Package 'rifi'

November 12, 2026

Title 'rifi' analyses data from rifampicin time series created by microarray or RNAseq

Version 1.16.0

Description 'rifi' analyses data from rifampicin time series created by microarray or RNAseq.
'rifi' is a transcriptome data analysis tool for the holistic identification of transcription and decay associated processes.
The decay constants and the delay of the onset of decay is fitted for each probe/bin. Subsequently, probes/bins of equal properties
are combined into segments by dynamic programming, independent of a existing genome annotation. This allows to detect transcript segments
of different stability or transcriptional events within one annotated gene.
In addition to the classic decay constant/half-
life analysis, 'rifi' detects processing sites, transcription pausing sites, internal transcription start sites in operons, sites of partial transcription termination in operons, identifies areas of likely transcriptional interference by the collision mechanism
and gives an estimate of the transcription velocity.
All data are integrated to give an estimate of continous transcriptional units, i.e. operons.
Comprehensive output tables and visualizations of the full genome result and the individual fits for all probes/bins are produced.

Depends R (>= 4.2)

Imports car, cowplot, doMC, parallel, dplyr, egg, foreach, ggplot2, graphics, grDevices, grid, methods, nls2, nnet, rlang, S4Vectors, scales, stats, stringr, SummarizedExperiment, tibble, rtracklayer, reshape2, utils

Suggests DescTools, devtools, knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

biocViews RNASeq, DifferentialExpression, GeneRegulation, Transcriptomics, Regression, Microarray, Software

BugReports https://github.com/CyanolabFreiburg/rifi

License GPL-3 + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.12

Language en-US

git_url https://git.bioconductor.org/packages/rifi

git_branch RELEASE_3_23

git_last_commit d66cefb

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Loubna Youssar [aut, ctb],
Walja Wanney [aut, ctb],
Jens Georg [aut, cre]

Maintainer Jens Georg <jens.georg@biologie.uni-freiburg.de>

Contents

apply_ancova . . . 3
apply_event_position . . . 5
apply_manova . . . 7
apply_Ttest_delay . . . 9
apply_t_test . . . 10
apply_t_test_ti . . . 12
check_input . . . 14
dataframe_summary . . . 15
dataframe_summary_events . . . 16
dataframe_summary_events_HL_int . . . 18
dataframe_summary_events_ps_itss . . . 19
dataframe_summary_events_velocity . . . 21
dataframe_summary_TI . . . 22
event_dataframe . . . 23
example_input_e_coli . . . 25
example_input_minimal . . . 26
example_input_synechocystis_6803 . . . 26
finding_PDD . . . 27
finding_TI . . . 28
fit_e_coli . . . 29
fit_minimal . . . 30
fit_synechocystis_6803 . . . 31
fold_change . . . 32
fragmentation_e_coli . . . 34
fragmentation_minimal . . . 35
fragmentation_synechocystis_6803 . . . 36
fragment_delay . . . 37
fragment_HL . . . 38
fragment_inty . . . 40
fragment_TI . . . 41

gff3_preprocess . . . 43
make_df . . . 44
make_pen . . . 45
nls2_fit . . . 47
penalties_e_coli . . . 48
penalties_minimal . . . 49
penalties_synechocystis_6803 . . . 50
predict_ps_itss . . . 51
preprocess_e_coli . . . 53
preprocess_minimal . . . 54
preprocess_synechocystis_6803 . . . 54
res_minimal . . . 55
rifi_fit . . . 57
rifi_fragmentation . . . 59
rifi_penalties . . . 60
rifi_preprocess . . . 62
rifi_stats . . . 64
rifi_summary . . . 65
rifi_visualization . . . 67
rifi_wrapper . . . 72
segment_pos . . . 73
stats_e_coli . . . 74
stats_minimal . . . 76
stats_synechocystis_6803 . . . 77
summary_e_coli . . . 79
summary_minimal . . . 83
summary_synechocystis_6803 . . . 86
TI_fit . . . 90
TUgether . . . 91
viz_pen_obj . . . 93
wrapper_e_coli . . . 94
wrapper_minimal . . . 94
wrapper_summary_synechocystis_6803 . . . 95

Index **96**

`apply_ancova` *%*

==========

apply_ancova

apply_ancova checks the variances between 2 segments showing either pausing site (ps) or internal starting site (ITSS) independently.

Description

apply_ancova is a statistical test to check if fragments showing ps and ITSS events have significant slope using Ancova test.The function uses ancova test. Ancova is applied when the data contains independent variables, dependent variables and covariant variables. In this case, segments are independent variables, position is the dependent variable and the delay is the covariant.

Usage

```
apply_ancova(inp)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.

Value

the SummarizedExperiment with the columns regarding statistics:

ID:	The bin/probe specific ID.
position:	The bin/probe specific position.
strand:	The bin/probe specific strand.
intensity:	The relative intensity at time point 0.
probe_TI:	An internal value to determine which fitting model is applied.
flag:	Information on which fitting model is applied.
position_segment:	The position based segment.
delay:	The delay value of the bin/probe.
half_life:	The half-life of the bin/probe.
TI_termination_factor:	String, the factor of TI fragment.
delay_fragment:	The delay fragment the bin belongs to.
velocity_fragment:	The velocity value of the respective delay fragment.
intercept:	The vintercept of fit through the respective delay fragment.
slope:	The slope of the fit through the respective delay fragment.
HL_fragment:	The half-life fragment the bin belongs to.
HL_mean_fragment:	The mean half-life value of the respective half-life fragment.
intensity_fragment:	The intensity fragment the bin belongs to.
intensity_mean_fragment:	The mean intensity value of the respective intensity fragment.
TU:	The overarching transcription unit.

`TI_termination_fragment:`
: The TI fragment the bin belongs to.

`TI_mean_termination_factor:`
: The mean termination factor of the respective TI fragment.

`seg_ID:` The combined ID of the fragment.

`pausing_site:` presence of pausing site indicated by +/-.

`iTSS_I:` presence of iTSS_I indicated by +/-.

`ps_ts_fragment:`
: The fragments involved in pausing site or iTSS_I.

`event_duration:`
: Integer, the duration between two delay fragments.

`event_ps_itss_p_value_Ttest:`
: p_value of pausing site or iTSS_I.

`p_value_slope:` Integer, the p_value added to the inp

`delay_frg_slope:`
: Integer, the slope value of the fit through the respective delay fragment

`velocity_ratio:`
: Integer, the ratio value of velocity from 2 delay fragments

Examples

```
data(stats_minimal)
apply_ancova(inp = stats_minimal)
```

`apply_event_position` *%*

==========

apply_event_position

apply_event_position extracts event time duration for pausing site or iTSS

Description

apply_event_position is a short version of apply_Ttest_delay function to extract event time duration for pausing site or iTSS. Its adds a new column with the duration.

Usage

```
apply_event_position(inp)
```

Arguments

`inp`	SummarizedExperiment: the input data frame with correct format.

Value

The SummarizedExperiment with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.

flag: Information on which fitting model is applied.

position_segment: The position based segment.

delay: The delay value of the bin/probe.

half_life: The half-life of the bin/probe.

TI_termination_factor: String, the factor of TI fragment.

delay_fragment: The delay fragment the bin belongs to.

velocity_fragment: The velocity value of the respective delay fragment.

intercept: The vintercept of fit through the respective delay fragment.

slope: The slope of the fit through the respective delay fragment.

HL_fragment: The half-life fragment the bin belongs to.

HL_mean_fragment: The mean half-life value of the respective half-life fragment.

intensity_fragment: The intensity fragment the bin belongs to.

intensity_mean_fragment: The mean intensity value of the respective intensity fragment.

TU: The overarching transcription unit.

TI_termination_fragment: The TI fragment the bin belongs to.

TI_mean_termination_factor: The mean termination factor of the respective TI fragment.

seg_ID: The combined ID of the fragment.

pausing_site: presence of pausing site indicated by +/-.

iTSS_I: presence of iTSS_I indicated by +/-.

ps_ts_fragment: The fragments involved in pausing site or iTSS_I.

event_duration: Integer, the duration between two delay fragments.

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I.

p_value_slope: Integer, the p_value added to the inp.

delay_frg_slope: Integer, the slope value of the fit through the respective delay fragment.

velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments.

event_position: Integer, position of the event added to the input.

Examples

```
data(stats_minimal)
apply_event_position(inp = stats_minimal)
```

apply_manova %

==========

apply_manova

apply_manova checks if the ratio of hl ratio and intensity ratio is statistically significant.

Description

apply_manova compares the variance between two fold-changes HL and intensity within the same TU (half-life frgA/half-life frgB/intensity frgA/intensity frgB). HL fragment could cover two intensity fragments therefore this function sets first fragments borders and uses manova_function. Manova checks the variance between 2 segments (independent variables) and two dependents variables (HL and intensity).

Usage

```
apply_manova(inp)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.

Value

The probe data frame with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.

flag: Information on which fitting model is applied.

position_segment: The position based segment.

delay: The delay value of the bin/probe.

half_life: The half-life of the bin/probe.

TI_termination_factor: String, the factor of TI fragment.

delay_fragment: The delay fragment the bin belongs to.

velocity_fragment: The velocity value of the respective delay fragment.

intercept: The vintercept of fit through the respective delay fragment.

slope: The slope of the fit through the respective delay fragment.

HL_fragment: The half-life fragment the bin belongs to.

HL_mean_fragment: The mean half-life value of the respective half-life fragment.

intensity_fragment: The intensity fragment the bin belongs to.

intensity_mean_fragment: The mean intensity value of the respective intensity fragment.

TU: The overarching transcription unit.

TI_termination_fragment: The TI fragment the bin belongs to.

TI_mean_termination_factor: The mean termination factor of the respective TI fragment.

seg_ID: The combined ID of the fragment.

pausing_site: presence of pausing site indicated by +/-.

iTSS_I: presence of iTSS_I indicated by +/-.

ps_ts_fragment: The fragments involved in pausing site or iTSS_I.

event_duration: Integer, the duration between two delay fragments.

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I.

p_value_slope: Integer, the p_value added to the inp.

delay_frg_slope: Integer, the slope value of the fit through the respective delay fragment.

velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments.

event_position: Integer, position of the event added to the input.

FC_HL: Integer, the fold change value of 2 HL fragments.

FC_fragment_HL: String, the fragments corresponding to HL fold change.

p_value_HL: Integer, the p_value added to the input of 2 HL fragments.

FC_intensity: Integer, the fold change value of 2 intensity fragments.

FC_fragment_intensity: String, the fragments corresponding to intensity fold change.

p_value_intensity: Integer, the p_value added to the input of 2 intensity fragments.

synthesis_ratio: Integer, the value correspomding to synthesis rate.

synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS.

FC_HL_intensity: Integer, the value corresponding to HL and intensity fold change.

FC_HL_intensity_fragment: String, the fragments corresponding to intensity and HL fold change.

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity,position of the half-life fragment is adapted to intensity fragment.

p_value_Manova: Integer, the p_value added to the input.

Examples

```
data(stats_minimal)
apply_manova(inp = stats_minimal)
```

apply_Ttest_delay %

==========

apply_Ttest_delay

apply_Ttest_delay checks the significance of the point between 2 segments showing pausing site (ps) and internal starting site (ITSS) independently

Description

apply_Ttest_delay: is a statistical test to check the significance of the point between 2 segments showing pausing site (ps) and internal starting site (ITSS) independently. The function uses t-test. The last point from the first segment and the first point from the second segment are selected and added to the residuals of each model. The sum is subjected to t-test.

