

Package ‘TraceQC’

April 16, 2022

Type Package

Title Quality Control of CRISPR Lineage Tracing Sequence Data

Version 1.0.1

Date 2021-10-3

Description Provides a generalized quantify control functions and data processing pipeline across several CRISPR-based lineage tracing platforms that can be directly used in data analysis.

Depends R (>= 3.5.0)

Imports circlize, tidyr, RColorBrewer, ggplot2, readr, purrr, tictoc, magrittr, dplyr, stringr, grid, rmarkdown, tibble, ComplexHeatmap

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

NeedsCompilation no

Author Jingyuan Hu [aut, ctb, cph],
Hyun-Hwan Jeong [aut, cre, ctb],
Rami Al-Ouran [ctb]

Maintainer Jingyuan Hu <jingyuan.hu@bcm.edu>

VignetteBuilder knitr

Suggests knitr,
cowplot,
testthat

R topics documented:

circular_chordgram	2
circular_histogram	3
filter_mutations	3
filter_mutations_per_cell	4
filter_mutations_per_UMI	4
find_position	5
format_mutation_df	5
get_abspath	6
get_read_count_per_UB	6
get_UMI_count_per_CB	7

mutation_type_donut	7
num_mutation_histogram	8
parse_ref_file	8
plot_alignment_permutation	9
plot_construct	9
plot_deletion_hotspot	10
plot_insertion_hotspot	10
plot_lorenz_curve	11
plot_point_substitution_hotspot	11
plot_score_distribution	12
sequence_alignment	12
sequence_alignment_for_10x	14
sequence_permutation	15
seq_split	16
seq_to_character	16

Index	18
--------------	-----------

circular_chordgram	<i>Display a circos plot with links for a given data frame.</i>
--------------------	---

Description

Display a circos plot with links for a given data frame.

Usage

```
circular_chordgram(df, title, ref, use_log_count = TRUE, count_cutoff = 1)
```

Arguments

df	A data frame that contains data to be visualized on the plot
title	The main title of the plot
ref	An reference object, output of ‘parse_ref_file’
use_log_count	Plot the circular plot using log()
count_cutoff	A cutoff to remove link whose log10-count are less than the value (default: 1).

Value

It doesn’t generate any specific output.

circular_histogram	<i>Display a circos plot with a histogram for a given data frame.</i>
--------------------	---

Description

Display a circos plot with a histogram for a given data frame.

Usage

```
circular_histogram(df, title, ref)
```

Arguments

df	a data frame that contains data to be visualized on the plot.
title	The main title of the plot.
ref	A traceQC object

Value

It doesn't generate any specific output.

filter_mutations	<i>Filter mutations based on read count per UMI</i>
------------------	---

Description

Filter mutations based on read count per UMI

Usage

```
filter_mutations(data, include_max = TRUE, freq_threshold)
```

Arguments

data	A data frame.
include_max	include the mutations with maximum read count.
freq_threshold	threshold of mutation frequency.

Value

A filtered data frame.

`filter_mutations_per_cell`*Filter mutations based on read count per cell*

Description

Filter mutations based on read count per cell

Usage

```
filter_mutations_per_cell(data, freq_threshold)
```

Arguments

`data` A data frame.
`freq_threshold` threshold of mutation frequency.

Value

A filtered data frame.

`filter_mutations_per_UMI`*Filter mutations based on read count per UMI*

Description

Filter mutations based on read count per UMI

Usage

```
filter_mutations_per_UMI(data, include_max = TRUE, freq_threshold)
```

Arguments

`data` A data frame.
`include_max` include the mutations with maximum read count.
`freq_threshold` threshold of mutation frequency.

Value

A filtered data frame.

find_position	<i>Creating a data frame of mutation events.</i>
---------------	--

Description

Creating a data frame of mutation events.

Usage

```
find_position(insertions, deletions, mutations, target_seq, score, read_count)
```

Arguments

insertions	A list that contains insertion events.
deletions	A list that contains deletion events.
mutations	A list that contains mutation (substitution) events.
target_seq	A list that contains the alignment for each event.
score	A list that contains the alignment score for each event.
read_count	A vector that contains counts for each event.

Value

A data frame that contains the event information.

format_mutation_df	<i>Format a mutation data frame for output.</i>
--------------------	---

Description

Format a mutation data frame for output.

Usage

```
format_mutation_df(mutation_df, is_singlecell)
```

Arguments

is_singlecell	If the dataframe is is single-cell.
mutations	A data frame of mutations. The output of seq_to_character.

Value

A formatted data frame of mutations.

