

# Package ‘CB2’

February 19, 2019

**Type** Package

**Title** CRISPR Pooled Screen Analysis using Beta-Binomial Test

**Version** 1.1

**Date** 2019-02-18

**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. (2018) <doi:10.1101/309302> and Baggerly et al. (2003) <doi:10.1093/bioinformatics/btg173>.

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Imports** Rcpp (>= 0.12.16), metap, magrittr, dplyr, tibble, stringr, ggplot2, tidyr, glue, pheatmap

**LinkingTo** Rcpp, RcppArmadillo

**Suggests** testthat, knitr, rmarkdown

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Hyun-Hwan Jeong [aut, cre]

**Maintainer** Hyun-Hwan Jeong <hyun-hwan.jeong@bcm.edu>

**Repository** CRAN

**Date/Publication** 2019-02-19 08:20:08 UTC

## R topics documented:

calc_mappability . . . . .	2
Evers_CRISPRn_RT112 . . . . .	3
fit_ab . . . . .	3
get_CPM . . . . .	4

join_count_and_design . . . . .	4
measure_gene_stats . . . . .	5
plot_corr_heatmap . . . . .	6
plot_count_distribution . . . . .	6
plot_dotplot . . . . .	7
plot_PCA . . . . .	8
quant . . . . .	8
run_estimation . . . . .	9
run_sgrna_quant . . . . .	10
Sanson_CRISPRn_A375 . . . . .	11

## Index 12

---

calc_mappability	<i>A function to calculate the mappabilities of each NGS sample.</i>
------------------	--

---

### 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**

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).

**genes** 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.

---

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)
```

**Arguments**

sgrna\_stat      A data frame created by 'run\_estimation'

**Value**

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

- 'gene': The 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.
- '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)
```

---

plot_corr_heatmap	<i>A function to show a heatmap sgRNA-level correlations of the NGS samples.</i>
-------------------	--

---

**Description**

A function to show a heatmap sgRNA-level correlations 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 heatmap 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	<i>A function to plot read count distribution.</i>
-------------------------	--

---

**Description**

A function to plot read count distribution.

**Usage**

```
plot_count_distribution(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 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	<i>A function to visualize dot plots for a gene.</i>
--------------	--

---

**Description**

A function to visualize dot plots for a gene.

**Usage**

```
plot_dotplot(sgcount, df_design, gene)
```

**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.

**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	<i>A function to plot the first two principal components of samples.</i>
----------	--

---

**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	<i>A C++ function to quantify sgRNA abundance from NGS samples.</i>
-------	---

---

**Description**

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

**Usage**

```
quant(ref_path, fastq_path)
```

**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.



---

run_estimation	<i>A function to perform a statistical test at a sgRNA-level</i>
----------------	--

---

### Description

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

### Usage

```
run_estimation(sgcount, design, group_a, group_b)
```

### 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.

### 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)
run_estimation(Evers_CRISPRn_RT112$count, Evers_CRISPRn_RT112$design, "before", "after")
```

---

run_sgrna_quant	<i>A function to run a sgRNA quantification algorithm from NGS sample</i>
-----------------	---

---

**Description**

A function to run a sgRNA quantification algorithm from NGS sample

**Usage**

```
run_sgrna_quant(lib_path, design)
```

**Arguments**

lib_path	The path of the annotation file.
design	A table contains the study design. It must contain ‘fastq_path’ and ‘sample_name.’

**Value**

It will return a list, and the list contains two 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.

**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**

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

## \*Topic **datasets**

Evers\_CRISPRn\_RT112, [3](#)

Sanson\_CRISPRn\_A375, [11](#)

calc\_mappability, [2](#)

Evers\_CRISPRn\_RT112, [3](#)

fit\_ab, [3](#)

get\_CPM, [4](#)

join\_count\_and\_design, [4](#)

measure\_gene\_stats, [5](#)

plot\_corr\_heatmap, [6](#)

plot\_count\_distribution, [6](#)

plot\_dotplot, [7](#)

plot\_PCA, [8](#)

quant, [8](#)

run\_estimation, [9](#)

run\_sgrna\_quant, [10](#)

Sanson\_CRISPRn\_A375, [11](#)