Usage

```
apply_Ttest_delay(inp)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.

Value

the SummarizedExperiment with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.

flag: Information on which fitting model is applied.

position_segment: The position based segment.

delay: The delay value of the bin/probe.

half_life: The half-life of the bin/probe.

TI_termination_factor: String, the factor of TI fragment.

delay_fragment: The delay fragment the bin belongs to.

velocity_fragment: The velocity value of the respective delay fragment.

intercept: The vintercept of fit through the respective delay fragment.

slope: The slope of the fit through the respective delay fragment.

HL_fragment: The half-life fragment the bin belongs to.

HL_mean_fragment: The mean half-life value of the respective half-life fragment.

intensity_fragment: The intensity fragment the bin belongs to.

intensity_mean_fragment: The mean intensity value of the respective intensity fragment.

TU: The overarching transcription unit.

TI_termination_fragment: The TI fragment the bin belongs to.

TI_mean_termination_factor: The mean termination factor of the respective TI fragment.

seg_ID: The combined ID of the fragment.

pausing_site: presence of pausing site indicated by +/-.

iTSS_I: presence of iTSS_I indicated by +/-.

ps_ts_fragment: The fragments involved in pausing site or iTSS_I.

event_duration: Integer, the duration between two delay fragments.

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I.

Examples

```
data(stats_minimal)
apply_Ttest_delay(inp = stats_minimal)
```

`apply_t_test` %

==========

apply_t_test

apply_t_test uses the statistical t_test to check if the fold-change of half -life (HL) fragments and the fold-change intensity fragments respectively are significant.

Description

apply_t_test compares the mean of two neighboring fragments within the same TU to check if the fold-change is significant.Fragments with distance above threshold are not subjected to t-test.Dataframes with less than 3 rows are excluded.

Usage

```
apply_t_test(inp, threshold = 300)
```

Arguments

`inp`	SummarizedExperiment: the input data frame with correct format.
`threshold`	integer: threshold.

Details

The functions used are:

1. fragment_function: checks number of fragments inside TU, less than 2 are excluded otherwise they are gathered for analysis.
2. t_test_function: excludes dataframes with less than 3 rows, makes fold-change and apply t-test, assign fragments names and ratio, add columns with the corresponding p_values.

Value

the SummarizedExperiment with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.

flag: Information on which fitting model is applied.

position_segment: The position based segment.

delay: The delay value of the bin/probe.

half_life: The half-life of the bin/probe.

TI_termination_factor: String, the factor of TI fragment.

delay_fragment: The delay fragment the bin belongs to.

velocity_fragment: The velocity value of the respective delay fragment.

intercept: The vintercept of fit through the respective delay fragment.

slope: The slope of the fit through the respective delay fragment.

HL_fragment: The half-life fragment the bin belongs to.

HL_mean_fragment: The mean half-life value of the respective half-life fragment.

intensity_fragment: The intensity fragment the bin belongs to.

intensity_mean_fragment: The mean intensity value of the respective intensity fragment.

TU: The overarching transcription unit.

TI_termination_fragment: The TI fragment the bin belongs to.

TI_mean_termination_factor: The mean termination factor of the respective TI fragment.

seg_ID: The combined ID of the fragment.

pausing_site: presence of pausing site indicated by +/-.

iTSS_I: presence of iTSS_I indicated by +/-.

ps_ts_fragment: The fragments involved in pausing site or iTSS_I.

event_duration: Integer, the duration between two delay fragments.

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I.

p_value_slope: Integer, the p_value added to the inp.

delay_frg_slope: Integer, the slope value of the fit through the respective delay fragment.

velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments.

event_position: Integer, position of the event added to the input.

FC_HL: Integer, the fold change value of 2 HL fragments.

FC_fragment_HL: String, the fragments corresponding to HL fold change.

p_value_HL: Integer, the p_value added to the input of 2 HL fragments.

FC_intensity: Integer, the fold change value of 2 intensity fragments.

FC_fragment_intensity: String, the fragments corresponding to intensity fold change.

p_value_intensity: Integer, the p_value added to the input of 2 intensity fragments.

Examples

```
data(stats_minimal)
apply_t_test(inp = stats_minimal, threshold = 300)
```

`apply_t_test_ti` %

==========

apply_t_test_ti

apply_t_test_ti compares the mean of two neighboring TI fragments within the same TU.

Description

apply_t_test_ti uses the statistical t_test to check if two neighboring TI fragments are significant.

Usage

```
apply_t_test_ti(inp)
```

Arguments

`inp`	SummarizedExperiment: the input data frame with correct format.

Value

the SummarizedExperiment with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.
flag: Information on which fitting model is applied.
position_segment: The position based segment.
delay: The delay value of the bin/probe.
half_life: The half-life of the bin/probe.
TI_termination_factor: String, the factor of TI fragment.
delay_fragment: The delay fragment the bin belongs to.
velocity_fragment: The velocity value of the respective delay fragment.
intercept: The vintercept of fit through the respective delay fragment.
slope: The slope of the fit through the respective delay fragment.
HL_fragment: The half-life fragment the bin belongs to.
HL_mean_fragment: The mean half-life value of the respective half-life fragment.
intensity_fragment: The intensity fragment the bin belongs to.
intensity_mean_fragment: The mean intensity value of the respective intensity fragment.
TU: The overarching transcription unit.
TI_termination_fragment: The TI fragment the bin belongs to.
TI_mean_termination_factor: The mean termination factor of the respective TI fragment.
seg_ID: The combined ID of the fragment.
pausing_site: presence of pausing site indicated by +/-.
iTSS_I: presence of iTSS_I indicated by +/-.
ps_ts_fragment: The fragments involved in pausing site or iTSS_I.
event_duration: Integer, the duration between two delay fragments.
event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I.
p_value_slope: Integer, the p_value added to the inp.
delay_frg_slope: Integer, the slope value of the fit through the respective delay fragment.
velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments.
event_position: Integer, position of the event added to the input.
FC_HL: Integer, the fold change value of 2 HL fragments.
FC_fragment_HL: String, the fragments corresponding to HL fold change.
p_value_HL: Integer, the p_value added to the input of 2 HL fragments.
FC_intensity: Integer, the fold change value of 2 intensity fragments.
FC_fragment_intensity: String, the fragments corresponding to intensity fold change.
p_value_intensity: Integer, the p_value added to the input of 2 intensity fragments.
synthesis_ratio: Integer, the value correspomding to synthesis rate.
synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS.
FC_HL_intensity: Integer, the value corresponding to HL and intensity fold change.
FC_HL_intensity_fragment: String, the fragments corresponding to intensity and HL fold change.
FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity,position of the half-life fragment is adapted to intensity fragment.
p_value_Manova: Integer, the p_value added to the input.
p_value_TI: Integer, the p_value added to the input.
TI_fragments_p_value: String, the fragments subjected to statistical test.

Examples

```
data(stats_minimal)
apply_t_test_ti(inp = stats_minimal)
```

---

check_input %

==========

Check_input

Check_input reviews the input given by the user

Description

check_input stops the operation if the input data frame has severe faults. Less severe faults lead to the removal of wrong IDs and a warnings describing the problem. The Summarized Experiment colData must have the columns "timepoint" with the timepoints convertible to numeric and containing the timepoint 0. If replicates are used the column in colData must be called "replicate". The replicate must be convertible to numeric. In the RowRanges, optionally, IDs can be given as character (except ",","|","_"), but need to refer to a unique position/strand combination. Strand information needs to be given. The relative intensity in the assay must be numeric. The relative intensity for the first time point cannot be 0 or NA.

Usage

```
check_input(inp, thrsh = 0)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
thrsh	numeric: the minimal allowed intensity for time point "0".

Value

the SummarizedExperiment object: checked, and with position, ID and filtration added to the rowRanges.

Examples

```
data(example_input_minimal)
check_input(inp = example_input_minimal, thrsh = 0)
```

dataframe_summary %

==========

dataframe_summary

dataframe_summary creates two tables relating gene annotation to fragments

Description

dataframe_summary creates two tables summary of segments and their half-lives. The first output is bin/probe features and the second one is intensity fragment based.The dataframe_summary creates one table with feature_type, gene, locus_tag, position, strand, TU, delay_fragment, HL_fragment, half_life, intensity_fragment, intensity and velocity. The second table is similar to the first one but in compact form. It contains the same columns, the only difference is on position where a start and end position are indicated separately.

Usage

```
dataframe_summary(data, input)
```

Arguments

data	SummarizedExperiment: the input data frame with correct format.
input	dataframe: dataframe from event_dataframe function.

Value

bin_df: all information regarding bins:

position: Integer, position of the bin/probe on the genome
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: Boolean. The bin/probe specific strand (+/-)
segment: String, the bin/probe segment on the genome
TU: String, The overarching transcription unit
delay_fragment: The delay fragment the bin belongs to
delay: Integer, the delay value of the bin/probe
HL_fragment: The half-life fragment the bin belongs to
half_life: The half-life of the bin/probe
intensity_fragment: The intensity fragment the bin belongs to
intensity: The relative intensity at time point 0
flag: String, the flag of the bin/probe, contains information or the distribution for the #'different fitting models

TI_termination_factor: String, the TI termination factor determined by TI

`frag_df:` all information regarding fragments:

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

first_position_frg: Integer, the bin/probe specific first position

last_position_frg: Integer, the bin/probe specific last position

strand: Boolean. The bin/probe specific strand (+/-)

TU: String, The overarching transcription unit

segment: String, the bin/probe segment on the genome

delay_fragment: String, the delay fragment the bin belongs to

HL_fragment: Integer, the half_life fragment of the bin/probe belongs to

half_life: Integer, the half-life of the bin/probe

HL_SD: Integer, the half-life standard deviation of the HL fragment, bin/probe based

HL_SE: Integer, the half-life standard error of the HL fragment, bin/probe based

intensity_fragment: Integer, the intensity fragment the bin belongs to

intensity: Integer, the relative intensity of bin/probe at time point 0

intensity_SD: Integer, the intensity standard deviation of the intensity fragment, bin/probe based

intensity_SE: Integer, the intensity standard error of the intensity fragment, bin/probe based

velocity: The velocity value of the respective delay fragment

Examples

```
data(stats_minimal)
data(res_minimal)
dataframe_summary(data = stats_minimal, input = res_minimal)
```

`dataframe_summary_events` *%*

==========

dataframe_summary_events

dataframe_summary_events creates one table with all events between the segments

---

Description

dataframe_summary_events creates one table with the following columns: event, features, p_value, event_position, event_duration, position, region, gene, locus_tag, strand, TU, segment_1, segment_2, length, velocity_ratio, FC_HL, FC_intensity, FC_HL/FC_intensity.

Usage

```
dataframe_summary_events(data, data_annotation)
```

Arguments

`data`	SummarizedExperiment: the input data frame with correct format.
`data_annotation`	dataframe: dataframe from processed gff3 file.

Value

event: String, event type either pausing site, iTSS_I, iTSS_II, Termination, HL_event, Int_event, HL_Int_event and velocity_change

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

FC_HL: Integer, the fold change value of 2 HL fragments

FC_intensity: Integer, the fold change value of 2 intensity fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

synthesis_ratio: Fold change of half-life/ fold change of intensity

event_position: Integer, the position middle between 2 fragments with an event

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: Boolean. The bin/probe specific strand (+/-)

TU: String, The overarching transcription unit

segment_1: String, the first segment of the event, includes the segment, TU, delay fragment in case of ps or iTSS_I. The rest of the events include HL fragment and intensity fragment

segment_2: String, same description as segment_1 but is the second fragment of the event

event_duration: Integer, the difference (min) between 2 delay fragment when ps or iTSS_I happen

gap_fragments: Integer, length in position (nt), calculated by the difference between the last position of the first fragment and the first position of the second fragment.

features: Integer, number of fragements involved on the event

Examples

```
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
data(stats_minimal)
dataframe_summary_events(data = stats_minimal,
data_annotation = metadata(stats_minimal)$annot[[1]])
```

---

dataframe_summary_events_HL_int

%

==========

dataframe_summary_events_HL_int

dataframe_summary_events_HL_int creates one table with all events between the segments

---

Description

The dataframe_summary_events_HL_int creates one table with the following columns: event, features, p_value, event_position, position, region, gene, locus_tag, strand, TU, segment_1, segment_2, length, FC_HL, FC_intensity, FC_HL/FC_intensity.

Usage

```
dataframe_summary_events_HL_int(data, data_annotation)
```

Arguments

data	SummarizedExperiment: the input data frame with correct format.
data_annotation	dataframe: dataframe from processed gff3 file.

Value

event: String, event type.

p_value: Integer, p_value of the event.

p_adjusted: Integer, p_value adjusted.

FC_HL: Integer, the fold change value of 2 HL fragments.

FC_intensity: Integer, the fold change value of 2 intensity fragments.

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment.

FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity.

feature_type: String, region annotation covering the fragments.

gene: String, gene annotation covering the fragments.

locus_tag: String, locus_tag annotation covering the fragments.

strand: Boolean. The bin/probe specific strand (+/-).

TU: String, The overarching transcription unit.

segment_1: String, the first segment of the event, includes the segment, TU, delay fragment in case of ps or iTSS_I. The rest of the events include HL fragment and could be extended intensity fragment.

segment_2: String, the second fragment of the two of fragments subjected to analysis.

event_duration: Integer, the difference (min) between 2 delay fragment when ps or iTSS_I happen.

gap_fragments: Integer, length in position (nt), calculated by the difference between the last position of the first fragment and the first position of the second fragment.

features: Integer, number of fragements involved on the event

Examples

```
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
data(stats_minimal)
dataframe_summary_events_HL_int(data = stats_minimal,
data_annotation = metadata(stats_minimal)$annot[[1]])
```

`dataframe_summary_events_ps_itss` %

==========

dataframe_summary_events_ps_itss

dataframe_summary_events_ps_itss creates one table with all events between the segments.

Description

The dataframe_summary_events_ps_itss creates one table with the following columns: event, features, p_value, event_position, event_duration, position, region, gene, locus_tag, strand, TU, segment_1, segment_2, length, velocity_ratio.

Usage

```
dataframe_summary_events_ps_itss(data, data_annotation)
```

Arguments

`data`	SummarizedExperiment: the input data frame with correct format.
`data_annotation`	dataframe: dataframe from processed gff3 file.

Value

event: String, event type.

p_value: Integer, p_value of the event.

p_adjusted: Integer, p_value adjusted.

event_position: Integer, the position middle between 2 fragments with an event.

velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments.

feature_type: String, region annotation covering the fragments.

gene: String, gene annotation covering the fragments.

locus_tag: String, locus_tag annotation covering the fragments.

strand: Boolean. The bin/probe specific strand (+/-).

TU: String, The overarching transcription unit.

segment_1: String, the first segment of the event, includes the segment, TU, delay fragment in case of ps or iTSS_I. The rest of the events include HL fragment and could be extended intensity fragment.

segment_2: String, the second fragment of the two of fragments subjected to analysis.

event_duration: Integer, the difference (min) between 2 delay fragment when ps or iTSS_I happen.

gap_fragments: Integer, length in position (nt), calculated by the difference between the last position of the first fragment and the first position of the second fragment.

features: Integer, number of fragements involved on the event

Examples

```
data(stats_minimal)
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
dataframe_summary_events_ps_itss(data = stats_minimal,
data_annotation = metadata(stats_minimal)$annot[[1]])
```

dataframe_summary_events_velocity
%

==========

dataframe_summary_events_velocity

dataframe_summary_events_velocity creates one table with all events between the segments.

