
`variables`	Internal function argument.
`case_pairs`	Internal function argument.
`adjustMethod`	Internal function argument.
`model_q`	Internal function argument.
`posthoc_q`	Internal function argument.
`theta_cutoff`	Internal function argument.
`nonzero_count_cutoff1`	Internal function argument.
`nonzero_count_cutoff2`	Internal function argument.
`verbose`	Internal function argument.

`rm_sparse_cont` *Remove the dependent variables that are below the threshold of sparsity when the data type is count data in longdat_cont()*

Description

Remove the dependent variables that are below the threshold of sparsity when the data type is count data in longdat_cont()

Arguments

values	Internal function argument.
data	Internal function argument.
nonzero_count_cutoff1	Internal function argument.
nonzero_count_cutoff2	Internal function argument.
theta_cutoff	Internal function argument.
Ps_null_model	Internal function argument.
prevalence	Internal function argument.
absolute_sparsity	Internal function argument.
mean_abundance	Internal function argument.
p_poho	Internal function argument.
assoc	Internal function argument.

`rm_sparse_disc` *Remove the dependent variables that are below the threshold of sparsity when the data type is count data in longdat_disc()*

Description

Remove the dependent variables that are below the threshold of sparsity when the data type is count data in longdat_disc()

Arguments

values	Internal function argument.
data	Internal function argument.
nonzero_count_cutoff1	Internal function argument.
nonzero_count_cutoff2	Internal function argument.
theta_cutoff	Internal function argument.
Ps_null_model	Internal function argument.
prevalence	Internal function argument.
absolute_sparsity	Internal function argument.
mean_abundance	Internal function argument.
Ps_poho_fdr	Internal function argument.
delta	Internal function argument.

theta_plot *Plot theta values of negative binomial models versus non-zero count for count data*

Description

Plot theta values of negative binomial models versus non-zero count for count data

Arguments

input	A data frame with the first column as "Individual" and all the columns of dependent variables (features, e.g. bacteria) at the end of the table.
test_var	The name of the independent variable you are testing for, should be a character vector (e.g. c("Time")) identical to its column name and make sure there is no space in it.
variable_col	The column number of the position where the dependent variable columns (e.g. bacteria) start in the table
fac_var	The column numbers of the position where the columns that aren't numerical (e.g. characters, categorical numbers, ordinal numbers), should be a numerical vector (e.g. c(1, 2, 5:7))
not_used	The column position of the columns not are irrelevant and can be ignored when in the analysis. This should be a number vector, and the default is NULL.
point_size	The point size for plotting in 'ggplot2'. The default is 1.
x_interval_value	The interval value for tick marks on x-axis. The default is 5.
y_interval_value	The interval value for tick marks on y-axis. The default is 5.
verbose	A boolean vector indicating whether to print detailed message. The default is TRUE.

Details

This function outputs a plot that facilitates the setting of theta_cutoff in longdat_disc() and longdat_cont(). This only applies when the dependent variables are count data. longdat_disc() and longdat_cont() implements negative binomial (NB) model for count data, and if the theta (dispersion parameter) of NB model gets too high, then the p value of it will be extremely low regardless of whether there is real significance or not. Therefore, the highest threshold of theta value is set and any variable beyond the threshold will be excluded from the test. The default value of theta_cutoff is set to 2^20 from the observation that 2^20 is a clear cutoff line for several datasets. Users can change theta_cutoff value to fit their own data.

Value

a 'ggplot' object

Examples

```
test_theta_plot <- theta_plot(input = LongDat_disc_master_table,
 test_var = "Time_point", variable_col = 7, fac_var = c(1:3))
```

`unlist_table` *Unlist confound (covariate) and inverse confound (covariate) tables, turn them into tables*

Description

Unlist confound (covariate) and inverse confound (covariate) tables, turn them into tables

Usage

```
unlist_table(x, N, variables)
```

Arguments

`x`	The list to be unlisted and turned into table
`N`	Internal function argument.
`variables`	Internal function argument.

`wilcox_posthoc` *Wilcoxon post-hoc test*

Description

Wilcoxon post-hoc test

Usage

```
wilcox_posthoc(
  result_neg_ctrl,
  model_q,
  melt_data,
  test_var,
  variables,
  data,
  N,
  verbose
)
```

Arguments

result_neg_ctrl	Internal function argument.
model_q	Internal function argument.
melt_data	Internal function argument.
test_var	Internal function argument.
variables	Internal function argument.
data	Internal function argument.
N	Internal function argument.
verbose	Internal function argument.

Index