Package 'RiboseQC'

February 28, 2019

Title Ribo-seQC, a comprehensive Ribo-seq analysis tool

Version 0.99.0

Description Ribo-seQC is a powerful analysis tool for the analysis of Ribo-seq data, which is able to provide read-length specific analysis of both cytoplasmic and organellar ribosome, and provides interactive visualization of results in a dynamic html report

Depends rmarkdown, rtracklayer, GenomicAlignments, BSgenome, GenomicFiles, devtools, reshape2, ggplot2, knitr, DT, gridExtra, ggpubr, viridis, Biostrings, GenomicFeatures, BiocGenerics

License GPL-3 or above

Encoding UTF-8

LazyData FALSE

Name RiboseQC

biocViews RiboSeq, GenomeAnnotation, Transcriptomics, Software

RoxygenNote 6.1.1

NeedsCompilation no

Author Lorenzo Calviello [aut], Dominique Sydow [aut], Dermott Harnett [ctb, cre], Uwe Ohler [rev, fnd]

Maintainer Dermott Harnett <Dermot.Harnett@mdc-berlin.de>

R topics documented:

`calc_cutoffs_from_profiles`

Calculate offsets from 5'end profiles

Description

This function calculates cutoffs and frame resolution for Ribo-seq reads, for each read length and compartment.

Usage

```
calc_cutoffs_from_profiles(reads_profile, length_max)
```

Arguments

`reads_profile`	Profile of 5'ends around start and stop codon, as a DataFrame object with tx_ids as rows and positions as columns
`length_max`	Maximum cutoff to use

Details

Three methods are used and combined in the final choice: the position of maximum coverage around start codon is calculated for each transcript, and the most frequent one is stored in the "*_tab" objects. Such frequency values are also subjected to k-means clustering (centers=3) and the first value belonging to the highest cluster is selected, output as "*km_tab" objects. Analysis of aggregate plots, instead of frequencies, is performed again using kmeans (centers=3) using the same analysis above and stored in the "*km_meta" objects, and by simply calculating the maximum value in the profile, stored in the "*meta" objects. for each method, all reads ("absolute_") or only in-frame positions ("in_frame_") are considered. The final choice takes the most frequent cutoff chosen in all methods applied to in-frame positions.

Value

a list with a `final_cutoff` object, the frame analysis containing the displaying the max frame and the average all the calculated cutoffs in `cutoffs`, data used for the frame analysis in `frames`, and profiles around start codons in `profiles_start`.

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

See Also

RiboseQC_analysis

---

choose_readlengths *Filter read lengths for P-sites position calculation*

Description

This function selects a subset of readlenghts to be used in the P-sites calculation step

Usage

```
choose_readlengths(summary_data, choice = "max_coverage", nt_signals)
```

Arguments

`summary_data`	output data from the `calc_cutoffs_from_profiles` function
`choice`	Method used to select readlengths, defaults to "max_coverage"
`nt_signals`	Profiles of 5'ends around start codons

Details

Three different methods are available to choose readlengths: the "max_coverage" method selects all read lenghts with more in-frame signal compared to out-of-frame signal, on all codons; the "max_inframe" method starts with the most accurate read length and progressively selects read lengths which add in-frame signals in codons not covered by previous read lengths; the "all" method selects all available read lengths

Value

a list object containing different compartments. Each sub-list contains `final_choice`, the set of chosen read lengths with cutoffs, and `data`, the complete stats for each selection method

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

See Also

`calc_cutoffs_from_profiles`

`create_html_report` *Create the Ribo-seQC analysis report in html*

Description

This function creates the Ribo-seQC html report based on the Ribo-seQC analysis files generated with `RiboseQC_analysis`.

Usage

```
create_html_report(input_files, input_sample_names, output_file,
  extended = F)
```

Arguments

`input_files`	Character vector with full paths to data files generated with `RiboseQC_analysis`. Must be of same length as `input_sample_names`.
`input_sample_names`	Character vector containing input names (max. 5 characters per name). Must be of same length as `input_files`.
`output_file`	String; full path to html report file.
`extended`	creates a large html report including codon occupancy for each read length. Defaults to `FALSE`

Details

This function creates the html report visualizing the RiboseQC analysis data.

Input are two lists of the same length:

a) `input_files`: list of full paths to one or multiple input files (Ribo-seQC analysis files generated with `RiboseQC_analysis`) and

b) `input_sample_names`: list of corresponding names describing the file content in max. 5 characters (these are used as names in the report).