Description

The dataframe_summary_events_velocity creates one table with the following columns: event, features, p_value, event_position, event_duration, position, region, gene, locus_tag, strand, TU, segment_1, segment_2, length, velocity_ratio.

Usage

```
dataframe_summary_events_velocity(data, data_annotation)
```

Arguments

data	SummarizedExperiment: the input data frame with correct format.
data_annotation	dataframe: dataframe from processed gff3 file.

Value

event: String, event type.

p_value: Integer, p_value of the event.

p_adjusted: Integer, p_value adjusted.

event_position: Integer, the position of event, calculated dividing the last position of the first fragment and the first position of the next fragment on 2.

velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments

feature_type: String, region annotation covering the fragments.

gene: String, gene annotation covering the fragments.

locus_tag: String, locus_tag annotation covering the fragments.

strand: Boolean. The bin/probe specific strand (+/-).

TU: String, The overarching transcription unit.

segment_1: String, the first segment of the event, includes the segment, TU, delay fragment in case of ps or iTSS_I. The rest of the events include HL fragment and could be extended intensity fragment.

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the difference (min) between 2 delay fragment when ps or iTSS_I happen.

gap_fragments: Integer, length in position (nt), calculated by the difference between the last position of the first fragment and the first position of the second fragment.

features: Integer, number of fragements involved on the event.

Examples

```
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
data(stats_minimal)
dataframe_summary_events_velocity(data = stats_minimal,
data_annotation = metadata(stats_minimal)$annot[[1]])
```

---

dataframe_summary_TI *%*

==========

dataframe_summary_TI

dataframe_summary_TI creates one table with all TI fragments,p_value and the coordinates

---

Description

The dataframe_summary creates one table with the following columns: event, TI_fragment, TI_factor, TI_fragments_TU, p_value, feature_type, gene, locus_tag, strand, TU, features, event_position, position_1 and position_2.

Usage

```
dataframe_summary_TI(data, input)
```

Arguments

data	SummarizedExperiment: the input data frame with correct format.
input	dataframe: dataframe from event_dataframe function.

Value

event: String, event type, transcription interference.

TI_fragment: String, the fragment with TI.

TI_termination_factor: String, the factor of TI fragment.

p_value: Integer, p_value of the event.

p_adjusted: Integer, p_value adjusted.

feature_type: String, region annotation covering the fragments.

gene: String, gene annotation covering the fragments.

locus_tag: String, locus_tag annotation covering the fragments.

strand: Boolean. The bin/probe specific strand (+/-).

TU: String, The overarching transcription unit.

Features: Integer, The number of segments within the TI.

event_position: Integer, the position middle between 2 TI fragments.

position_1: String, the first position of TI fragment, if 2 fragments, first position is from the first fragment.

position_2: String, the last position of TI fragment, if 2 fragments, last position is from the second fragment.

WIP

Examples

```
data(stats_minimal)
data(res_minimal)
dataframe_summary_TI(data = stats_minimal, input = res_minimal)
```

`event_dataframe` *%*

==========

event_dataframe

event_dataframe creates a dataframe only with events

Description

event_dataframe creates a dataframe connecting segments, events and the annotation.

Usage

```
event_dataframe(data, data_annotation)
```

Arguments

`data`	dataframe: the probe based data frame.
`data_annotation`	dataframe: the coordinates are extracted from gff3

Details

The functions used are:

position_function: adds the specific position of ps or iTSS event.

annotation_function_event: adds the events to the annotated genes.

annotation file needs to be supplied. Strand is indicated in case of stranded data The event_dataframe selects columns with statistical features. ID, position, strand and TU columns are required.

Value

A dataframe with unique intensity fragments

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: Boolean. The bin/probe specific strand (+/-)

TU: String, The overarching transcription unit

position: Integer, position of the bin/probe on the genome

segment: String, the bin/probe segment on the genome

FC_fragment_intensity: String, the fragments subjected to fold change

FC_intensity: Integer, the fold change value of 2 intensity fragments

p_value_intensity: Integer, p_value of the FC_intensity

FC_fragment_HL: String, the fragments subjected to fold change

FC_HL: Integer, the fold change value of 2 HL fragments

p_value_HL: Integer, p_value of the FC_HL

FC_HL_FC_intensity_fragment: String, fragments subjected to FC_HL/FC_intensity

FC_HL_FC_intensity: Integer, the value of FC_HL/FC_intensity

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

p_value_manova: Integer, p_value of the event FC_HL/FC_intensity

synthesis_ratio: Integer, the value correspomding to synthesis rate

synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS

pausing_site: Boolean, presence or absence of pausing_site event (ps)

iTSS_I: Boolean, presence or absence of internal starting site event (iTSS_I)

ps_ts_fragment: String, fragments involved on the event ps or iTSS_I

event_position: Integer, the position middle between 2 fragments with an event

event_duration: Integer, the duration between two delay fragments

delay: Integer, the delay value of the bin/probe

half_life: The half-life of the bin/probe

intensity: The relative intensity at time point 0

delay_frg_slope: Integer, the slope value of the fit through the respective delay fragment

p_value_slope: Integer, the p_value added to the inp

Examples

```
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
data(stats_minimal)
event_dataframe(data = stats_minimal,
data_annotation = metadata(stats_minimal)$annot[[1]])
```

`example_input_e_coli` *An example SummarizedExperiment from E. coli An example SummarizedExperiment from RNA-seq containing information about the intensities at all time points (assay). Seqnames, IRanges and strand columns (rowRanges)and colData with time point series and replicates.*

Description

An example SummarizedExperiment from E. coli An example SummarizedExperiment from RNA-seq containing information about the intensities at all time points (assay). Seqnames, IRanges and strand columns (rowRanges)and colData with time point series and replicates.

Usage

```
data(example_input_e_coli)
```

Format

A assay:

0: relative intensities at 0 min

1: relative intensities at 1 min

10: relative intensities at 10 min

15: relative intensities at 15 min

2: relative intensities at 2 min

20: relative intensities at 20 min

3: relative intensities at 3 min

4: relative intensities at 4 min

5: relative intensities at 5 min

6: relative intensities at 6 min

8: relative intensities at 8 min

Source

https://github.com/CyanolabFreiburg/rifi

example_input_minimal *An artificial example SummarizedExperiment An example SummarizedExperiment containing information about the intensities at all time points (assay). Seqnames, IRanges and strand columns (rowRanges) and colData with time point series and replicates.*

Description

An artificial example SummarizedExperiment An example SummarizedExperiment containing information about the intensities at all time points (assay). Seqnames, IRanges and strand columns (rowRanges) and colData with time point series and replicates.

Usage

```
data(example_input_minimal)
```

Format

An object of class `RangedSummarizedExperiment` with 4 rows and 33 columns.

Source

https://github.com/CyanolabFreiburg/rifi

example_input_synechocystis_6803

An example input data frame from Synechocystis PCC 6803 A SummarizedExperiment from microarrays data containing information about the intensities at all time points (assay), Seqnames, IRanges and strand columns (rowRanges) and colData with time point series and averaged replicates.

Description

An example input data frame from Synechocystis PCC 6803 A SummarizedExperiment from microarrays data containing information about the intensities at all time points (assay), Seqnames, IRanges and strand columns (rowRanges) and colData with time point series and averaged replicates.

Usage

```
data(example_input_synechocystis_6803)
```

Format

Assay with 3000 rows and 10 variables:

0: relative intensities at 0 min

2: relative intensities at 2 min

4: relative intensities at 4 min

8: relative intensities at 8 min

16: relative intensities at 16 min

32: relative intensities at 32 min

64: relative intensities at 64 min

Source

https://github.com/CyanolabFreiburg/rifi

---

finding_PDD %

==========

finding_PDD

finding_PDD Flags potential candidates for post transcription decay

---

Description

'finding_PDD' uses 'score_fun_linear_PDD' to make groups by the difference to the slope. The slope is further checked for steepness to decide for PDD. '*PDD*' is added to the 'flag' column. Post transcription decay is characterized by a strong decrease of intensity by position. The rowRanges need to contain at least 'ID', 'intensity', 'position' and 'position_segment'!

Usage

```
finding_PDD(inp, cores = 1, pen = 2, pen_out = 1, thrsh = 0.001)
```

Arguments

inp	SummarizedExperiment: the input.
cores	integer: the number of assigned cores for the task
pen	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Advised to be kept at 2. Default is 2.
pen_out	numeric: an internal parameter for the dynamic programming. Higher values result in fewer possible outliers. Advised to be kept at 1. Default is 1.
thrsh	numeric: an internal parameter that allows fragments with slopes steeper than the thrsh to be flagged with '*PDD*'. Higher values result in fewer candidates. Advised to be kept at 0.001. Default is 0.001.

Value

The SummarizedExperiment object: with "*PDD*" added to the flag column.

Examples

```
data(preprocess_minimal)
finding_PDD(inp = preprocess_minimal, cores = 2, pen = 2,
pen_out = 1, thrsh = 0.001)
```

---

finding_TI %

==========

finding_TI

finding_TI flags potential candidates for transcription interference

---

Description

finding_TI uses 'score_fun_ave' to make groups by the mean of "probe_TI". "TI" is added to the "flag" column. TI is characterized by relative intensities at time points later than "0". The rowRanges need to contain at least "ID", "probe_TI" and "position_segment"!

Usage

```
finding_TI(inp, cores, pen = 10, thrsh = 0.5, add = 1000)
```

Arguments

inp	SummarizedExperiment: the input.
cores	integer: the number of assigned cores for the task
pen	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Advised to be kept at 10. Default is 10.
thrsh	numeric: an internal parameter that allows fragments with a certain amount of IDs with higher relative intensities at time points later than "0" to be flagged as "*TI*". Higher values result in fewer candidates. -0.5 is 25 %, 0 is 50%, 0.5 is 75%. Advised to be kept at 0.5. Default is 0.5.
add	integer: range of nucleotides before and after a potential TI event wherein IDs are fitted with the TI fit.

Value

The SummarizedExperiment object: with "*TI*" added to the flag column.

Examples

```
data(preprocess_minimal)
finding_TI(inp = preprocess_minimal, cores = 2, pen = 10, thrsh = 0.5,
add = 1000)
```

`fit_e_coli` *The result of rifi_fit for E.coli example data A SummarizedExperiment containing the output from rifi_fit as an extension of rowRanges and metadata.*

Description

The result of rifi_fit for E.coli example data A SummarizedExperiment containing the output from rifi_fit as an extension of rowRanges and metadata.

Usage

```
data(fit_e_coli)
```

Format

Three data frames with 290 rows and 10 variables, 155 rows and 5 variables, and 135 rows and 9 variables are generated. The columns of the first data frame are added to the rowRanges and the rest are added as metadata.

inp: The SummarizedExperiment:

- **ID:** The bin/probe specific ID
- **position:** The bin/probe specific position
- **intensity:** The relative intensity at time point 0
- **probe_TI:** An internal value to determine which fitting model is applied
- **flag:** Information on which fitting model is applied
- **postion_segment:** The position based segment
- **delay:** The delay value of the bin/probe
- **half_life:** The half-life of the bin/probe
- **TI_termination_factor:** String, the factor of TI fragment

fit_obj_STD: the fit object for the standard fit:

- **ID:** The bin/probe specific ID
- **delay:** The delay value of the bin/probe
- **half_life:** The half-life of the bin/probe
- **inty_S0:** The relative intensity at time point 0
- **intyf:** The background value of the fit

fit_obj_TI: the fit object for the TI fit:

- **delay:** The delay value of the bin/probe

ti_delay: The ti-delay value of the bin/probe

half_life: The half-life of the bin/probe

ti_value: The ti-value of the bin/probe

TI_termination_factor: String, the factor of TI fragment

synthesis_rate: The synthesis rate of the bin/probe

TI_background: The background value of the fit

position: The bin/probe specific position

ID: The bin/probe specific ID

Source

https://github.com/CyanolabFreiburg/rifi

`fit_minimal` *The artificial result of rifi_fit for artificial example data A SummarizedExperiment containing the output from rifi_fit.*

Description

The artificial result of rifi_fit for artificial example data A SummarizedExperiment containing the output from rifi_fit.

Usage

```
data(fit_minimal)
```

Format

An object of class `RangedSummarizedExperiment` with 4 rows and 33 columns.

Source

https://github.com/CyanolabFreiburg/rifi

`fit_synechocystis_6803` *The result of rifi_fit for Synechocystis 6803 example data A SummarizedExperiment containing the output from rifi_fit as an extension of rowRanges and metadata.*

Description

The result of rifi_fit for Synechocystis 6803 example data A SummarizedExperiment containing the output from rifi_fit as an extension of rowRanges and metadata.

