Package: CB2 (via r-universe)

October 28, 2024

Type Package

Title CRISPR Pooled Screen Analysis using Beta-Binomial Test

Version 1.3.4

Date 2020-07-23

Description Provides functions for hit gene identification and quantification of sgRNA (single-guided RNA) abundances for CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) pooled screen data analysis. Details are in Jeong et al. (2019) <doi:10.1101/gr.245571.118> and Baggerly et al. (2003) <doi:10.1093/bioinformatics/btg173>.

Depends R (>= 3.5.0)

License MIT + file LICENSE

LazyData true

Imports Rcpp (>= 0.12.16), metap, magrittr, dplyr, tibble, stringr, ggplot2, tidyr, glue, pheatmap, tools, readr, parallel, R.utils

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, knitr, rmarkdown

RoxygenNote 7.1.1

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation yes

Author Hyun-Hwan Jeong [aut, cre]

Maintainer Hyun-Hwan Jeong <jeong.hyunhwan@gmail.com>

Repository CRAN

Date/Publication 2020-07-24 09:42:24 UTC

Contents

calc_mappability 2
Evers_CRISPRn_RT112 3

fit_ab . . . 3
get_CPM . . . 4
join_count_and_design . . . 5
measure_gene_stats . . . 5
measure_sgrna_stats . . . 6
plot_corr_heatmap . . . 8
plot_count_distribution . . . 8
plot_dotplot . . . 9
plot_PCA . . . 10
quant . . . 10
run_estimation . . . 11
run_sgrna_quant . . . 12
Sanson_CRISPRn_A375 . . . 13

Index **14**

`calc_mappability` *A function to calculate the mappabilities of each NGS sample.*

Description

A function to calculate the mappabilities of each NGS sample.

Usage

```
calc_mappability(count_obj, df_design)
```

Arguments

`count_obj`	A list object is created by ‘run_sgrna_quant‘.
`df_design`	The table contains a study design.

Examples

```
library(CB2)
library(magrittr)
library(tibble)
library(dplyr)
library(glue)
FASTA <- system.file("extdata", "toydata", "small_sample.fasta", package = "CB2")
ex_path <- system.file("extdata", "toydata", package = "CB2")

df_design <- tribble(
  ~group, ~sample_name,
  "Base", "Base1",
  "Base", "Base2",
  "High", "High1",
  "High", "High2") %>%
    mutate(fastq_path = glue("{ex_path}/{sample_name}.fastq"))
```

```
cb2_count <- run_sgrna_quant(FASTA, df_design)
calc_mappability(cb2_count, df_design)
```

`Evers_CRISPRn_RT112` *A benchmark CRISPRn pooled screen data from Evers et al.*

Description

A benchmark CRISPRn pooled screen data from Evers et al.

Usage

```
data(Evers_CRISPRn_RT112)
```

Format

The data object is a list and contains below information:

count The count matrix from Evers et al.'s paper and contains the CRISPRn screening result using RT112 cell-line. It contains three different replicates for T0 (before) and contains different three replicates for T1 (after).

egenes The list of 46 essential genes used in Evers et al.'s study.

ngenes The list of 47 non-essential genes used in Evers et al.'s study.

design The data.frame contains study design.

sg_stat The data.frame contains the sgRNA-level statistics.

gene_stat The data.frame contains the gene-level statistics.

Source

https://www.ncbi.nlm.nih.gov/pubmed/27111720

`fit_ab` *A C++ function to perform a parameter estimation for the sgRNA-level test. It will estimate two different parameters 'phat' and 'vhat,' and we assume input count data follows the beta-binomial distribution. Dr. Keith Baggerly initially implemented this code in Matlab, and it has been rewritten it in C++ for the speed-up.*

Description

A C++ function to perform a parameter estimation for the sgRNA-level test. It will estimate two different parameters 'phat' and 'vhat,' and we assume input count data follows the beta-binomial distribution. Dr. Keith Baggerly initially implemented this code in Matlab, and it has been rewritten it in C++ for the speed-up.

Usage

```
fit_ab(xvec, nvec)
```

Arguments

xvec	a matrix contains sgRNA read counts.
nvec	a vector contains the library size.

`get_CPM` *A function to normalize sgRNA read counts.*

Description

A function to normalize sgRNA read counts.