For the report, a RMarkdown file is rendered as html document, saved as `output_file`.

Additionally, all figures in the report are saved as PDF figures in an extra folder in the same directory as the report html file.

Example:
`output_file <- "\mydir\myreport.html"` will generate the html report `\mydir\myreport.html` and the folder `\mydir\myreport_plots\` for the RDS object files to be stored in.

Value

The function saves the html report file with the file path `output_file` and a folder containing all figures shown in the html report as RDS object files (located in the same directory as the html report).

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

plot_read_biotype_dist_1, plot_read_biotype_dist_2, plot_read_length_dist, plot_read_length_dist_by_bio
plot_read_biotype_dist_by_length, get_metagene_data, plot_metagene_hm_rmd, plot_metagene_hm,
plot_metagene_bar_rmd, plot_metagene_bar, plot_frame_dist_boxplot_rmd, plot_frame_dist_boxplot,
get_rl_and_cutoffs, get_default_rl_selection, get_top50_mapping, get_top50_cds_genes,
get_top50_all_genes, get_codon_usage_data, plot_codon_usage_positional_rmd, plot_codon_usage_positional
plot_codon_usage_bulk_rmd, plot_codon_usage_bulk

---

create_pdfs_from_rds_objects

Generate PDF files from RDS object files

Description

This function generates figures as PDF files from RDS object files.

Usage

```
create_pdfs_from_rds_objects(output_rds_path)
```

Arguments

`output_rds_path`
: String; full path to output folder for RDS object files. Example: /my_path_to/rds/

Value

This function creates PDF files from RDS object files.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

generate_rdata_list *Generate a list of R data objects*

Description

This function generates a list of loaded RData files to be used during Ribo-seQC report generation.

Usage

```
generate_rdata_list(input_files)
```

Arguments

input_files	List of RData file paths generated by RiboseQC_analysis. Example: `input_files <- c(sample1="//path//to//sample1", sample2="//path//to//sample2")`

Value

This function returns a list of loaded RData objects that were generated by RiboseQC_analysis.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

`get_codon_usage_data` *Get codon usage data (positional and bulk)*

Description

This function processes codon usage data generated by `RiboseQC_analysis`, i.e. codon usage

- per nucleotide position (i.e. positional codon usage) as well as
- summed up over all positions* (i.e. bulk codon usage)

for a specific data type, originating compartment, and read length, as well as based on a user-defined genetic code.

This data is used as input in `plot_codon_usage_bulk` to generate a bar plot, and in `plot_codon_usage_bulk_rmd` to iterativly generate plots for the Ribo-seQC report in section 7.2.

Please check `plot_codon_usage` for positional codon usage (instead of bulk codon usage).

* Information on positions included in analysis:
Based on P-site corrected reads, the codon usage within CDS regions for protein coding genes is examplatory calculated

- for the first 11 codons of the CDS (referred to as *start*),
- for 11 codons from the middle of the CDS (referred to as *middle*), and
- for the last 11 codons of the CDS (referred to as *stop*).

Usage

```
get_codon_usage_data(data, data_type, comp, rl)
```

Arguments

`data`	Object (list of lists) generated by `RiboseQC_analysis`: `res_all`.
`data_type`	String; select one of the following:

- `Codon_counts`: codon count in defined positions*,
- `P_sites_percodon`: P-sites count in defined positions*, or
- `P_sites_percodon_ratio`: ratio of P-sites counts to codon counts in defined positions*.
- `E_sites_percodon`: E-sites count in defined positions*, or
- `E_sites_percodon_ratio`: ratio of E-sites counts to codon counts in defined positions*.

- A_sites_percodon: A-sites count in defined positions*, or
- A_sites_percodon_ratio: ratio of A-sites counts to codon counts in defined positions*.

comp — String for originating compartment.
Check for available originating compartments in the data set using:
names(res_all$profiles_P_sites$Codon_count)

rl — String for read length.
Check for available read lengths in the data set using:
names(res_all$profiles_P_sites[[data_type]][[comp]])

Value

This function returns a list (e.g. called codon_usage_bulk_data) with information on bulk codon usage:

- codon_usage_bulk_data$data contains the data on bulk codon usage,
- codon_usage_bulk_data$data_type saves the data type used (see parameter data_type)
- codon_usage_bulk_data$comp saves the originating compartment used (see parameter comp), and
- codon_usage_bulk_data$rl saves the read length used (see parameter rl).