Usage

```
data(fit_synechocystis_6803)
```

Format

Three data frames with 3000 rows and 10 variables, 2811 rows and 5 variables, and 189 rows and 9 variable are generated. The columns of the first data frame are added to the rowRanges and the rest are added as metadata.

inp: the SummarizedExperiment:

- **ID:** The bin/probe specific ID
- **position:** The bin/probe specific position
- **strand:** The bin/probe specific strand
- **intensity:** The relative intensity at time point 0
- **probe_TI:** An internal value to determine which fitting model is applied
- **flag:** Information on which fitting model is applied
- **postion_segment:** The position based segment
- **delay:** The delay value of the bin/probe
- **half_life:** The half-life of the bin/probe
- **TI_termination_factor:** String, the factor of TI fragment

fit_obj_STD: the fit object for the standard fit:

- **ID:** The bin/probe specific ID
- **delay:** The delay value of the bin/probe
- **half_life:** The half-life of the bin/probe
- **inty_S0:** The relative intensity at time point 0
- **intyf:** The background value of the fit

fit_obj_TI: the fit object for the TI fit:

- **delay:** The delay value of the bin/probe
- **ti_delay:** The ti-delay value of the bin/probe
- **half_life:** The half-life of the bin/probe
- **ti_value:** The ti-value of the bin/probe

TI_termination_factor: String, the factor of TI fragment

synthesis_rate: The synthesis rate of the bin/probe

TI_background: The background value of the fit

position: The bin/probe specific position

ID: The bin/probe specific ID

Source

https://github.com/CyanolabFreiburg/rifi

---

fold_change %

==========

fold_change

fold_change sets a fold-change ratio between the neighboring fragments of Half-life (HL) and intensity

---

Description

fold_change sets fold change on intensity and fold change HL fragments of two successive fragments. Two intensity fragments could belong to one HL fragment. This function sets first the borders using the position and applies the fold change ratio between the neighboring fragments of HL and those from intensity log2(intensity frgA/intensity frgB/half-life frgA/half-life frgB). All grepped fragments are from the same TU excluding outliers.

Usage

```
fold_change(inp)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.

Details

The function used is: synthesis_r_Function: assigns events depending on the ratio between HL and intensity of two consecutive fragments. intensity(int) = synthesis rate(k)/decay(deg) (steady state), int1/int2 = k1/deg1*deg2/k2 int1 * (deg1/int2) * deg2 = k1/k2 => synthesis ratio. In case of synthesis ratio is: synthesis ratio > 0 -> New start synthesis ratio < 0 -> Termination

Value

the SummarizedExperiment with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.

flag: Information on which fitting model is applied.

position_segment: The position based segment.

delay: The delay value of the bin/probe.

half_life: The half-life of the bin/probe.

TI_termination_factor: String, the factor of TI fragment.

delay_fragment: The delay fragment the bin belongs to.

velocity_fragment: The velocity value of the respective delay fragment.

intercept: The vintercept of fit through the respective delay fragment.

slope: The slope of the fit through the respective delay fragment.

HL_fragment: The half-life fragment the bin belongs to.

HL_mean_fragment: The mean half-life value of the respective half-life fragment.

intensity_fragment: The intensity fragment the bin belongs to.

intensity_mean_fragment: The mean intensity value of the respective intensity fragment.

TU: The overarching transcription unit.

TI_termination_fragment: The TI fragment the bin belongs to.

TI_mean_termination_factor: The mean termination factor of the respective TI fragment.

seg_ID: The combined ID of the fragment.

pausing_site: presence of pausing site indicated by +/-.

iTSS_I: presence of iTSS_I indicated by +/-.

ps_ts_fragment: The fragments involved in pausing site or iTSS_I.

event_duration: Integer, the duration between two delay fragments.

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I.

p_value_slope: Integer, the p_value added to the inp.

delay_frg_slope: Integer, the slope value of the fit through the respective delay fragment.

velocity_ratio: Integer, the ratio value of velocity from 2 delay fragments.

event_position: Integer, position of the event added to the input.

FC_HL: Integer, the fold change value of 2 HL fragments.

FC_fragment_HL: String, the fragments corresponding to HL fold change.

p_value_HL: Integer, the p_value added to the input of 2 HL fragments.

FC_intensity: Integer, the fold change value of 2 intensity fragments.

FC_fragment_intensity: String, the fragments corresponding to intensity fold change.

p_value_intensity: Integer, the p_value added to the input of 2 intensity fragments.

synthesis_ratio: Integer, the value correspomding to synthesis rate.

synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS.

FC_HL_intensity: Integer, the value corresponding to HL and intensity fold change.

FC_HL_intensity_fragment: String, the fragments corresponding to intensity and HL fold change.

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity,position of the half-life fragment is adapted to intensity fragment.

Examples

```
data(stats_minimal)
fold_change(inp = stats_minimal)
```

---

fragmentation_e_coli *The result of rifi_fragmentation for E.coli example data A SummarizedExperiment containing the output from rifi_fragmentation as an extension of rowRanges*

Description

The result of rifi_fragmentation for E.coli example data A SummarizedExperiment containing the output from rifi_fragmentation as an extension of rowRanges

Usage

```
data(fragmentation_e_coli)
```

Format

rowRanges of the SummarizedExperiment with 290 rows and 22 variables:

ID: The bin/probe specific ID

position: The bin/probe specific position

intensity: The relative intensity at time point 0

probe_TI: An internal value to determine which fitting model is applied

flag: Information on which fitting model is applied

position_segment: The position based segment

delay: The delay value of the bin/probe

half_life: The half-life of the bin/probe

TI_termination_factor: String, the factor of TI fragment

delay_fragment: The delay fragment the bin belongs to

velocity_fragment: The velocity value of the respective delay fragment

intercept: The vintercept of fit through the respective delay fragment

slope: The slope of the fit through the respective delay fragment

HL_fragment: The half-life fragment the bin belongs to

HL_mean_fragment: The mean half-life value of the respective half-life fragment

intensity_fragment: The intensity fragment the bin belongs to

intensity_mean_fragment: The mean intensity value of the respective intensity fragment

TU: The overarching transcription unit

TI_termination_fragment: The TI fragment the bin belongs to

TI_mean_termination_factor: The mean termination factor of the respective TI fragment

seg_ID: The combined ID of the fragment

Source

https://github.com/CyanolabFreiburg/rifi

`fragmentation_minimal` *The result of rifi_fragmentation for artificial example data A SummarizedExperiment containing the output from rifi_fragmentation as an extension of rowRanges and metadata.*

Description

The result of rifi_fragmentation for artificial example data A SummarizedExperiment containing the output from rifi_fragmentation as an extension of rowRanges and metadata.

Usage

```
data(fragmentation_minimal)
```

Format

An object of class `RangedSummarizedExperiment` with 24 rows and 33 columns.

Source

https://github.com/CyanolabFreiburg/rifi

`fragmentation_synechocystis_6803` *The result of rifi_fragmentation for Synechocystis 6803 example data A SummarizedExperiment containing the output from rifi_fragmentation as an extension fo rowRanges*

Description

The result of rifi_fragmentation for Synechocystis 6803 example data A SummarizedExperiment containing the output from rifi_fragmentation as an extension fo rowRanges

Usage

```
data(fragmentation_synechocystis_6803)
```

Format

rowRanges of the SummarizedExperiment:

ID: The bin/probe specific ID
position: The bin/probe specific position
intensity: The relative intensity at time point 0
probe_TI: An internal value to determine which fitting model is applied
flag: Information on which fitting model is applied
position_segment: The position based segment
delay: The delay value of the bin/probe
half_life: The half-life of the bin/probe
TI_termination_factor: String, the factor of TI fragment
delay_fragment: The delay fragment the bin belongs to
velocity_fragment: The velocity value of the respective delay fragment
intercept: The vintercept of fit through the respective delay fragment
slope: The slope of the fit through the respective delay fragment
HL_fragment: The half-life fragment the bin belongs to
HL_mean_fragment: The mean half-life value of the respective half-life fragment
intensity_fragment: The intensity fragment the bin belongs to
intensity_mean_fragment: The mean intensity value of the respective intensity fragment
TU: The overarching transcription unit
TI_termination_fragment: The TI fragment the bin belongs to
TI_mean_termination_factor: The mean termination factor of the respective TI fragment
seg_ID: The combined ID of the fragment

Source

https://github.com/CyanolabFreiburg/rifi

fragment_delay %

==========

fragment_delay

fragment_delay performs the delay fragmentation

Description

fragment_delay makes delay_fragments based on position_segments and assigns all gathered information to the SummarizedExperiment object. The columns "delay_fragment", "velocity_fragment", "intercept" and "slope" are added. fragment_delay makes delay_fragments, assigns slopes, which are 1/velocity at the same time, and intercepts for the TU calculation. The function used is: score_fun_linear the input is the SummarizedExperiment object. pen is the penalty for new fragments in the dynamic programming, pen_out is the outlier penalty.

Usage

```
fragment_delay(inp, cores = 1, pen, pen_out)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
cores	cores: integer: the number of assigned cores for the task.
pen	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.
pen_out	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.

Value

the SummarizedExperiment object:

ID:	The bin/probe specific ID.
position:	The bin/probe specific position.
intensity:	The relative intensity at time point 0.
probe_TI:	An internal value to determine which fitting model is applied.
flag:	Information on which fitting model is applied.
position_segment:	The position based segment.
delay:	The delay value of the bin/probe.
half_life:	The half-life of the bin/probe.

`TI_termination_factor:`
: String, the factor of TI fragment.

`delay_fragment:`
: The delay fragment the bin belongs to.

`velocity_fragment:`
: The velocity value of the respective delay fragment.

`intercept:`
: The vintercept of fit through the respective delay fragment.

`slope:`
: The slope of the fit through the respective delay fragment.

Examples

```
data(fragmentation_minimal)
fragment_delay(inp = fragmentation_minimal, cores = 2, pen = 2, pen_out = 1)
```

`fragment_HL` *%*

==========

fragment_HL

fragment_HL performs the half_life fragmentation

Description

fragment_HL makes HL_fragments based on delay_fragments and assigns all gathered information to the SummarizedExperiment object.

Usage

```
fragment_HL(inp, cores = 1, pen, pen_out)
```

Arguments

`inp`
: SummarizedExperiment: the input data frame with correct format.

`cores`
: integer: the number of assigned cores for the task.

`pen`
: numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.

`pen_out`
: numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.

Details

The columns "HL_fragment" and "HL_mean_fragment" are added.

fragment_HL makes half-life_fragments and assigns the mean of each fragment.

The function used is:

.score_fun_ave.

The input the SummarizedExperiment object.

pen is the penalty for new fragments in the dynamic programming, pen_out is the outlier penalty.

Value

The SummarizedExperiment object:

ID: The bin/probe specific ID

position: The bin/probe specific position

intensity: The relative intensity at time point 0

probe_TI: An internal value to determine which fitting model is applied

flag: Information on which fitting model is applied

position_segment: The position based segment

delay: The delay value of the bin/probe

half_life: The half-life of the bin/probe

TI_termination_factor: String, the factor of TI fragment

delay_fragment: The delay fragment the bin belongs to

velocity_fragment: The velocity value of the respective delay fragment

intercept: The vintercept of fit through the respective delay fragment

slope: The slope of the fit through the respective delay fragment

HL_fragment: The half-life fragment the bin belongs to

HL_mean_fragment: The mean half-life value of the respective half-life fragment

Examples

```
data(fragmentation_minimal)
fragment_HL(inp = fragmentation_minimal, cores = 2, pen = 2, pen_out = 1)
```

fragment_inty %

==========

fragment_inty

fragment_inty performs the intensity fragmentation

Description

fragment_inty makes intensity_fragments based on HL_fragments and assigns all gathered information to the SummarizedExperiment object.

Usage

```
fragment_inty(inp, cores = 1, pen, pen_out)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
cores	cores: integer: the number of assigned cores for the task.
pen	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.
pen_out	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.

Details

The columns "intensity_fragment" and "intensity_mean_fragment" are added.

fragment_inty makes intensity_fragments and assigns the mean of each fragment.

The function used is:

.score_fun_ave.

The input is the the SummarizedExperiment object.

pen is the penalty for new fragments in the dynamic programming, pen_out is the outlier penalty.

Value

The SummarizedExperiment object:

ID:	The bin/probe specific ID
position:	The bin/probe specific position
intensity:	The relative intensity at time point 0
probe_TI:	An internal value to determine which fitting model is applied
flag:	Information on which fitting model is applied

position_segment:
: The position based segment

delay:
: The delay value of the bin/probe

half_life:
: The half-life of the bin/probe

TI_termination_factor:
: String, the factor of TI fragment

delay_fragment:
: The delay fragment the bin belongs to

velocity_fragment:
: The velocity value of the respective delay fragment

intercept:
: The vintercept of fit through the respective delay fragment

slope:
: The slope of the fit through the respective delay fragment

HL_fragment:
: The half-life fragment the bin belongs to

HL_mean_fragment:
: The mean half-life value of the respective half-life fragment

intensity_fragment:
: The intensity fragment the bin belongs to

intensity_mean_fragment:
: The mean intensity value of the respective intensity fragment

Examples

```
data(fragmentation_minimal)
fragment_inty(inp = fragmentation_minimal, cores = 2, pen = 2, pen_out = 1)
```

---

fragment_TI %

==========

fragment_TI

fragment_TI performs the TI fragmentation

Description

fragment_TI makes TI_fragments based on TUs and assigns all gathered information to the SummarizedExperiment object. The columns TI_termination_fragment" and the TI_mean_termination_factor are added.

Usage

```
fragment_TI(inp, cores = 1, pen, pen_out)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
cores	cores: integer: the number of assigned cores for the task.
pen	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.
pen_out	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.

Details

The function used is:

.score_fun_ave.

The input is the SummarizedExperiment object.

pen is the penalty for new fragments in the dynamic programming, pen_out is the outlier penalty.