Examples

```
out_df <- format_mutation_df(mutation,is_singlecell=FALSE)
```

get_abspath	<i>Get absolute path of a file.</i>
-------------	-------------------------------------

Description

Get absolute path of a file.

Usage

```
get_abspath(f)
```

Arguments

f	A relative or absolute file path.
---	-----------------------------------

Value

It returns an absolute path for a file.

get_read_count_per_UB	<i>Get read count per UMI.</i>
-----------------------	--------------------------------

Description

Get read count per UMI.

Usage

```
get_read_count_per_UB(df)
```

Arguments

df	aligned reads
----	---------------

Value

A data frame of UMI

get_UMI_count_per_CB *Get UMI count per cell.*

Description

Get UMI count per cell.

Usage

```
get_UMI_count_per_CB(df)
```

Arguments

df aligned reads

Value

A data frame of Cells.

mutation_type_donut *A pie chart that shows a summary of mutation types.*

Description

A pie chart that shows a summary of mutation types.

Usage

```
mutation_type_donut(mutations)
```

Arguments

mutations A mutation dataframe

Value

A ggplot2 object that shows the pie chart

Examples

```
mutation <- read_tsv(system.file("extdata/test_data", "mutation.txt", package="TraceQC"))
mutation_type_donut(mutation)
```

`num_mutation_histogram`*A barplot to show distribution of the number of mutations per barcode*

Description

A barplot to show distribution of the number of mutations per barcode

Usage

```
num_mutation_histogram(mutations)
```

Arguments

`mutations` A mutation dataframe

Value

A ggplot2 object that shows the barplot

Examples

```
mutation <- read_tsv(system.file("extdata/test_data", "mutation.txt", package="TraceQC"))
num_mutation_histogram(mutation)
```

`parse_ref_file`*Parsing reference sequence file*

Description

Parsing reference sequence file

Usage

```
parse_ref_file(ref_file)
```

Arguments

`ref_file` A path of a reference sequence file.

Value

A list with those four elements.

- ‘refseq’: The reference sequence.
- ‘regions’: Detailed information about the reference sequence.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_carlin.txt", package="TraceQC")
ref <- parse_ref_file(ref_file)
```

plot_alignment_permutation	<i>Visualization of alignment permutation.</i>
----------------------------	--

Description

Visualization of alignment permutation.

Usage

```
plot_alignment_permutation(alignment_permutation)
```

Arguments

ref	an data frame of permutation sequence, output of ‘sequence_permutation’
-----	---

Value

it returns A ggplot2 object that shows the permutation.

plot_construct	<i>Visualization of the construct (reference sequence) information.</i>
----------------	---

Description

Visualization of the construct (reference sequence) information.

Usage

```
plot_construct(ref, chr_per_row = 50, chr_size = 10)
```

Arguments

ref	an reference object, output of ‘parse_ref_file’
chr_per_row	number of charachters per row.
chr_size	the font size of characters.

Value

it returns A ggplot2 object that shows the construct reference sequence.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_carlin.txt", package="TraceQC")
ref <- parse_ref_file(ref_file)
plot_construct(ref, chr_per_row=50, chr_size=5)
```

`plot_deletion_hotspot` *Display a circos plot that shows overall deletion pattern across the barcodes.*

Description

Display a circos plot that shows overall deletion pattern across the barcodes.

Usage

```
plot_deletion_hotspot(mutations, ref, use_log_count = TRUE, count_cutoff = 1)
```

Arguments

<code>mutations</code>	A mutations dataframe.
<code>ref</code>	A reference object.
<code>count_cutoff</code>	A cutoff to remove link whose count (or log10-count) are less than the value (default: 1).

Value

It doesn't generate any specific output.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_hgRNA_invitro.txt", package="TraceQC")
ref <- parse_ref_file(ref_file)
mutation <- read_tsv(system.file("extdata/test_data", "mutation.txt", package="TraceQC"))
plot_deletion_hotspot(mutation, ref)
```

`plot_insertion_hotspot` *Display a circos plot that shows overall insertion pattern across the barcodes.*

Description

Display a circos plot that shows overall insertion pattern across the barcodes.

Usage

```
plot_insertion_hotspot(mutations, ref, use_log_count = TRUE, count_cutoff = 1)
```

Arguments

<code>mutations</code>	A mutations dataframe.
<code>ref</code>	A reference object.
<code>count_cutoff</code>	A cutoff to remove link whose count (or log10-count) are less than the value (default: 1).

Value

It won't return any specific object.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_hgRNA_invitro.txt", package="TraceQC")
ref <- parse_ref_file(ref_file)
mutation <- read_tsv(system.file("extdata/test_data", "mutation.txt", package="TraceQC"))
plot_insertion_hotspot(mutation, ref)
```

plot_lorenz_curve	<i>Drawing Lorenz Curve</i>
-------------------	-----------------------------

Description

The Lorenz curve shows an inequality of barcode distribution of the sample.