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

get_default_rl_selection

Get default choice of read lengths

Description

This function returns selected read lengths per originating compartment. These read lengths build the basis for all P-site based calculations such as metagene and codon usage analyses (e.g. plot_metagene_hm and plot_codon_usage)

This data is used in the Ribo-seQC report in section 4.2.3 (displayed as table).

Usage

```
get_default_rl_selection(rdata_list)
```

Arguments

rdata_list	List of RiboseQC analysis RData objects generated by generate_rdata_list.

Value

This function returns data.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

get_metagene_data *Get 5'/P-site profile data for metagene analysis*

Description

This function processes profile data generated by RiboseQC_analysis.

This data is used as input in plot_metagene_hm to generate plot for the Ribo-seQC report in section 4.1/4.3.

Usage

```
get_metagene_data(data, profile_type, res, comp)
```

Arguments

res	Resolution (subcodon or bins)

Subcodon resolution:
five_prime_subcodon
(in order to call res_all$profiles_fivepr$five_prime_subcodon) or
P_sites_subcodon
(in order to call res_all$profiles_P_sites$P_sites_subcodon)

Read coverage for the first 25nt after the transcription start site (TSS), 25nt before and 33nt after the start codon, 33nt from the middle of the CDS, 33nt before and 25nt after the stop codon, and the last 25nt before the transcription end site (TES).

Bins:
five_prime_bins
(in order to call res_all$profiles_fivepr$five_prime_bins) or

P_sites_bins
(in order to call res_all$profiles_P_sites$P_sites_bins)

Read coverage for 50 bins between TSS and start codon, 100 bins for the CDS, and 50 after stop codon to TES.

comp — String for originating compartment

Check for available originating compartments in the data set using: names(res_all$profiles_fivepr$f

profiles — 5' or P-site profile data generated by RiboseQC_analysis:

res_all$profiles_fivepr or
res_all$profiles_P_sites

Consists of DataFrames each containing counts of 5' or P-site profiles, calculated for different resolution types (see parameter res), originating compartments (see parameter comp), and read lengths per input sample.

Example to access DataFrame:
res_all$profiles_fivepr[[res]][[comp]][[read_length]] or
res_all$profiles_P_sites[[res]][[comp]][[read_length]]

Value

This function returns data profile_data as list(data_single, data_all, res) with profile data

- for all read lengths individually (profile_data[1]] is data_single) and
- for all read lenghts summarized (profile_data[[2]] is data_all).

with different scaling: no scaling (none), log2 scaling (log2), and z scoring (zscore), accessable via e.g. profile_data[[1]]$none or profile_data[[2]]$zscore.

profile_data[[3]] saves the resolution type res for later use during plotting.

profile_data[[4]] stores information on whether data represents 5' or P site profiles (profile_type) for later use during plotting.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

get_ps_fromsplicemin *Offset spliced reads on minus strand*

Description

This function calculates P-sites positions for spliced reads on the minus strand

Usage

```
get_ps_fromsplicemin(x, cutoff)
```

Arguments

x	a GAlignments object with a cigar string
cutoff	number representing the offset value

Value

a GRanges object with offset reads

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

get_ps_fromspliceplus *Offset spliced reads on plus strand*

Description

This function calculates P-sites positions for spliced reads on the plus strand

Usage

```
get_ps_fromspliceplus(x, cutoff)
```

Arguments

x	a GAlignments object with a cigar string
cutoff	number representing the offset value

Value

a GRanges object with offset reads

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

get_rl_and_cutoffs *Get selected read lengths and cutoffs*

Description

This function retrieves data on selected read lengths (per originating compartment), e.g. their cutoff values, frame preference and codon gain.

This data is used in the Ribo-seQC report in section 4.2.2 (displayed as table).

Usage

```
get_rl_and_cutoffs(rdata_list)
```

Arguments

rdata_list	List of RiboseQC analysis RData objects generated by generate_rdata_list.

Value

This function returns data.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

get_top50_all_genes *Get top 50 abundant genes (all genes)*

Description

This function retrieves data on the top 50 abundant genes.

This data is used in the Ribo-seQC report in section 6 (displayed as table).

Usage

```
get_top50_all_genes(rdata_list)
```

Arguments

rdata_list	List of RiboseQC analysis RData objects generated by generate_rdata_list.