Value

The SummarizedExperiment object:

ID:	The bin/probe specific ID
position:	The bin/probe specific position
intensity:	The relative intensity at time point 0
probe_TI:	An internal value to determine which fitting model is applied
flag:	Information on which fitting model is applied
position_segment:	The position based segment
delay:	The delay value of the bin/probe
half_life:	The half-life of the bin/probe
TI_termination_factor:	String, the factor of TI fragment
delay_fragment:	The delay fragment the bin belongs to
velocity_fragment:	The velocity value of the respective delay fragment
intercept:	The vintercept of fit through the respective delay fragment
slope:	The slope of the fit through the respective delay fragment
HL_fragment:	The half-life fragment the bin belongs to
HL_mean_fragment:	The mean half-life value of the respective half-life fragment
intensity_fragment:	The intensity fragment the bin belongs to
intensity_mean_fragment:	The mean intensity value of the respective intensity fragment

`TI_termination_fragment:`
: The TI fragment the bin belongs to

`TI_mean_termination_factor:`
: The mean termination factor of the respective TI fragment

Examples

```
data(fragmentation_minimal)
fragment_TI(inp = fragmentation_minimal, cores = 2, pen = 2, pen_out = 1)
```

`gff3_preprocess` *%*

==========

gff3_preprocess

gff3_preprocess process gff3 file from database for multiple usage

---

Description

gff3_preprocess processes the gff3 file extracting gene names and locus_tag from all coding regions (CDS), UTRs/ncRNA/asRNA are also extracted if available.

Usage

```
gff3_preprocess(path)
```

Arguments

`path`	path: path to the directory containing the gff3 file.

Details

The resulting dataframe contains region, positions, strand, gene and locus_tag.

Value

A list with 2 items:

data annotation: **region:** String, the region from the gff file
start: Integer, the start of the annotation
end: Integer, the end of the annotation
strand: Boolean, the strand of the annotation
gene: String, the annotated gene name
locus_tag: String, the annotated locus tag

genome length: a numeric vector containing the length of the genome

Examples

```
gff3_preprocess(
path = gzfile(system.file("extdata", "gff_e_coli.gff3.gz", package = "rifi"))
)
```

---

make_df %

==========

make_df

make_df adds important columns to the SummarizedExperiment object

---

Description

'make_df' adds to the SummarizedExperiment object with the columns: "intensity", "probe_TI" and "flag".

Usage

```
make_df(inp, cores = 1, bg = 0, rm_FLT = TRUE)
```

Arguments

inp	SummarizedExperiment: the (checked) input.
cores	integer: the number of assigned cores for the task.
bg	numeric: threshold over which the last timepoint has to be fitted with the above background mode.
rm_FLT	logical: remove IDs where all replicates are marked as filtered. Default is FALSE.

Details

The replicates are collapsed into their respective means.

"intensity" is the mean intensity from time point 0.

"probe_TI" is a value needed for the distribution for the different fitting models.

"flag" contains information or the distribution for the different fitting models.

Probes that don't reach the background level expression are flagged as "*ABG*" ("above background"). This is only needed for microarray data and is controlled by the bg parameter. The default for bg = 0, resulting in all probes to be above background (0 is advised for RNAseq data).

Probes where all replicates were filtered in the optional filtration step can be fully removed by rm_FLT = TRUE! If you wish to keep all information in the assay set to FALSE!

Value

the SummarizedExperiment object: with intensity, probe_TI and flag added to the rowRanges.

Examples

```
data(preprocess_minimal)
make_df(inp = preprocess_minimal, cores = 2, bg = 0, rm_FLT = TRUE)
```

---

make_pen *%*

==========

make_pen

make_pen assigns automatically a penalties

Description

'make_pen' calls one of four available penalty functions to automatically assign penalties for the dynamic programming.

Usage

```
make_pen(
  inp,
  FUN,
  cores = 1,
  logs,
  dpt = 1,
  smpl_min = 10,
  smpl_max = 100,
  sta_pen = 0.5,
  end_pen = 4.5,
  rez_pen = 9,
  sta_pen_out = 0.5,
  end_pen_out = 3.5,
  rez_pen_out = 7
)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
FUN	function: one of the four bottom level functions (see details)
cores	integer: the number of assigned cores for the task
logs	numeric vector: the logbook vector.

dpt	integer: the number of times a full iteration cycle is repeated with a more narrow range based on the previous cycle. Default is 2.
smpl_min	integer: the smaller end of the sampling size. Default is 10.
smpl_max	integer: the larger end of the sampling size. Default is 100.
sta_pen	numeric: the lower starting penalty. Default is 0.5.
end_pen	numeric: the higher starting penalty. Default is 4.5.
rez_pen	numeric: the number of penalties iterated within the penalty range. Default is 9.
sta_pen_out	numeric: the lower starting outlier penalty. Default is 0.5.
end_pen_out	numeric: the higher starting outlier penalty. Default is 3.5.
rez_pen_out	numeric: the number of outlier penalties iterated within the outlier penalty range. Default is 7.

Details

The four functions to be called are:

1. fragment_delay_pen
2. fragment_HL_pen
3. fragment_inty_pen
4. fragment_TI_pen

These functions return the amount of statistically correct and statistically wrong splits at a specific pair of penalties. 'make_pen' iterates over many penalty pairs and picks the most suitable pair based on the difference between wrong and correct splits. The sample size, penalty range and resolution as well as the number of cycles can be customized. The primary start parameters create a matrix with n = rez_pen rows and n = rez_pen_out columns with values between sta_pen/sta_pen_out and end_pen/end_pen_out. The best penalty pair is picked. If dept is bigger than 1 the same process is repeated with a new matrix of the same size based on the result of the previous cycle. Only position segments with length within the sample size range are considered for the penalties to increase run time. Returns a penalty object (list of 4 objects) the first being the logbook.

Value

A list with 4 items:

logbook: Interger, the logbook vector containing all penalty information

penalties: Integer, a vetor with the respective penalty and outlier penalty

correct: Matrix, a matrix of the correct splits

wrong: Matrix, a matrix of the incorrect splits

Examples

```
data(fit_minimal)
make_pen(
  inp = fit_minimal, FUN = rifi:::fragment_HL_pen, cores = 2,
  logs = as.numeric(rep(NA, 8)), dpt = 1, smpl_min = 10, smpl_max = 50,
```

```
  sta_pen = 0.5, end_pen = 4.5, rez_pen = 9, sta_pen_out = 0.5,
  end_pen_out = 3.5, rez_pen_out = 7
)
```

---

nls2_fit %

==========

nls2_fit

nls2_fit estimates decay for each probe or bin

Description

nls2_fit uses nls2 function to fit a probe or bin using intensities of the time series data from different time point. nls2 uses different starting values through expand grid and selects the best fit. Different filters could be applied prior fitting to the model.

Usage

```
nls2_fit(
  inp,
  cores = 1,
  decay = seq(0.01, 0.11, by = 0.02),
  delay = seq(0, 10, by = 0.1),
  k = seq(0.1, 1, 0.2),
  bg = 0.2
)
```

Arguments

inp	SummarizedExperiment: the input with correct format.
cores	integer: the number of assigned cores for the task.
decay	numeric vector: A sequence of starting values for the decay. Default is seq(.08, 0.11, by=.02)
delay	numeric vector: A sequence of starting values for the delay. Default is seq(0,10, by=.1)
k	numeric vector: A sequence of starting values for the synthesis rate. Default is seq(0.1,1,0.2)
bg	numeric vector: A sequence of starting values. Default is 0.2.

Details

To apply nls2_fit function, prior filtration could applied.

1. generic_filter_BG: filter probes with intensities below background using threshold. Those probes are filtered.
2. filtration_below_backg: additional functions exclusive to microarrays could be applied. Its very strict to the background (not recommended in usual case).
3. filtration_above_backg: selects probes with a very high intensity and above the background (recommended for special transcripts). Probes are flagged with "*ABG*". Those transcripts are usually related to a specific function in bacteria. This filter selects all probes with the same ID, the mean is applied, the last time point is selected and compared to the threshold.

The model used estimates the delay, decay, intensity of the first time point (synthesis rate/decay) and the background. The coefficients are gathered in vectors with the corresponding IDs. Absence of the fit or a very bad fit are assigned with NA. In case of probes with very high intensities and above the background, the model used makes abstinence of background coefficient. The output of all coefficients is saved in the metadata. The fits are plotted using the function_plot_fit.r through rifi_fit.

Value

the SummarizedExperiment object: with delay and decay added to the rowRanges. The full fit data is saved in the metadata as "fit_STD".

delay: Integer, the delay value of the bin/probe

half_life: Integer, the half-life of the bin/probe

Examples

```
data(preprocess_minimal)
nls2_fit(inp = preprocess_minimal, cores = 2)
```

---

penalties_e_coli

The result of rifi_penalties for E.coli example data. A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata.

Description

The result of rifi_penalties for E.coli example data. A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata.

Usage

```
data(penalties_e_coli)
```

Format

A list with 5 items:

logbook: The logbook vector containing all penalty information

pen_obj_delay: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **delay_penalties:** a vetor with the delay penalty and delay outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

pen_obj_HL: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **HL_penalties:** a vetor with the half-life penalty and half-life outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

pen_obj_inty: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **inty_penalties:** a vetor with the intensity penalty and intensity outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

pen_obj_TI: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **TI_penalties:** a vetor with the TI penalty and TI outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

Source

https://github.com/CyanolabFreiburg/rifi

---

penalties_minimal

The result of rifi_penalties for artificial example data A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata.

Description

The result of rifi_penalties for artificial example data A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata.

Usage

```
data(penalties_minimal)
```

Format

An object of class `RangedSummarizedExperiment` with 24 rows and 33 columns.

Source

https://github.com/CyanolabFreiburg/rifi

`penalties_synechocystis_6803`

The result of rifi_penalties for Synechocystis 6803 example data. A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata.

Description

The result of rifi_penalties for Synechocystis 6803 example data. A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata.

Usage

```
data(penalties_synechocystis_6803)
```

Format

A list with 5 items:

logbook: The logbook vector containing all penalty information

pen_obj_delay: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **delay_penalties:** a vetor with the delay penalty and delay outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

pen_obj_HL: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **HL_penalties:** a vetor with the half-life penalty and half-life outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

pen_obj_inty: A list with 4 items:

- **logbook:** The logbook vector containing all penalty information
- **inty_penalties:** a vetor with the intensity penalty and intensity outlier penalty
- **correct:** a matrix of the correct splits
- **wrong:** a matrix of the incorrect splits

pen_obj_TI: A list with 4 items:

logbook: The logbook vector containing all penalty information
TI_penalties: a vetor with the TI penalty and TI outlier penalty
correct: a matrix of the correct splits
wrong: a matrix of the incorrect splits

Source

https://github.com/CyanolabFreiburg/rifi

---

predict_ps_itss %

==========

predict_ps_itss

predict_ps_itss predicts pausing sites (ps) and internal starting sites (ITSS) between delay fragments.

Description

predict_ps_itss predicts ps and ITSS within the same TU. Neighboring delay segments are compared to each other by positioning the intercept of the second segment into the first segment using slope and intercept coefficients.

Usage

```
predict_ps_itss(inp, maxDis = 300)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
maxDis	integer: the maximal distance allowed between two successive fragments.

Details

predict_ps_itss uses 3 steps to identify ps and ITSS:

1. select unique TU.
2. select from the input dataframe the columns: ID, position, strand, delay. delay fragment, TU and slope coordinates, velocity_fragment and intercept.
3. select delay segments in the TU.
4. loop into all delay segments and estimate the coordinates of the last point of the first segment using the coefficients of the second segment and vice versa. We get two predicted positions, the difference between them is compared to the threshold.

In case the strand is "-", additional steps are added:

The positions of both segments are ordered from the last position to the first one.

All positions are merged in one column and subtracted from the maximum position. the column is split in 2. The first and second correspond to the positions of the first and second segments respectively.

Both segments are subjected to lm fit and the positions predicted are used on the same way as the opposite strand.

If the difference between the positions predicted is lower than negative threshold, ps is assigned otherwise, and if the difference is higher than the positive threshold, ITSS is assigned.

Value

The SummarizedExperiment with the columns regarding statistics:

ID: The bin/probe specific ID.

position: The bin/probe specific position.

strand: The bin/probe specific strand.

intensity: The relative intensity at time point 0.

probe_TI: An internal value to determine which fitting model is applied.

flag: Information on which fitting model is applied.

position_segment: The position based segment.

delay: The delay value of the bin/probe.

half_life: The half-life of the bin/probe.

TI_termination_factor: String, the factor of TI fragment.

delay_fragment: The delay fragment the bin belongs to.

velocity_fragment: The velocity value of the respective delay fragment.

intercept: The vintercept of fit through the respective delay fragment.

slope: The slope of the fit through the respective delay fragment.

HL_fragment: The half-life fragment the bin belongs to.

HL_mean_fragment: The mean half-life value of the respective half-life fragment.

intensity_fragment: The intensity fragment the bin belongs to.

intensity_mean_fragment: The mean intensity value of the respective intensity fragment.

TU: The overarching transcription unit.

TI_termination_fragment: The TI fragment the bin belongs to.