Usage

```
get_CPM(sgcount)
```

Arguments

sgcount	The input table contains read counts of sgRNAs for each sample A function to calculate the CPM (Counts Per Million) (required)

Value

a normalized CPM table will be returned

Examples

```
library(CB2)
data(Evers_CRISPRn_RT112)
get_CPM(Evers_CRISPRn_RT112$count)
```

join_count_and_design *A function to join a count table and a design table.*

Description

A function to join a count table and a design table.

Usage

```
join_count_and_design(sgcount, df_design)
```

Arguments

sgcount	The input matrix contains read counts of sgRNAs for each sample.
df_design	The table contains a study design.

Value

A tall-thin and combined table of the sgRNA read counts and study design will be returned.

Examples

```
library(CB2)
data(Evers_CRISPRn_RT112)
head(join_count_and_design(Evers_CRISPRn_RT112$count, Evers_CRISPRn_RT112$design))
```

measure_gene_stats *A function to perform gene-level test using a sgRNA-level statistics.*

Description

A function to perform gene-level test using a sgRNA-level statistics.

Usage

```
measure_gene_stats(sgrna_stat, logFC_level = "sgRNA")
```

Arguments

sgrna_stat	A data frame created by 'measure_sgrna_stats'
logFC_level	The level of 'logFC' value. It can be 'gene' or 'sgRNA'.

Value

A table contains the gene-level test result, and the table contains these columns:

- 'gene': Theg gene name to be tested.
- 'n_sgrna': The number of sgRNA targets the gene in the library.
- 'cpm_a': The mean of CPM of sgRNAs within the first group.
- 'cpm_b': The mean of CPM of sgRNAs within the second group.
- 'logFC': The log fold change of the gene between two groups. Taking the mean of sgRNA 'logFC's is default, and 'logFC' is calculated by 'log2(cpm_b+1) - log2(cpm_a+1)' if 'logFC_level' parameter is set to 'gene'.
- 'p_ts': The p-value indicates a difference between the two groups at the gene-level.
- 'p_pa': The p-value indicates enrichment of the first group at the gene-level.
- 'p_pb': The p-value indicates enrichment of the second group at the gene-level.
- 'fdr_ts': The adjusted P-value of 'p_ts'.
- 'fdr_pa': The adjusted P-value of 'p_pa'.
- 'fdr_pb': The adjusted P-value of 'p_pb'.

Examples

```
data(Evers_CRISPRn_RT112)
measure_gene_stats(Evers_CRISPRn_RT112$sg_stat)
```

---

measure_sgrna_stats *A function to perform a statistical test at a sgRNA-level*

Description

A function to perform a statistical test at a sgRNA-level

Usage

```
measure_sgrna_stats(
  sgcount,
  design,
  group_a,
  group_b,
  delim = "_",
  ge_id = NULL,
  sg_id = NULL
)
```

Arguments

sgcount	This data frame contains read counts of sgRNAs for the samples.
design	This table contains study design. It has to contain 'group.'
group_a	The first group to be tested.
group_b	The second group to be tested.
delim	The delimiter between a gene name and a sgRNA ID. It will be used if only rownames contains sgRNA ID.
ge_id	The column name of the gene column.
sg_id	The column/columns of sgRNA identifiers.

Value

A table contains the sgRNA-level test result, and the table contains these columns:

- 'sgRNA': The sgRNA identifier.
- 'gene': The gene is the target of the sgRNA
- 'n_a': The number of replicates of the first group.
- 'n_b': The number of replicates of the second group.
- 'phat_a': The proportion value of the sgRNA for the first group.
- 'phat_b': The proportion value of the sgRNA for the second group.
- 'vhat_a': The variance of the sgRNA for the first group.
- 'vhat_b': The variance of the sgRNA for the second group.
- 'cpm_a': The mean CPM of the sgRNA within the first group.
- 'cpm_b': The mean CPM of the sgRNA within the second group.
- 'logFC': The log fold change of sgRNA between two groups.
- 't_value': The value for the t-statistics.
- 'df': The value of the degree of freedom, and will be used to calculate the p-value of the sgRNA.
- 'p_ts': The p-value indicates a difference between the two groups.
- 'p_pa': The p-value indicates enrichment of the first group.
- 'p_pb': The p-value indicates enrichment of the second group.
- 'fdr_ts': The adjusted P-value of 'p_ts'.
- 'fdr_pa': The adjusted P-value of 'p_pa'.
- 'fdr_pb': The adjusted P-value of 'p_pb'.