Value

This function returns data to be displayed as table in the html report.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

`get_top50_cds_genes` *Get top 50 abundant genes (CDS regions for protein coding genes)*

Description

This function retrieves data on the top 50 abundant CDS regions for protein coding genes.

This data is used in the Ribo-seQC report in section 6 (displayed as table).

Usage

```
get_top50_cds_genes(rdata_list)
```

Arguments

`rdata_list`	List of RiboseQC analysis RData objects generated by `generate_rdata_list`.

Value

This function returns data to be displayed as table in the html report.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

get_top50_mapping *Get top 50 mapping positions*

Description

This function retrieves data on the top 50 positions (nucleotide resolution) are listed where most reads (their 5' end) map to, revealing possibly contaminating sequences.

This data is used in the Ribo-seQC report in section 5 (displayed as table).

Usage

```
get_top50_mapping(rdata_list)
```

Arguments

rdata_list	List of RiboseQC analysis RData objects generated by generate_rdata_list.

Value

This function returns data to be displayed as table in the html report.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

load_annotation *Load genomic features and genome sequence*

Description

This function loads the annotation created by the prepare_annotation_files function

Usage

```
load_annotation(path)
```

Arguments

path	Full path to the *Rannot R file in the annotation directory used in the prepare_annotation_files funct

Value

introduces a GTF_annotation object and a genome_seq object in the parent environment

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

See Also

prepare_annotation_files

---

plot_codon_usage_bulk *Plot bulk codon usage bar plots*

Description

This function plots the codon usage summed up over all positions* (bulk codon usage) as bar plot for a specific data type, originating compartment, and read length, as well as based on a user-defined genetic code.

* Information on positions included in analysis:
Based on P-site corrected reads, the codon usage within CDS regions for protein coding genes is examplatory calculated

- for the first 11 codons of the CDS (referred to as *start*),
- for 11 codons from the middle of the CDS (referred to as *middle*), and
- for the last 11 codons of the CDS (referred to as *stop*).

Usage

```
plot_codon_usage_bulk(codon_usage_data, sample = "",
  output_rds_path = "")
```

Arguments

sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.
codon_usage_bulk_data	List containing codon usage bulk data and meta data, generated by get_codon_usage_data.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

plot_codon_usage_bulk_rmd

Plot bulk codon usage bar plots within the RMarkdown document

---

Description

This function generates iteratively all bulk codon usage bar plots; iteration over originating compartments, read length, and data type (codon count, read count, codon-read count ratio).

These plots are displayed in the Ribo-seQC report in section 8.

Usage

```
plot_codon_usage_bulk_rmd(data, sample = "", output_rds_path = "")
```

Arguments

data	Object (list of lists) generated by RiboseQC_analysis: res_all.
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns iteratively all bulk codon usage plots for the html report and saves the same plots as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report, get_metagene_data, plot_metagene_hm

plot_codon_usage_positional *Plot positional codon usage heatmap*

Description

This function plots the codon usage per nucleotide position* (positional codon usage) as heatmap for a specific data type, originating compartment, read length, and scaling method, as well as based on a user-defined genetic code.

* Information on positions included in analysis:
Based on P-site corrected reads, the codon usage within CDS regions for protein coding genes is examplatory calculated

- for the first 11 codons of the CDS (referred to as *start*),
- for 11 codons from the middle of the CDS (referred to as *middle*), and
- for the last 11 codons of the CDS (referred to as *stop*).

Usage

```
plot_codon_usage_positional(codon_usage_data, scal, sample = "",
  output_rds_path = "")
```

Arguments

sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.
codon_usage_bulk_data	List containing codon usage bulk data and meta data, generated by get_codon_usage_data.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_codon_usage_positional_rmd *Plot positional codon usage heatmaps within the RMarkdown document*

Description

This function generates iteratively all positional codon usage heatmaps; iteration over originating compartments, read length, data type (codon count, read count, codon-read count ratio), and scaling method (none, log2, zscale).

These plots are displayed in the Ribo-seQC report in section 7.

Usage

```
plot_codon_usage_positional_rmd(data, sample = "",
  output_rds_path = "")
```

Arguments

data	Object (list of lists) generated by RiboseQC_analysis: res_all.
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns iteratively all positional codon usage plots for the html report and saves the same plots as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_frame_dist_boxplot *Plot frame coverage of 5'-site profiles*

Description

This function plots the frame coverage of 5'-site profiles, i.e. the fraction of reads (their 5' end) in frame 0, 1, and 2 (defined by start codon), as boxplots (per-frame distributions over all transcripts) for individual read lengths as well as for all read lengths summarized.