TI_mean_termination_factor: The mean termination factor of the respective TI fragment.

seg_ID: The combined ID of the fragment.

pausing_site: presence of pausing site indicated by +/-.

iTSS_I: presence of iTSS_I indicated by +/-.

ps_ts_fragment: The fragments involved in pausing site or iTSS_I.

event_duration: Integer, the duration between two delay fragments.

Examples

```
data(fragmentation_minimal)
predict_ps_itss(inp = fragmentation_minimal, maxDis = 300)
```

`preprocess_e_coli` *The result of rifi_preprocess for E.coli example data A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata. A list containing the output from rifi_preprocess, including the inp and the modified input_df.*

Description

The result of rifi_preprocess for E.coli example data A SummarizedExperiment containing the output from rifi_penalties including the logbook and the four penalty objects as metadata. A list containing the output from rifi_preprocess, including the inp and the modified input_df.

Usage

```
data(preprocess_e_coli)
```

Format

A SummarizedExperiment:

inp: the SummarizedExperiment:

- **ID:** The bin/probe specific ID
- **position:** The bin/probe specific position
- **intensity:** The relative intensity at time point 0
- **probe_TI:** An internal value to determine which fitting model is applied
- **flag:** Information on which fitting model is applied
- **postion_segment:** The position based segment

fit_obj_TI: the fit object for the TI fit:

- **0:** relative intensities at 0 min
- **1:** relative intensities at 1 min
- **10:** relative intensities at 10 min
- **15:** relative intensities at 15 min
- **2:** relative intensities at 2 min
- **20:** relative intensities at 20 min
- **3:** relative intensities at 3 min
- **4:** relative intensities at 4 min
- **5:** relative intensities at 5 min
- **6:** relative intensities at 6 min
- **8:** relative intensities at 8 min

ID: The bin/probe specific ID
position: The bin/probe specific position
filtration: indicator wether the replicate is filtered or not

Source

https://github.com/CyanolabFreiburg/rifi

`preprocess_minimal` *The result of rifi_preprocess for artificial example data A SummarizedExperiment containing the output from rifi_preprocess*

Description

The result of rifi_preprocess for artificial example data A SummarizedExperiment containing the output from rifi_preprocess

Usage

```
data(preprocess_minimal)
```

Format

An object of class `RangedSummarizedExperiment` with 4 rows and 33 columns.

Source

https://github.com/CyanolabFreiburg/rifi

`preprocess_synechocystis_6803` *The result of rifi_preprocess for Synechocystis 6803 example data is a A SummarizedExperiment containing the output of rifi_preprocess as an extention to rowRanges*

Description

The result of rifi_preprocess for Synechocystis 6803 example data is a A SummarizedExperiment containing the output of rifi_preprocess as an extention to rowRanges

Usage

```
data(preprocess_synechocystis_6803)
```

Format

A SummarizedExperiment:

inp: the SummarizedExperiment:

ID: The bin/probe specific ID
position: The bin/probe specific position
strand: The bin/probe specific strand
intensity: The relative intensity at time point 0
probe_TI: An internal value to determine which fitting model is applied
flag: Information on which fitting model is applied
postion_segment: The position based segment

fit_obj_TI: the fit object for the TI fit:

0: relative intensities at 0 min
2: relative intensities at 2 min
4: relative intensities at 4 min
8: relative intensities at 8 min
16: relative intensities at 16 min
32: relative intensities at 32 min
64: relative intensities at 64 min
ID: The bin/probe specific ID
position: The bin/probe specific position
filtration: indicator wether the replicate is filtered or not

Source

https://github.com/CyanolabFreiburg/rifi

`res_minimal` *The result of event_dataframe for E.coli artificial example. A data frame combining the processed genome annotation and a SummarizedExperiment data from rifi_stats. The dataframe is*

Description

The result of event_dataframe for E.coli artificial example. A data frame combining the processed genome annotation and a SummarizedExperiment data from rifi_stats. The dataframe is

Usage

```
data(res_minimal)
```

Format

A list with 2 items:

region: the region from the gff file

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: the strand of the annotation

TU: The overarching transcription unit

position: The bin/probe specific position

FC_fragment_intensity: String, fragments involved in fold change between 2 intensity fragments

FC_intensity: Integer, the fold change value of 2 intensity fragments

p_value_intensity: p_value of the fold change of intensity fragments

FC_fragment_HL: Integer, the fold change value of 2 intensity fragments

FC_HL: Integer, the fold change value of 2 HL fragments

p_value_HL: p_value of the fold change of HL fragments

FC_HL_intensity_fragment: fragments involved on ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments

FC_HL_intensity: ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

p_value_Manova: p_value of the variance between two fold-changes, HL and intensity

synthesis_ratio: Integer, the value correspomding to synthesis rate

synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS

pausing_site: presence of pausing site indicated by +/-

iTSS_I: presence of iTSS_I indicated by +/-

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I

ps_ts_fragment: The fragments involved in pausing site or iTSS_I

event_position: Integer, the position middle between 2 fragments with an event

event_duration: Integer, the duration between two delay fragments

delay_frg_slope: the slope value of the respective delay fragment

p_value_slope: p_value of the slope

delay: The delay value of the bin/probe

half_life: The half-life of the bin/probe

intensity: The relative intensity at time point 0

Source

https://github.com/CyanolabFreiburg/rifi

rifi_fit %

==========

rifi_fit

rifi_fit wraps conveniently all fitting steps

Description

rifi_fit wraps the functions:

1. nls2_fit
2. TI_fit
3. plot_nls2_function
4. plot_singleProbe_function

Usage

```
rifi_fit(
  inp,
  cores = 1,
  viz = FALSE,
  restr = 0.2,
  decay = seq(0.08, 0.11, by = 0.02),
  delay = seq(0, 10, by = 0.1),
  k = seq(0.1, 1, 0.2),
  bg = 0.2,
  TI_k = seq(0, 1, by = 0.5),
  TI_decay = c(0.05, 0.1, 0.2, 0.5, 0.6),
  TI = seq(0, 1, by = 0.5),
  TI_delay = seq(0, 2, by = 0.5),
  TI_rest_delay = seq(0, 2, by = 0.5),
  TI_bg = 0
)
```

Arguments

inp	SummarizedExperiment: the input with correct format.
cores	integer: the number of assigned cores for the task.
viz	logical: whether to visualize the output.
restr	numeric: a parameter that restricts the freedom of the fit to avoid wrong TI-term_factors, ranges from 0 to 0.2
decay	numeric vector: A sequence of starting values for the decay. Default is seq(.08, 0.11, by=.02)

delay	numeric vector: A sequence of starting values for the delay. Default is seq(0,10, by=.1)
k	numeric vector: A sequence of starting values for the synthesis rate.Default is seq(0.1,1,0.2)
bg	numeric vector: A sequence of starting values. Default is 0.2.
TI_k	numeric vector: A sequence of starting values for the synthesis rate. Default is seq(0, 1, by = 0.5).
TI_decay	numeric vector: A sequence of starting values for the decay. Default is c(0.05, 0.1, 0.2, 0.5, 0.6).
TI	numeric vector: A sequence of starting values for the TI. Default is seq(0, 1, by = 0.5).
TI_delay	numeric vector: A sequence of starting values for the delay. Default is seq(0, 2, by = 0.5).
TI_rest_delay	numeric vector: A sequence of starting values. Default is seq(0, 2, by = 0.5).
TI_bg	numeric vector: A sequence of starting values. Default is 0.

Value

the SummarizedExperiment object: with delay, decay and TI_termination_factor added to the rowRanges. The full fit data is saved in the metadata as "fit_STD" and "fit_TI". A plot is given if viz = TRUE.

See Also

`nls2_fit`

`TI_fit`

`plot_nls2`

`plot_singleProbe`

Examples

```
data(preprocess_minimal)
rifi_fit(
  inp = preprocess_minimal,
  cores = 1, viz = FALSE, restr = 0.1,
  decay = seq(.08, 0.11, by = .02),
  delay = seq(0, 10, by = .1), k = seq(0.1, 1, 0.2), bg = 0.2,
  TI_k = seq(0, 1, by = 0.5), TI_decay = c(0.05, 0.1, 0.2, 0.5, 0.6),
  TI = seq(0, 1, by = 0.5), TI_delay = seq(0, 2, by = 0.5),
  TI_rest_delay = seq(0, 2, by = 0.5), TI_bg = 0
)
```

rifi_fragmentation %

==========

rifi_fragmentation

rifi_fragmentation wraps conveniently all fragmentation steps

Description

rifi_fragmentation is wrapper of the following functions:

1. fragment_delay
2. fragment_HL
3. fragment_inty
4. TUgether
5. fragment_TI

Usage

```
rifi_fragmentation(
  inp,
  cores = 1,
  pen_delay = NULL,
  pen_out_delay = NULL,
  pen_HL = NULL,
  pen_out_HL = NULL,
  pen_inty = NULL,
  pen_out_inty = NULL,
  pen_TU = NULL,
  pen_TI = NULL,
  pen_out_TI = NULL
)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
cores	integer: the number of assigned cores for the task.
pen_delay	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.
pen_out_delay	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.
pen_HL	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.

pen_out_HL	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.
pen_inty	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.
pen_out_inty	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.
pen_TU	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default -0.75.
pen_TI	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default is the auto generated value.
pen_out_TI	numeric: an internal parameter for the dynamic programming. Higher values result in fewer allowed outliers. Default is the auto generated value.

Value

the SummarizedExperiment object: with delay_fragment, HL_fragment, intensity_fragment, TI_termination_fragment and TU, and the respective values added to the rowRanges.

See Also

`fragment_delay`

`fragment_HL`

`fragment_inty`

`TUgether`

`fragment_TI`

Examples

```
data(penalties_minimal)
rifi_fragmentation(inp = penalties_minimal, cores = 2)
```

`rifi_penalties` %

==========

rifi_penalties

rifi_penalties wraps conveniently all penalty steps

Description

rifi_penalties wraps the functions:

1. make_pen,
2. viz_pen_obj

Usage

```
rifi_penalties(
  inp,
  details = FALSE,
  viz = FALSE,
  top_i = 25,
  cores = 1,
  dpt = 1,
  smpl_min = 10,
  smpl_max = 100,
  sta_pen = 0.5,
  end_pen = 4.5,
  rez_pen = 9,
  sta_pen_out = 0.5,
  end_pen_out = 4.5,
  rez_pen_out = 9
)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
details	logical: whether to return the penalty objects or just the logbook.
viz	logical: whether to visualize the output or not. Default is FALSE
top_i	integer: the number of top results visualized. Default is all.
cores	integer: the number of assigned cores for the task.
dpt	integer: the number of times a full iteration cycle is repeated with a more narrow range based on the previous cycle. Default is 2.
smpl_min	integer: the smaller end of the sampling size. Default is 10.
smpl_max	integer: the larger end of the sampling size. Default is 100.
sta_pen	numeric: the lower starting penalty. Default is 0.5.
end_pen	numeric: the higher starting penalty. Default is 4.5.
rez_pen	numeric: the number of penalties iterated within the penalty range. Default is 9.
sta_pen_out	numeric: the lower starting outlier penalty. Default is 0.5.
end_pen_out	numeric: the higher starting outlier penalty. Default is 3.5.
rez_pen_out	numeric: the number of outlier penalties iterated within the outlier penalty range. Default is 7.

Value

The SummarizedExperiment object: with the penalties in the logbook added to the metadata. Also adds logbook_details if details is TRUE, and plots the penalties if viz is TRUE.

See Also

make_pen

viz_pen_obj

Examples

```
data(fit_minimal)
rifi_penalties(
  inp = fit_minimal, details = FALSE, viz = FALSE,
  top_i = 25, cores = 2, dpt = 1, smpl_min = 10, smpl_max = 100,
  sta_pen = 0.5, end_pen = 4.5, rez_pen = 9, sta_pen_out = 0.5,
  end_pen_out = 4.5, rez_pen_out = 9
)
```

---

rifi_preprocess *%*

==========

rifi_preprocess

rifi_preprocess wraps conveniently all pre-processing steps

Description

rifi_preprocess wraps the functions:

1. check_input
2. make_df
3. function_seg
4. finding_PDD
5. finding_TI

Usage

```
rifi_preprocess(
  inp,
  cores,
  FUN_filter = function(x) {     FALSE },
  bg = 0,
  rm_FLT = FALSE,
  thrsh_check = 0,
  dista = 300,
  run_PDD = FALSE,
  pen_PDD = 2,
  pen_out_PDD = 1,
  thrsh_PDD = 0.001,
  pen_TI = 10,
  thrsh_TI = 0.5,
  add = 1000
)
```

Arguments

inp	SummarizedExperiment: the input.
cores	integer: the number of assigned cores for the task.
FUN_filter	function: A function of x, returning a logical. x is the numeric vector of the intensity from all time points for a specific replicate.
bg	numeric: threshold over which the last time point has to be to be fitted with the above background mode.
rm_FLT	logical: remove IDs where all replicates are marked as filtered by the background check. Default is FALSE.
thrsh_check	numeric: the minimal allowed intensity for time point "0". Advised to be kept at 0! Default is 0.
dista	integer: the amount of nucleotides defining the gap. Default is 300.
run_PDD	logical: running the PDD flag function
pen_PDD	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Advised to be kept at 2. Default is 2.
pen_out_PDD	numeric: an internal parameter for the dynamic programming. Higher values result in fewer possible outliers. Advised to be kept at 1. Default is 1.
thrsh_PDD	numeric: an internal parameter that allows fragments with slopes steeper than the threshold to be flagged with "*PDD*". Higher values result in fewer candidates . Advised to be kept at 0.001. Default is 0.001.
pen_TI	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Advised to be kept at 10. Default is 10.
thrsh_TI	numeric: an internal parameter that allows fragments with a certain amount of IDs with higher relative intensities at time points later than "0" to be flagged as "*TI*". Higher values result in fewer candidates. -0.5 is 25 %, 0 is 50%, 0.5 is 75%. Advised to be kept at 0.5. Default is 0.5.
add	integer: range of nucleotides before a potential TI event where in IDs are fitted with the TI fit.