Examples

```
library(CB2)
data(Evers_CRISPRn_RT112)
measure_sgrna_stats(Evers_CRISPRn_RT112$count, Evers_CRISPRn_RT112$design, "before", "after")
```

`plot_corr_heatmap` *A function to show a heatmap sgRNA-level corrleations of the NGS samples.*

Description

A function to show a heatmap sgRNA-level corrleations of the NGS samples.

Usage

```
plot_corr_heatmap(sgcount, df_design, cor_method = "pearson")
```

Arguments

`sgcount`	The input matrix contains read counts of sgRNAs for each sample.
`df_design`	The table contains a study design.
`cor_method`	A string parameter of the correlation measure. One of the three - "pearson", "kendall", or "spearman" will be the string.

Value

A pheatmap object contains the correlation heatmap

library(CB2) data(Evers_CRISPRn_RT112) plot_corr_heatmap(Evers_CRISPRn_RT112$count, Evers_CRISPRn_RT112$design)

`plot_count_distribution` *A function to plot read count distribution.*

Description

A function to plot read count distribution.

Usage

```
plot_count_distribution(sgcount, df_design, add_dots = FALSE)
```

Arguments

`sgcount`	The input matrix contains read counts of sgRNAs for each sample.
`df_design`	The table contains a study design.
`add_dots`	The function will display dots of sgRNA counts if it is set to ‘TRUE‘.

Value

A ggplot2 object contains a read count distribution plot for ‘sgcount‘.

Examples

```
library(CB2)
data(Evers_CRISPRn_RT112)
cpm <- get_CPM(Evers_CRISPRn_RT112$count)
plot_count_distribution(cpm, Evers_CRISPRn_RT112$design)
```

---

plot_dotplot *A function to visualize dot plots for a gene.*

Description

A function to visualize dot plots for a gene.

Usage

```
plot_dotplot(sgcount, df_design, gene, ge_id = NULL, sg_id = NULL)
```

Arguments

sgcount	The input matrix contains read counts of sgRNAs for each sample.
df_design	The table contains a study design.
gene	The gene to be shown.
ge_id	A name of the column contains gene names.
sg_id	A name of the column contains sgRNA IDs.

Value

A ggplot2 object contains dot plots of sgRNA read counts for a gene.

Examples

```
library(CB2)
data(Evers_CRISPRn_RT112)
plot_dotplot(get_CPM(Evers_CRISPRn_RT112$count), Evers_CRISPRn_RT112$design, "RPS7")
```

plot_PCA *A function to plot the first two principal components of samples.*

Description

This function will perform a principal component analysis, and it returns a ggplot object of the PCA plot.

Usage

```
plot_PCA(sgcount, df_design)
```

Arguments

sgcount	The input matrix contains read counts of sgRNAs for each sample.
df_design	The table contains a study design.

Value

A ggplot2 object contains a PCA plot for the input.

library(CB2) data(Evers_CRISPRn_RT112) plot_PCA(Evers_CRISPRn_RT112$count, Evers_CRISPRn_RT112$design)

quant *A C++ function to quantify sgRNA abundance from NGS samples.*

Description

A C++ function to quantify sgRNA abundance from NGS samples.

Usage

```
quant(ref_path, fastq_path, verbose = FALSE)
```

Arguments

ref_path	the path of the annotation file and it has to be a FASTA formatted file.
fastq_path	a list of the FASTQ files.
verbose	Display some logs during the quantification if it is set to ‘true‘.

run_estimation *A function to perform a statistical test at a sgRNA-level, deprecated.*

Description

A function to perform a statistical test at a sgRNA-level, deprecated.

Usage

```
run_estimation(
  sgcount,
  design,
  group_a,
  group_b,
  delim = "_",
  ge_id = NULL,
  sg_id = NULL
)
```

Arguments

sgcount	This data frame contains read counts of sgRNAs for the samples.
design	This table contains study design. It has to contain 'group.'
group_a	The first group to be tested.
group_b	The second group to be tested.
delim	The delimiter between a gene name and a sgRNA ID. It will be used if only rownames contains sgRNA ID.
ge_id	The column name of the gene column.
sg_id	The column/columns of sgRNA identifiers.