Usage

```
plot_frame_dist_boxplot(analysis_frame_cutoff, comp, sample = "",
  output_rds_path = "")
```

Arguments

analysis_frame_cutoff	Object containing statistics on frame coverage (per originating compartment and read length) generated by RiboseQC_analysis. Example: res_all$selection_cutoffs$analysis_frame_cutoff
comp	String for originating compartment Check for available originating compartments in the data set using: names(res_all$profiles_fivepr$f
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Details

This plot is used in the Ribo-seQC report in section 4.2.1.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_frame_dist_boxplot_rmd *Plot frame coverage of 5'-site profiles within the RMarkdown document*

Description

This function generates iteratively all frame coverage boxplots (iteration over originating compartments).

These plots are displayed in the Ribo-seQC report in section 4.2.1.

Usage

```
plot_frame_dist_boxplot_rmd(analysis_frame_cutoff, sample = "",
  output_rds_path = "")
```

Arguments

analysis_frame_cutoff	Object containing statistics on frame coverage (per originating compartment and read length) generated by RiboseQC_analysis. Example: res_all$selection_cutoffs$analysis_frame_cutoff
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function integrates plots in the html report.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_metagene_bar *Plot 5'/P-site profile per read length as barplot*

Description

This function plots a 5' or P-site profile as barplot (for a specific originating compartment, resolution type, and read length).

This plot is used in the Ribo-seQC report in section 4.1/4.3.

Usage

```
plot_metagene_bar(metagene_data, rl, sample = "", output_rds_path = "")
```

Arguments

sample	String; sample name (selected from the input names iven in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.
data	Profile data for a resolution type, specific originating compartment, and read length. Example: **res_all$profiles_fivepr**$five_prime_subcodon$nucl$'30' or **res_all$profiles_P_sites**$five_prime_subcodon$nucl$'30' Shown in bold are fixed list names, remaining list names should be adapted to the resolution type, specific originating compartment, and read length of interest.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_metagene_bar_rmd *Plot 5'/P-site profile per read length as barplot in R Markdown*

Description

This function plots a 5' or P-site profile as barplot (for a specific originating compartment, resolution type, and read length).

This plot is used in the Ribo-seQC report in section 4.1/4.3.

Usage

```
plot_metagene_bar_rmd(metagene_data, sample = "", output_rds_path = "")
```

Arguments

metagene_data	...
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

...

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_metagene_hm *Plot 5'/P-site profiles (for all read lengths) as heatmap*

Description

This function plots a 5' or P-site profiles for all read lengths as heatmap (for a specific originating compartment, resolution type, and scaling method).

This plot is used in the Ribo-seQC report in section 4.1/4.3.

Usage

```
plot_metagene_hm(metagene_data, scal, sample = "",
  output_rds_path = "")
```

Arguments

scal	Scaling method: no scaling (none), log2 scaling (log2), or z scoring (zscore).
sample	String; sample name (selected from the input names iven in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.
data_profile	Profile data for a specific originating compartment and resolution type, generated using get_metagene_data.

Value

This function returns a plot that can be integrated in the html report and #' that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

plot_metagene_hm_rmd *Plot 5'/P-site profiles as heatmaps within the RMarkdown document*

Description

This function generates iteratively all heatmap plots (iteration over originating compartments, resolution types, and scaling methods).

These plots are displayed in the Ribo-seQC report in section 4.1/4.3.

Usage

```
plot_metagene_hm_rmd(data, profile_type, sample = "",
  output_rds_path = "")
```

Arguments

sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.
profiles	5' or P-site profile data generated by RiboseQC_analysis: res_all$profiles_fivepr or res_all$profiles_P_sites Consists of DataFrames each containing counts of 5' or P-site profiles, calculated for different resolution types (see parameter res), originating compartments (see parameter comp), and read lengths per input sample. Example to access DataFrame: res_all$profiles_fivepr[[res]][[comp]][[read_length]] or res_all$profiles_P_sites[[res]][[comp]][[read_length]]

Value

This function returns iteratively all 5' or P-site profile plots for the html report and saves the same plots as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report, get_metagene_data, plot_metagene_hm

---

plot_read_biotype_dist_1

Plot read location distribution by biotype (and originating compartment)

Description

This function plots the read location distribution by biotype (and originating compartment) for one input sample.