Details

rifi_preprocess allows for the optional integration of filter functions. Filter functions mark replicates with TRUE. Those are then not considered in the fit! FUN_filter is a general filter usually to exclude probes with low expression or "bad" patterns.

Value

The SummarizedExperiment object: checked, and with position, ID, intensity, probe_TI, position_segment, flag and filtration added to the rowRanges.

See Also

check_input

make_df

segment_pos

finding_PDD

finding_TI

Examples

```
data(example_input_minimal)
rifi_preprocess(
  inp = example_input_minimal, cores = 2, bg = 100, rm_FLT = FALSE,
  thrsh_check = 0, dista = 300, run_PDD = FALSE
  )
```

`rifi_stats` *rifi_stats wraps all statistical prediction steps conveniently*

Description

rifi_stats wraps the functions:

1. predict_ps_itss
2. apply_Ttest_delay
3. apply_ancova
4. apply_event_position
5. apply_t_test
6. fold_change
7. apply_manova
8. apply_t_test_ti
9. gff3_preprocess

Usage

```
rifi_stats(inp, dista = 300, path)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
dista	integer: the maximal distance allowed between two successive fragments. Default is the auto generated value.
path	path: to the directory containing the gff3 file.

Value

The SummarizedExperiment object: ID with position, strand, intensity, probe_TI, flag, position_segment, delay, half_life, TI_termination_factor, delay_fragment, velocity_fragment, intercept, slope, HL_fragment, HL_mean_fragment, intensity_fragment, intensity_mean_fragment, TU, TI_termination_fragment, TI_mean_termination_factor, seg_ID, pausing_site, iTSS_I, ps_ts_fragment, event_ps_itss_p_value_Ttest, p_value_slope, delay_frg_slope, velocity_ratio, event_duration, event_position, FC_HL, FC_fragment_HL, p_value_HL, FC_intensity, FC_fragment_intensity, p_value_intensity, FC_HL_intensity, FC_HL_intensity_fragment, FC_HL_adapted, synthesis_ratio, synthesis_ratio_event, p_value_Manova, p_value_TI, TI_fragments_p_value

See Also

`predict_ps_itss`

`apply_Ttest_delay`

`apply_ancova`

`apply_event_position`

`apply_t_test`

`fold_change`

`apply_manova`

`apply_t_test_ti`

`gff3_preprocess`

Examples

```
data(fragmentation_minimal)
rifi_stats(inp = fragmentation_minimal, dista = 300,
path = gzfile(system.file("extdata", "gff_e_coli.gff3.gz",
package = "rifi")))
```

`rifi_summary` %

==========

rifi_summary

rifi_summary wraps conveniently and summarize all rifi outputs

---

Description

rifi_summary wraps the functions:

1. event_dataframe
2. dataframe_summary
3. dataframe_summary_events

4. dataframe_summary_events_HL_int
5. dataframe_summary_events_ps_itss
6. dataframe_summary_events_velocity
7. dataframe_summary_TI

Usage

```
rifi_summary(inp, data_annotation = metadata(inp)$annot[[1]])
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
data_annotation	dataframe: gff3 dataframe after processing.

Value

WIP

See Also

event_dataframe

dataframe_summary

dataframe_summary_events

dataframe_summary_events_HL_int

dataframe_summary_events_ps_itss

dataframe_summary_events_velocity

dataframe_summary_TI

Examples

```
data(stats_minimal)
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
rifi_summary(inp = stats_minimal, data_annotation =
metadata(stats_minimal)$annot[[1]])
```

rifi_visualization %

==========

rifi_visualization

rifi_visualization plots all the data with fragments and events from both strands

Description

rifi_visualization plots the whole genome with genes, transcription units (TUs), delay, half-life (HL), intensity fragments, features, events, velocity, annotation, coverage if available.

Usage

```
rifi_visualization(
  data,
  genomeLength,
  annot,
  coverage = 0,
  chr_fwd = NA,
  chr_rev = NA,
  region = c("CDS", "asRNA", "5'UTR", "ncRNA", "3'UTR", "tRNA"),
  color_region = c("grey0", "red", "blue", "orange", "yellow", "green", "white",
    "darkseagreen1", "grey50", "black"),
  color_text.1 = "grey0",
  color_text.2 = "black",
  color_TU = "blue",
  Alpha = 0.5,
  size_tu = 1.6,
  size_locusTag = 1.6,
  size_gene = 1.6,
  Limit = 10,
  shape = 22,
  col_outiler = "grey50",
  col_coverage = "grey",
  shape_outlier = 13,
  limit_intensity = NA,
  face = "bold",
  tick_length = 0.3,
  arrow.color = "darkseagreen1",
  minVelocity = 3000,
  medianVelocity = 6000,
  col_above20 = "#00FFFF",
  fontface = "plain",
  shape_above20 = 14,
```

```
  col_outlierabove10 = "darkorchid",
  shape_outlierabove10 = 5,
  axis_text_y_size = 3,
  axis_title_y_size = 6,
  TI_threshold = 1.1,
  termination_threshold = -0.5,
  iTSS_threshold = 0.5,
  p_value_int = 0.05,
  p_value_event = 0.05,
  p_value_hl = 0.05,
  p_value_TI = 0.05,
  p_value_manova = 0.05,
  event_duration_ps = 1,
  event_duration_itss = -1,
  HL_threshold_1 = log2(1.5),
  HL_threshold_2 = -log2(1.5),
  vel_threshold = 200,
  HL_threshold_color = "black",
  vel_threshold_color = "grey52",
  ps_color = "orange",
  iTSS_I_color = "blue"
)
```

Arguments

data	SummarizedExperiment: the input data frame with correct format.
genomeLength	integer: genome length output of gff3_preprocess function and element of metadata of SummarizedExperiment.
annot	dataframe: the annotation file, output of gff3_preprocess function and element of metadata of SummarizedExperiment.
coverage	integer: in case the coverage is available.
chr_fwd	string object: coverage of the forward strand.
chr_rev	string object: coverage of the reverse strand.
region	dataframe: gff3 features of the genome.
color_region	string vector: vector of colors.
color_text.1	string: TU color text
color_text.2	string: genes color text
color_TU	string. TU color
Alpha	integer: color transparency degree.
size_tu	integer: TU size
size_locusTag	integer: locus_tag size
size_gene	integer: font size for gene annotation.
Limit	integer: value for y-axis limit.
shape	integer: value for shape.

col_outiler string: outlier color.

col_coverage integer: color for coverage plot.

shape_outlier integer: value for outlier shape.

limit_intensity integer: intensity limit if applicable.

face string: label font.

tick_length integer: value for ticks.

arrow.color string: arrows color.

minVelocity integer: threshold to fix the minimum of velocity.

medianVelocity integer: threshold to fix the maximum of velocity.

col_above20 string: color for probes/bin above value 20.

fontface integer: font type

shape_above20 integer: shape for probes/bins above value 20.

col_outlierabove10 string: color for probes/bin outliers between 10 and 20,

shape_outlierabove10 integer: shape for probes/bin outliers between 10 and 20,

axis_text_y_size integer: text size for y-axis.

axis_title_y_size integer: title size for y-axis.

TI_threshold integer: threshold for TI between two fragments in case the TI termination factor drops from the first segment to the second, default 1.1. If threshold is reached a line is drawn to seperates the two TI segments.

termination_threshold integer: threshold for termination to plot, default .8.

iTSS_threshold integer: threshold for iTSS_II selected to plot, default 1.2.

p_value_int integer: p_value of intensity fragments fold-change to plot, default 0.05.

p_value_event integer: p_value of t-test from pausing site and iTSS_I events to plot, default 0.05.

p_value_hl integer: p_value of half_life fragments fold-change to plot, default 0.05.

p_value_TI integer: p_value of TI fragments selected to be plotted, default 0.05.

p_value_manova integer: p_value of manova test fragments to plot, default 0.05.

event_duration_ps integer: threshold for pausing sites selected to plot, default -2.

event_duration_itss integer: threshold for iTSS_I selected to plot, default 2.

HL_threshold_1 integer: threshold for log2FC(HL) selected to plot, default log2(1.5). log2FC(HL) >= log2(1.5) are indicated by black color. If p_value <= p_value_hl (default 0.05), log2FC(HL) is indicated by HL* otherwise HL.

`HL_threshold_2` integer: threshold for log2FC(HL) selected to plot, default -log2(1.5). log2FC(HL) <= -log2(1.5) are indicated by green color. If p_value <= p_value_hl (default 0.05), log2FC(HL) is indicated by HL* otherwise HL. In case of p_value is significant and the log2FC(HL) is between -log2FC(1.5) and log2FC(1.5), FC is assigned by green color and HL*.

`vel_threshold` integer: threshold for velocity ratio selected to plot, default 200.

`HL_threshold_color` string: color for HL fold change plot.

`vel_threshold_color` string: color for velocity ratio plot.

`ps_color` string: color for pausing site plot.

`iTSS_I_color` string: color for iTSS_I plot.

Details

rifi_visualization uses several functions to plot the genes including as-RNA and ncRNA and TUs as segments. The function plots delay, HL and intensity fragments with statistical t-test between the neighboring fragment, significant t-test is assigned with *''*. *t-test and Manova statistical test are also depicted as ''*.

The functions used are:

1. annotation_plot: plots the corresponding annotation.
2. positive_strand_function: plots delay, HL, intensity and events of positive strand.
3. negative_strand_function: plots delay, HL, intensity and events of negative strand.
4. empty_data_positive: plots empty boxes in case no data is available for positive strand.
5. empty_data_negative: plots empty boxes in case no data is available for negative strand.
6. strand_selection: check if data is stranded and arrange by position.
7. splitGenome_function: splits the genome into fragments.
8. indice_function: assign a new column to the data to distinguish between fragments, outliers from delay or HL or intensity.
9. TU_annotation: designs the segments border for the genes and TUs annotation
10. gene_annot_function: it requires gff3 file, returns a dataframe adjusting each fragment according to its annotation. It allows as well the plot of genes and TUs shared into two pages.
11. label_log2_function: used to add log scale to intensity values.
12. label_square_function: used to add square scale to coverage values.
13. coverage_function: this function is used only in case of coverage is available.
14. secondaryAxis: adjusts the half-life or delay to 20 in case of the dataframe row numbers is equal to 1 and the half-life or delay exceed the limit, they are plotted with different shape and color.
15. outlier_plot: plot the outliers with half-life between 10 and 30 on the maximum of the yaxis.
16. add_genomeBorders: when the annotated genes are on the borders, they can not be plotted, therefore the region was split in 2 adding the row corresponding to the split part to the next annotation (i + 1) except for the first page.

17. my_arrow: creates an arrow for the annotation.
18. arrange_byGroup: selects the last row for each segment and add 40 nucleotides in case of negative strand for a nice plot.
19. regr: plots the predicted delay from linear regression if the data is on negative strand.
20. meanPosition: assign a mean position for the plot.
21. delay_mean: adds a column in case of velocity is NA or equal to 60. The mean of the delay is calculated outliers.
22. my_segment_T: plots terminals and pausing sites labels.
23. my_segment_NS: plots internal starting sites 'iTSS'.
24. min_value: returns minimum value for event plots in intensity plot.
25. velocity_fun: function for velocity plot.
26. limit_function: for values above 10 or 20 in delay and hl. Limit of the axis is set differently. y-axis limit is applied only if we have more than 3 values above 10 and lower or equal to 20. An exception is added in case a dataframe has less than 3 rows and 1 or more values are above 10, the rest of the values above 20 are adjusted to 20 on "secondaryAxis" function.
27. empty_boxes: used only in case the dataframe from the positive strand is not empty, the TU are annotated.
28. function_TU_arrow: used to avoid plotting arrows when a TU is split into two pages.
29. terminal_plot_lm: draws a linear regression line when terminal outliers have an intensity above a certain threshold and are consecutive. Usually are smallRNA (ncRNA, asRNA).
30. slope_function: replaces slope lower than 0.0009 to 0.
31. velo_function: replaces infinite velocity with NA.
32. plot the coverage of RNA_seq in exponential phase growth

Value

The visualization plot.

Examples

```
data(stats_minimal)
if(!require(SummarizedExperiment)){
suppressPackageStartupMessages(library(SummarizedExperiment))
}
rifi_visualization(data = stats_minimal,
genomeLength = metadata(stats_minimal)$annot[[2]],
annot = metadata(stats_minimal)$annot[[1]])
```

rifi_wrapper %

==========

rifi_wrapper

rifi_wrapper wraps conveniently all functions included on rifi workflow

Description

rifi_wrapper wraps the functions:

1. rifi_preprocess
2. rifi_fit
3. rifi_penalties
4. rifi_fragmentation
5. rifi_stats
6. rifi_summary
7. rifi_visualization.