Value

A table contains the sgRNA-level test result, and the table contains these columns:

- 'sgRNA': The sgRNA identifier.
- 'gene': The gene is the target of the sgRNA
- 'n_a': The number of replicates of the first group.
- 'n_b': The number of replicates of the second group.
- 'phat_a': The proportion value of the sgRNA for the first group.
- 'phat_b': The proportion value of the sgRNA for the second group.
- 'vhat_a': The variance of the sgRNA for the first group.
- 'vhat_b': The variance of the sgRNA for the second group.
- 'cpm_a': The mean CPM of the sgRNA within the first group.

- 'cpm_b': The mean CPM of the sgRNA within the second group.
- 'logFC': The log fold change of sgRNA between two groups.
- 't_value': The value for the t-statistics.
- 'df': The value of the degree of freedom, and will be used to calculate the p-value of the sgRNA.
- 'p_ts': The p-value indicates a difference between the two groups.
- 'p_pa': The p-value indicates enrichment of the first group.
- 'p_pb': The p-value indicates enrichment of the second group.
- 'fdr_ts': The adjusted P-value of 'p_ts'.
- 'fdr_pa': The adjusted P-value of 'p_pa'.
- 'fdr_pb': The adjusted P-value of 'p_pb'.

`run_sgrna_quant` *A function to run a sgRNA quantification algorithm from NGS sample*

Description

A function to run a sgRNA quantification algorithm from NGS sample

Usage

```
run_sgrna_quant(lib_path, design, map_path = NULL, ncores = 1, verbose = FALSE)
```

Arguments

`lib_path`	The path of the FASTA file.
`design`	A table contains the study design. It must contain 'fastq_path' and 'sample_name.'
`map_path`	The path of file contains gene-sgRNA mapping.
`ncores`	The number that indicates how many processors will be used with a parallelization. The parallelization will be enabled if users do not set the parameter as '-1" (it means the full physical cores will be used) or greater than '1'.
`verbose`	Display some logs during the quantification if it is set to 'TRUE'

Value

It will return a list, and the list contains three elements. The first element ('count') is a data frame contains the result of the quantification for each sample. The second element ('total') is a numeric vector contains the total number of reads of each sample. The last element ('sequence') a data frame contains the sequence of each sgRNA in the library.

Examples

```
library(CB2)
library(magrittr)
library(tibble)
library(dplyr)
library(glue)
FASTA <- system.file("extdata", "toydata", "small_sample.fasta", package = "CB2")
ex_path <- system.file("extdata", "toydata", package = "CB2")

df_design <- tribble(
  ~group, ~sample_name,
  "Base", "Base1",
  "Base", "Base2",
  "High", "High1",
  "High", "High2") %>%
    mutate(fastq_path = glue("{ex_path}/{sample_name}.fastq"))

cb2_count <- run_sgrna_quant(FASTA, df_design)
```

Sanson_CRISPRn_A375 *A benchmark CRISPRn pooled screen data from Sanson et al.*

Description

A benchmark CRISPRn pooled screen data from Sanson et al.

Usage

```
data(Sanson_CRISPRn_A375)
```

Format

The data object is a list and contains below information:

count The count matrix from Sanson et al.'s paper and contains the CRISPRn screening result using A375 cell-line. It contains a sample of plasimd, and three biological replicates after three weeks.

egenes The list of 1,580 essential genes used in Sanson et al.'s study.

ngenes The list of 927 non-essential genes used in Sanson et al.'s study.

design The data.frame contains study design.

Source

https://www.ncbi.nlm.nih.gov/pubmed/30575746

Index

∗ **datasets**
Evers_CRISPRn_RT112, 3
Sanson_CRISPRn_A375, 13

calc_mappability, 2

Evers_CRISPRn_RT112, 3

fit_ab, 3

get_CPM, 4

join_count_and_design, 5

measure_gene_stats, 5
measure_sgrna_stats, 6

plot_corr_heatmap, 8
plot_count_distribution, 8
plot_dotplot, 9
plot_PCA, 10

quant, 10

run_estimation, 11
run_sgrna_quant, 12

Sanson_CRISPRn_A375, 13