This plot is used in the Ribo-seQC report in section 1.1.

Usage

```
plot_read_biotype_dist_1(pos, sample, output_rds_path = "")
```

Arguments

pos	res_all$read_stats$positions generated by RiboseQC_analysis. data.frame containing the number of reads per biotype (rows) and originating compartment (columns).
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

plot_read_biotype_dist_2

Plot read location distribution by originating compartment (and biotype)

Description

This function plots the read location distribution by originating compartment (and biotype) for all input samples.

This plot is used in the Ribo-seQC report in section 1.2.

Usage

```
plot_read_biotype_dist_2(rdata_list, output_rds_path = "")
```

Arguments

rdata_list	List of RiboseQC analysis RData objects generated by generate_rdata_list.
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

`plot_read_biotype_dist_by_length` *Plot read length and location distribution (distribution per read length)*

Description

This funtion plots the biotype distribution for each originating compartment as distribtion per read length for one input sample (displayed as read count and as read count fraction).

This plot is used in the Ribo-seQC report in section 3.2.

Usage

```
plot_read_biotype_dist_by_length(reads_summary, sample,
  output_rds_path = "")
```

Arguments

`reads_summary`	`res_all$read_stats$reads_summary` generated by `RiboseQC_analysis` List of DataFrames: one DataFrame for each originating compartment, each DataFrame contains read counts per biotype (rows) and read lengths (columns).
`sample`	String; sample name (selected from the input names given in the `input_sample_names` parameter of `create_html_report`).
`output_rds_path`	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

---

plot_read_length_dist *Plot read length distribution*

Description

This function plots the read length distribution per originating compartment for one input sample.

This plot is used in the Ribo-seQC report in section 2.

Usage

```
plot_read_length_dist(rld, sample, output_rds_path = "")
```

Arguments

rld	res_all$read_stats$rld generated by RiboseQC_analysis data.frame containing the number of reads per originating compartment (rows) and read lengths (columns).
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

plot_read_length_dist_by_biotype *Plot read length and location distribution (distribution per biotype)*

Description

This funtion plots the read length distribution for each originating compartment as distribtion per biotype for one input sample (displayed as read count and as read count fraction).

This plot is used in the Ribo-seQC report in section 3.1.

Usage

```
plot_read_length_dist_by_biotype(reads_summary, sample,
  output_rds_path = "")
```

Arguments

reads_summary	res_all$read_stats$reads_summary generated by RiboseQC_analysis List of DataFrames: one DataFrame for each originating compartment, each DataFrame contains read counts per biotype (rows) and read lengths (columns).
sample	String; sample name (selected from the input names given in the input_sample_names parameter of create_html_report).
output_rds_path	String; full path to output folder for RDS object files created by this function. Defaults to NOT save RDS; to save RDS, provide path to destination folder.

Value

This function returns a plot that can be integrated in the html report and that can be saved as RDS object file.

Author(s)

Dominique Sydow, <dominique.sydow@posteo.de>

See Also

create_html_report

prepare_annotation_files *Prepare comprehensive sets of annotated genomic features*

Description

This function processes a gtf file and a twobit file (created using faToTwoBit from ucsc tools: http://hgdownload.soe.ucsc.edu/admin/exe/) to create a comprehensive set of genomic regions of interest in genomic and transcriptomic space (e.g. introns, UTRs, start/stop codons). In addition, by linking genome sequence and annotation, it extracts additional info, such as gene and transcript biotypes, genetic codes for different organelles, or chromosomes and transcripts lengths.

Usage

```
prepare_annotation_files(annotation_directory, twobit_file, gtf_file,
  scientific_name = "Homo.sapiens", annotation_name = "genc25",
  export_bed_tables_TxDb = T, forge_BSgenome = T, create_TxDb = T)
```

Arguments

annotation_directory	The target directory which will contain the output files
twobit_file	Full path to the genome file in twobit format
gtf_file	Full path to the annotation file in GTF format
scientific_name	A name to give to the organism studied; must be two words separated by a ".", defaults to Homo.sapiens
annotation_name	A name to give to annotation used; defaults to genc25
export_bed_tables_TxDb	Export coordinates and info about different genomic regions in the annotation_directory? It defaults to TRUE
forge_BSgenome	Forge and install a BSgenome package? It defaults to TRUE
create_TxDb	Create a TxDb object and a *Rannot object? It defaults to TRUE