Usage

```
plot_lorenz_curve(aligned_reads)
```

Arguments

aligned_reads A aligned_reads dataframe.

Value

A ggplot2 object that shows Lorenz Curve

Examples

```
plot_lorenz_curve(aligned_reads)
```

plot_point_substitution_hotspot	<i>Display a mutation hotspot circos plot.</i>
---------------------------------	--

Description

The circos plot shows the frequency of mutation events for each nucleotide.

Usage

```
plot_point_substitution_hotspot(mutations, ref)
```

Arguments

mutations A mutations dataframe.
ref A reference object.

Value

It won't return any specific object.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_hgRNA_invitro.txt", package="TraceQC")
ref <- parse_ref_file(ref_file)
mutation <- read_tsv(system.file("extdata/test_data", "mutation.txt", package="TraceQC"))
plot_insertion_hotspot(mutation, ref)
```

plot_score_distribution

Drawing a score distribution plot

Description

Drawing a score distribution plot

Usage

```
plot_score_distribution(aligned_reads)
```

Arguments

aligned_reads A aligned_reads dataframe.

Value

A ggplot2 object that shows alignment score distribution.

Examples

```
plot_score_distribution(aligned_reads)
```

sequence_alignment

Function for a sequence alignment between the reference file and sample.

Description

The function is an wrapper of a python function which performs a global pairwise sequence alignment by biopython package.

Usage

```
sequence_alignment(
  input_file,
  ref_file,
  output_file = "aligned_reads.txt",
  python_path = "python3",
  match = 2,
  mismatch = -2,
  gapopen = -6,
  gapextension = -0.1,
  ncores = 4,
  penalize_end_gaps = 1,
  return_df = FALSE
)
```

Arguments

<code>input_file</code>	A FASTQ file path (required).
<code>ref_file</code>	A path of a reference sequence file (required).
<code>output_file</code>	The output path. An output of the alignment will be stored at the path (default: "aligned_reads.txt").
<code>python_path</code>	The path to Python. (default: "python3").
<code>match</code>	The score for a correct basepair matching (default: 2).
<code>mismatch</code>	The penalty score for a basepair mismatching (default: -2).
<code>gapopen</code>	The gap opening score for the alignment (default: -6).
<code>gapextension</code>	The gap extension score for the alignment (default: -0.1).
<code>ncores</code>	The number of cores for the parallel processing.
<code>penalize_end_gaps</code>	Whether penalize the end gap, the value should take 0/1 (default: 1).
<code>return_df</code>	A logical argument to report what type of output will be created from the function.

Value

It returns a data frame of the alignment result if 'return_df' is 'T' and 'NULL' otherwise.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_carlin.txt", package="TraceQC")
ref <- parse_ref_file(ref_file)
input_file <- system.file("extdata/test_data/raw_sequence", "hgRNA_example.fastq.gz", package="TraceQC")
output_file <- "./aligned_reads.txt"
sequence_alignment(input_file=input_file, ref_file=ref_file,
  output_file=output_file)
```

```
sequence_alignment_for_10x
```

Function for a sequence alignment between the reference file and sample for 10x data.

Description

The function is an wrapper of a python function which performs a global pairwise sequence alignment by biopython package.

Usage

```
sequence_alignment_for_10x(
  input_file,
  ref_file,
  output_file = "aligned_reads.txt",
  python_path = "python3",
  match = 2,
  mismatch = -2,
  gapopen = -6,
  gapextension = -0.1,
  penalize_end_gaps = 1,
  ncores = 4
)
```

Arguments

<code>input_file</code>	A FASTQ file path (required).
<code>ref_file</code>	A path of a reference sequence file (required).
<code>output_file</code>	The output path. An output of the alignment will be stored at the path (default: "aligned_reads.txt").
<code>python_path</code>	The path to Python. (default: "python3").
<code>match</code>	The score for a correct basepair matching (default: 2).
<code>mismatch</code>	The penalty score for a basepair mismatching (default: -2).
<code>gapopen</code>	The gap opening score for the alignment (default: -6).
<code>gapextension</code>	The gap extension score for the alignment (default: -0.1).
<code>penalize_end_gaps</code>	Whether penalize the end gap, the value should take 0/1 (default: 1).
<code>ncores</code>	The number of cores for the parallel processing.