Details

This function uses the makeTxDbFromGFF function to create a TxDb object and extract genomic regions and other info to a *Rannot R file; the mapToTranscripts and mapFromTranscripts functions are used to map features to genomic or transcript-level coordinates. GTF file mist contain "exon" and "CDS" lines, where each line contains "transcript_id" and "gene_id" values. Additional values such as "gene_biotype" or "gene_name" are also extracted. Regarding sequences, the twobit file, together with input scientific and annotation names, is used to forge and install a BSgenome package using the forgeBSgenomeDataPkg function.

The resulting GTF_annotation object (obtained after runnning `load_annotation`) contains:

`txs`: annotated transcript boundaries.
`txs_gene`: GRangesList including transcript grouped by gene.
`seqinfo`: indicating chromosomes and chromosome lengths.
`start_stop_codons`: the set of annotated start and stop codon, with respective transcript and gene_ids. reprentative_mostcommon,reprentative_boundaries and reprentative_5len represent the most common start/stop codon, the most upstream/downstream start/stop codons and the start/stop codons residing on transcripts with the longest 5'UTRs
`cds_txs`: GRangesList including CDS grouped by transcript.
`introns_txs`: GRangesList including introns grouped by transcript.
`cds_genes`: GRangesList including CDS grouped by gene.
`exons_txs`: GRangesList including exons grouped by transcript.
`exons_bins`: the list of exonic bins with associated transcripts and genes.
`junctions`: the list of annotated splice junctions, with associated transcripts and genes.
`genes`: annotated genes coordinates.
`threeutrs`: collapsed set of 3'UTR regions, with correspinding gene_ids. This set does not overlap CDS region.
`fiveutrs`: collapsed set of 5'UTR regions, with correspinding gene_ids. This set does not overlap CDS region.
`ncIsof`: collapsed set of exonic regions of protein_coding genes, with correspinding gene_ids. This set does not overlap CDS region.
`ncRNAs`: collapsed set of exonic regions of non_coding genes, with correspinding gene_ids. This set does not overlap CDS region.
`introns`: collapsed set of intronic regions, with correspinding gene_ids. This set does not overlap exonic region.
`intergenicRegions`: set of intergenic regions, defined as regions with no annotated genes on either strand.
`trann`: DataFrame object including (when available) the mapping between gene_id, gene_name, gene_biotypes, transcript_id and transcript_biotypes.
`cds_txs_coords`: transcript-level coordinates of ORF boundaries, for each annotated coding transcript. Additional columns are the same as as for the `start_stop_codons` object.
`genetic_codes`: an object containing the list of genetic code ids used for each chromosome/organelle. see GENETIC_CODE_TABLE for more info.
`genome_package`: the name of the forged BSgenome package. Loaded with `load_annotation` function.
`stop_in_gtf`: stop codon, as defined in the annotation.

Value

a TxDb file and a *Rannot files are created in the specified `annotation_directory`. In addition, a BSgenome object is forged, installed, and linked to the *Rannot object

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

See Also

load_annotation, forgeBSgenomeDataPkg, makeTxDbFromGFF.

RiboseQC_analysis *Perform a Ribo-seQC analysis*

Description

This function loads annotation created by the prepare_annotation_files function, and analyzes a BAM file.

Usage

```
RiboseQC_analysis(annotation_file, bam_files, read_subset = T,
  readlength_choice_method = "max_coverage", chunk_size = 5000000L,
  write_tmp_files = T, dest_names = NA, rescue_all_rls = FALSE,
  fast_mode = T, create_report = T, sample_names = NA,
  report_file = NA, extended_report = F, pdf_plots = T)
```

Arguments

annotation_file	Full path to the annotation file (*Rannot). Or, a vector with paths to one annotation file per bam file.
bam_files	character vector containing the full path to the bam files
read_subset	Select readlengths up to 99 percent of the reads, defaults to TRUE. Must be of length 1 or same length as bam_files.
readlength_choice_method	Method used to subset relevant read lengths (see choose_readlengths function); defaults to "max_coverage". Must be of length 1 or same length as bam_files.
chunk_size	the number of alignments to read at each iteration, defaults to 5000000, increase when more RAM is available. Must be between 10000 and 100000000
write_tmp_files	Should output all the results (in *results_RiboseQC_all)? Defaults to TRUE. Must be of length 1 or same length as bam_files.
dest_names	character vector containing the prefixes to use for the result output files. Defaults to same as bam_files
rescue_all_rls	Set cutoff of 12 for read lengths ignored because of insufficient coverage. Defaults to FALSE. Must be of length 1 or same length as bam_files.
fast_mode	Use only top 500 genes to build profiles? Defaults to TRUE. Must be of length 1 or same length as bam_files.
create_report	Create an html report showing the RiboseQC analysis results. Defaults to TRUE

sample_names	character vector containing the names for each sample analyzed (for the html report). Defaults to "sample1", "sample2" ...
report_file	desired filename for for the html report file. Defaults to the first entry of bam_files followed by ".html"
extended_report	creates a large html report including codon occupancy for each read length. Defaults to FALSE
pdf_plots	creates a pdf file for each produced plot. Defaults to TRUE

Details

This function loads different genomic regions created in the prepare_annotation_files step, separating features on different recognized organelles. The bam files is then analyzed in chunks to minimize RAM usage.
The complete list of analysis and output is as follows:

read_stats: contains:
read length distribution (rld) per organelle, positions containes mapping statistics on different genomic regions, reads_pos1 contains 5' end mapping positions for each read, separated by read length. counts_cds_genes: contains read mapping statistics on CDS regions of protein coding genes, including gene symbols, counts, RPKM and TPM values counts_all_genes: is a similar object, but contains statistics on all annotated genes. reads_summary: reports mapping statistics on different genomic regions and divided by read length and organelle.

profiles_fivepr contains:
five_prime_bins: a DataFrame object (one for each read length and compartment) with signal values over 50 5'UTR bins, 100 CDS bins and 50 3'UTR bins; one representative transcript (reprentative_mostcommon) is selected for each gene. five_prime_subcodon containes a similar structure, but for 25nt downstream the Transcription Start Site (TSS), 25nt upstream start codons, 33nt donwstream the start codon, 33nt in the middle of the ORF, 33nt upstream the stop codon, 25nt downstream the stop codon, and 25nt upstream the Transcription End Site (TES).

selection_cutoffs contains:
results_choice: containing the calculated cutoffs and selected readlengths, together with data about the different methods. results_cutoffs has statistics about calculated cutoffs, while analysis_frame_cutoff has extensive statistics concerning cutoff calculations and read length selection, see calc_cutoffs_from_profiles for more details.

P_sites_stats: contains the list of calculated P_sites, from all reads (P_sites_all), uniquely mapping reads (P_sites_all_uniq), or uniquely mapping reads with mismatches (P_sites_uniq_mm). junctions contains stastics on read mapping on annotated splice junctions. coverage for entire reads (no 5'ends or P_sites-transformed) on different strands and for all and uniquely mapping reads are also calculated.

profiles_P_sites contains:
P_sites_bins: profiles for each organelle and read length around binned transcript locations.
P_sites_subcodon: profiles for each organelle and read length around transcript start/ends and ORF start/ends.

Codon_counts: codon occurrences in the first 11 codons, middle 11 codons, and last 11 codons for each ORF.
P_sites_percodon: P_sites counts on each codon, separated by ORF positions as described above. Values are separated by organelle and read length.
P_sites_percodon_ratio: ratio of P_sites_percodon/Codon_counts, as a measure of P_site occupancy on each codon, divided again by organelle and read length, for different ORF positions.

sequence_analysis: contains a DataFrame object with the 50top mapping location in the genome, with the corresponding DNA sequence, number of reads mapping (also in percentage of total n of reads), and genomic feature annotation.

summary_P_sites: contains a DataFrame object summarizing the P_sites calculation and read length selection, including statistics on percentage of total reads used.

Value

the function saves a "results_RiboseQC_all" R file appended to the bam_files path including the complete list of outputs described here. In addition, bigwig files for coverage value and P_sites position is appended to the bam_files path, including also a summary of P_sites selection statistics, a smaller "results_RiboseQC" R file used for creating a dynamic html report, and a "for_SaTAnn" R object that can be used in the SaTAnn pipeline.

Author(s)

Lorenzo Calviello, <calviello.l.bio@gmail.com>

See Also

prepare_annotation_files, calc_cutoffs_from_profiles, choose_readlengths, create_html_report.

Index