Value

It returns a data frame of the alignment result if 'return_df' is 'T' and 'NULL' otherwise.

Examples

```
input_file <- system.file("extdata/test_data/raw_sequence", "carlin_example.bam", package="TraceQC")
output_file <- "./aligned_reads.txt"
sequence_alignment_for_10x(input_file=input_file, ref_file=ref_file,
                           output_file=output_file)
```

sequence_permutation	<i>Function for finding threshold of sequence alignment. The function randomly permute certain percentage reference sequence and perform global alignment with the original reference sequence. By use the permuted sequence alignment score, users can filter the TraceQC alignment result.</i>
----------------------	--

Description

Function for finding threshold of sequence alignment. The function randomly permute certain percentage reference sequence and perform global alignment with the original reference sequence. By use the permuted sequence alignment score, users can filter the TraceQC alignment result.

Usage

```
sequence_permutation(
  ref_file,
  python_path = "python3",
  match = 2,
  mismatch = -2,
  gapopen = -6,
  gapextension = -0.1,
  penalize_end_gaps = 1,
  read_length = 0,
  permute_percent = seq(0, 1, length.out = 101),
  n = 2,
  output_file = "alignment_threshold.txt"
)
```

Arguments

ref_file	A path of a reference sequence file (required).
python_path	The path to Python. (default: "python3").
match	The score for a correct basepair matching (default: 2).
mismatch	The penalty score for a basepair mismatching (default: -2).
gapopen	The gap opening score for the alignment (default: -6).
gapextension	The gap extension score for the alignment (default: -0.1).
penalize_end_gaps	Whether penalize the end gap, the value should take 0/1 (default: 1).
output_file	The output path. An output of the alignment will be stored at the path (default: "alignment_threshold.txt").

Value

It returns a data frame of the alignment result with randomly permuted sequence.

Examples

```
ref_file <- system.file("extdata/test_data/ref", "ref_carlin.txt", package="TraceQC")
sequence_permutation(ref_file, out="./seq_permutate.txt")
```

seq_split	<i>Split a string by a fixed length and joined with
 HTML tag.</i>
-----------	--

Description

Split a string by a fixed length and joined with
 HTML tag.

Usage

```
seq_split(s, len = 50)
```

Arguments

s	The input string
len	The fixed length

Value

A string splited by the 'len' then joined by '
'

seq_to_character	<i>Identifying mutation events.</i>
------------------	-------------------------------------

Description

Identifying mutation events.

Usage

```
seq_to_character(
  aligned_reads,
  use_CPM,
  alignment_score_cutoff = 0,
  abundance_cutoff = 0
)
```

Arguments

use_CPM	Use count per million
alignment_score_cutoff	Minimum cutoff for alignment score
abundance_cutoff	Minimum cutoff for read count. This parameter are used with use_CPM.
traceQC_input	A TraceQC object.

Value

A data frame that contains the columns:

- ‘type’: The type of mutation.
- ‘start’: The starting position of mutation event.
- ‘length’: The length of mutation.
- ‘mutation_to’: A string that shows what mutation is occurred.
- ‘count’: The total number of the mutation events from the sample.

Examples

```
unique_sequences <- group_by(aligned_reads, target_seq, target_ref) %>%  
  summarise(score=max(score)) %>%  
  ungroup  
alignment_score_cutoff <- predict(model, 0.4)  
mutation <- seq_to_character(unique_sequences, use_CPM=FALSE,  
                             alignment_score_cutoff=alignment_score_cutoff)
```

Index

circular_chordgram, [2](#)
circular_histogram, [3](#)

filter_mutations, [3](#)
filter_mutations_per_cell, [4](#)
filter_mutations_per_UMI, [4](#)
find_position, [5](#)
format_mutation_df, [5](#)

get_abspath, [6](#)
get_read_count_per_UB, [6](#)
get_UMI_count_per_CB, [7](#)

mutation_type_donut, [7](#)

num_mutation_histogram, [8](#)

parse_ref_file, [8](#)
plot_alignment_permutation, [9](#)
plot_construct, [9](#)
plot_deletion_hotspot, [10](#)
plot_insertion_hotspot, [10](#)
plot_lorenz_curve, [11](#)
plot_point_substitution_hotspot, [11](#)
plot_score_distribution, [12](#)

seq_split, [16](#)
seq_to_character, [16](#)
sequence_alignment, [12](#)
sequence_alignment_for_10x, [14](#)
sequence_permutation, [15](#)