Usage

```
rifi_wrapper(inp, cores, path, bg, restr)
```

Arguments

inp	data frame: the input data frame with correct format.
cores	integer: the number of assigned cores for the task.
path	path: path to an annotation file in gff format.
bg	numeric: threshold over which the last time point has to be to be fitted with the above background mode.
restr	numeric: a parameter that restricts the freedom of the fit to avoid wrong TI-term_factors, ranges from 0 to 0.2

Value

All intermediate objects

See Also

rifi_preprocess

rifi_fit

rifi_penalties

rifi_fragmentation

rifi_stats

rifi_summary

rifi_visualization

Examples

```
data(example_input_minimal)
rifi_wrapper(inp = example_input_minimal, cores = 2, path =
gzfile(system.file("extdata", "gff_e_coli.gff3.gz", package = "rifi")),
bg = 0, restr = 0.01)
```

---

segment_pos *%*

==========

segment_pos

segment_pos divides all IDs by position into position_segments

---

Description

segment_pos adds the column "position_segment" to the rowRanges. To reduce run time, the data is divided by regions of no expression larger than "dist" nucleotides.

Usage

```
segment_pos(inp, dista = 300)
```

Arguments

inp	SummarizedExperiment: the input.
dista	integer: the amount of nucleotides defining the gap. Default is 300.

Value

The SummarizedExperiment object:

`ID:`	The bin/probe specific ID
`position:`	The bin/probe specific position
`intensity:`	The relative intensity at time point 0
`probe_TI:`	An internal value to determine which fitting model is applied
`flag:`	Information on which fitting model is applied
`position_segment:`	The position based segment

Examples

```
data(preprocess_minimal)
segment_pos(inp = preprocess_minimal, dista = 300)
```

`stats_e_coli` *The result of rifi_stats for E.coli example data A SummarizedExperiment containing the output from rifi_stats*

Description

The result of rifi_stats for E.coli example data A SummarizedExperiment containing the output from rifi_stats

Usage

```
data(stats_e_coli)
```

Format

A SummarizedExperiment:

ID: The bin/probe specific ID

position: The bin/probe specific position

strand: The bin/probe specific strand

intensity: The relative intensity at time point 0

probe_TI: An internal value to determine which fitting model is applied

flag: Information on which fitting model is applied

position_segment: The position based segment

delay: The delay value of the bin/probe

half_life: The half-life of the bin/probe

TI_termination_factor: String, the factor of TI fragment

delay_fragment: The delay fragment the bin belongs to
velocity_fragment: The velocity value of the respective delay fragment
intercept: The vintercept of fit through the respective delay fragment
slope: The slope of the fit through the respective delay fragment
HL_fragment: The half-life fragment the bin belongs to
HL_mean_fragment: The mean half-life value of the respective half-life fragment
intensity_fragment: The intensity fragment the bin belongs to
intensity_mean_fragment: The mean intensity value of the respective intensity fragment
TU: The overarching transcription unit
TI_termination_fragment: The TI fragment the bin belongs to
TI_mean_termination_factor: The mean termination factor of the respective TI fragment
seg_ID: The combined ID of the fragment
pausing_site: presence of pausing site indicated by +/-
iTSS_I: presence of iTSS_I indicated by +/-
ps_ts_fragment: The fragments involved in pausing site or iTSS_I
event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I
p_value_slope: p_value of the slope
delay_frg_slope: the slope value of the respective delay fragment
velocity_ratio: Integer, ratio of velocity between 2 delay fragments
event_duration: Integer, the duration between two delay fragments
event_position: Integer, the position middle between 2 fragments with an event
FC_HL: Integer, the fold change value of 2 HL fragments
FC_fragment_HL: Integer, the fold change value of 2 intensity fragments
p_value_HL: p_value of the fold change of HL fragments
FC_intensity: Integer, the fold change value of 2 intensity fragments
FC_fragment_intensity: String, fragments involved in fold change between 2 intensity fragments
p_value_intensity: p_value of the fold change of intensity fragments
FC_HL_intensity: ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments
FC_HL_intensity_fragment: fragments involved on ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments
FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment
synthesis_ratio: Integer, the value correspomding to synthesis rate
synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS
p_value_Manova: p_value of the variance between two fold-changes, HL and intensity
p_value_TI: p_value of TI fragment
TI_fragments_p_value: p_value of 2 TI fragments

Source

https://github.com/CyanolabFreiburg/rifi

`stats_minimal` *The result of rifi_stats for artificial example data A SummarizedExperiment containing the output of rifi_stats as an extention to rowRanges and metadata (gff file processed, see gff file documentation)*

Description

The result of rifi_stats for artificial example data A SummarizedExperiment containing the output of rifi_stats as an extention to rowRanges and metadata (gff file processed, see gff file documentation)

Usage

```
data(stats_minimal)
```

Format

A rowRanges of SummarizedExperiment with 24 rows and 45 variables:

ID: The bin/probe specific ID

position: The bin/probe specific position

intensity: The relative intensity at time point 0

probe_TI: An internal value to determine which fitting model is applied

flag: Information on which fitting model is applied

position_segment: The position based segment

delay: The delay value of the bin/probe

half_life: The half-life of the bin/probe

TI_termination_factor: String, the factor of TI fragment

delay_fragment: The delay fragment the bin belongs to

velocity_fragment: The velocity value of the respective delay fragment

intercept: The vintercept of fit through the respective delay fragment

slope: The slope of the fit through the respective delay fragment

HL_fragment: The half-life fragment the bin belongs to

HL_mean_fragment: The mean half-life value of the respective half-life fragment

intensity_fragment: The intensity fragment the bin belongs to

intensity_mean_fragment: The mean intensity value of the respective intensity fragment

TU: The overarching transcription unit

TI_termination_fragment: The TI fragment the bin belongs to

TI_mean_termination_factor: The mean termination factor of the respective TI fragment

seg_ID: The combined ID of the fragment

pausing_site: presence of pausing site indicated by +/-

iTSS_I: presence of iTSS_I indicated by +/-

ps_ts_fragment: The fragments involved in pausing site or iTSS_I

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I

p_value_slope: p_value of the slope

delay_frg_slope: the slope value of the respective delay fragment

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

event_duration: Integer, the duration between two delay fragments

event_position: Integer, the position middle between 2 fragments with an event

FC_HL: Integer, the fold change value of 2 HL fragments

FC_fragment_HL: Integer, the fold change value of 2 intensity fragments

p_value_HL: p_value of the fold change of HL fragments

FC_intensity: Integer, the fold change value of 2 intensity fragments

FC_fragment_intensity: String, fragments involved in fold change between 2 intensity fragments

p_value_intensity: p_value of the fold change of intensity fragments

FC_HL_intensity: ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments

FC_HL_intensity_fragment: fragments involved on ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

synthesis_ratio: Integer, the value correspomding to synthesis rate

synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS

p_value_Manova: p_value of the variance between two fold-changes, HL and intensity

p_value_TI: p_value of TI fragment

TI_fragments_p_value: p_value of 2 TI fragments

Source

https://github.com/CyanolabFreiburg/rifi

`stats_synechocystis_6803`

The result of rifi_stats for Synechocystis 6803 example data A SummarizedExperiment containing the output of rifi_stats as an extention to rowRanges

---

Description

The result of rifi_stats for Synechocystis 6803 example data A SummarizedExperiment containing the output of rifi_stats as an extention to rowRanges

Usage

```
data(stats_synechocystis_6803)
```

Format

The rowRanges of SummarizedExperiment:

ID: The bin/probe specific ID

position: The bin/probe specific position

intensity: The relative intensity at time point 0

probe_TI: An internal value to determine which fitting model is applied

flag: Information on which fitting model is applied

position_segment: The position based segment

delay: The delay value of the bin/probe

half_life: The half-life of the bin/probe

TI_termination_factor: String, the factor of TI fragment

delay_fragment: The delay fragment the bin belongs to

velocity_fragment: The velocity value of the respective delay fragment

intercept: The vintercept of fit through the respective delay fragment

slope: The slope of the fit through the respective delay fragment

HL_fragment: The half-life fragment the bin belongs to

HL_mean_fragment: The mean half-life value of the respective half-life fragment

intensity_fragment: The intensity fragment the bin belongs to

intensity_mean_fragment: The mean intensity value of the respective intensity fragment

TU: The overarching transcription unit

TI_termination_fragment: The TI fragment the bin belongs to

TI_mean_termination_factor: The mean termination factor of the respective TI fragment

seg_ID: The combined ID of the fragment

pausing_site: presence of pausing site indicated by +/-

iTSS_I: presence of iTSS_I indicated by +/-

ps_ts_fragment: The fragments involved in pausing site or iTSS_I

event_ps_itss_p_value_Ttest: p_value of pausing site or iTSS_I

p_value_slope: p_value of the slope

delay_frg_slope: the slope value of the respective delay fragment

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

event_duration: Integer, the duration between two delay fragments

event_position: Integer, the position middle between 2 fragments with an event

FC_HL: Integer, the fold change value of 2 HL fragments

FC_fragment_HL: Integer, the fold change value of 2 intensity fragments

p_value_HL: p_value of the fold change of HL fragments

FC_intensity: Integer, the fold change value of 2 intensity fragments

FC_fragment_intensity: String, fragments involved in fold change between 2 intensity fragments

p_value_intensity: p_value of the fold change of intensity fragments

FC_HL_intensity: ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments

FC_HL_intensity_fragment: fragments involved on ratio of fold change between 2 half-life fragments and fold change between 2 intensity fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

synthesis_ratio: Integer, the value correspomding to synthesis rate

synthesis_ratio_event: String, the event assigned by synthesis rate either Termination or iTSS

p_value_Manova: p_value of the variance between two fold-changes, HL and intensity

p_value_TI: p_value of TI fragment

TI_fragments_p_value: p_value of 2 TI fragments

Source

https://github.com/CyanolabFreiburg/rifi

`summary_e_coli` *The result of rifi_summary for E.coli example data A SummarizedExperiment containing the output of rifi_stats as an extention to rowRanges*

Description

The result of rifi_summary for E.coli example data A SummarizedExperiment containing the output of rifi_stats as an extention to rowRanges

Usage

```
data(summary_e_coli)
```

Format

The rowRanges of SummarizedExperiment:

bin_df: all information regarding bins:

ID: The bin/probe specific ID

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

position: The bin/probe specific position
strand: The bin/probe specific strand
segment: The segment the bin/probe belongs to
TU: The overarching transcription unit
delay_fragment: The delay fragment the bin/probe belongs to
delay: The delay of the bin/probe
HL_fragment: The half-life fragment the bin/probe belongs to
half_life: The half-life of the bin/probe
intensity_fragment: The intensity fragment the bin/probe belongs to
intensity: The relative intensity at time point 0
flag: The flag of the bin/probe(TI, PDD)
TI_termination_factor: String, the factor of TI fragment

frag_df: all information regarding fragments:

feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
first_position_frg: The first position of the fragment on the genome
last_position_frg: The last position of the fragment on the genome
strand: The bin/probe specific strand
TU: The overarching transcription unit
segment: The segment the fragment belongs to
delay_fragment: The delay fragment of the fragment
HL_fragment: The half-life fragment of the fragment
half_life: The half-life mean of the fragment
HL_SD: The half-life standard deviation of the fragment
HL_SE: The half-life standard error of the fragment
intensity_fragment: The intensity_fragment of the fragment
intensity: The relative intensity at time point 0
intensity_SD: The intensity standard deviation of the fragment
intensity_SE: The intensity standard error of the fragment
velocity: The velocity value of the respective delay fragment

event_df: all information regarding events:

event: String, event type
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
FC_HL: Integer, the fold change value of 2 HL fragments
FC_intensity: Fold change of intensity
FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment
FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity
event_position: Integer, the position middle between 2 fragments with an event
velocity_ratio: Integer, ratio of velocity between 2 delay fragments

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

segment_1: String, the first fragment of the two of fragments subjected to analysis

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

events_HL_int_df: all information regarding events related to half-life and intensity:

event: String, event type

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

FC_HL: Integer, the fold change value of 2 HL fragments

FC_intensity: Integer, the fold change value of 2 intensity fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity

event_position: Integer, the position middle between 2 fragments with an event

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

segment_1: String, the first fragment of the two of fragments subjected to analysis

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

events_ps_itss_df: all information regarding events related to pausing sites and iTSS_I:

event: String, event type

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

event_position: Integer, the position middle between 2 fragments with an event

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit
segment_1: String, the first fragment of the two of fragments subjected to analysis
segment_2: String, the second fragment of the two of fragments subjected to analysis
event_duration: Integer, the duration between two delay fragments
gap_fragments: Integer, the distance between two delay fragments
features: Integer, number of fragements involved on the event

events_velocity_df: all information regarding events related to velocity:

event: String, event type
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
event_position: Integer, the position middle between 2 fragments with an event
velocity_ratio: Integer, ratio of velocity between 2 delay fragments
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: The bin/probe specific strand
TU: The overarching transcription unit
segment_1: String, the first fragment of the two of fragments subjected to analysis
segment_2: String, the second fragment of the two of fragments subjected to analysis
event_duration: Integer, the duration between two delay fragments
gap_fragments: Integer, the distance between two delay fragments
features: Integer, number of fragements involved on the event

TI_df: all information regarding TI:

event: String, event type
TI_fragment: String, the fragment with TI
TI_termination_factor: String, the factor of TI fragment
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: The bin/probe specific strand
TU: The overarching transcription unit
features: Integer, number of fragements involved on the event
event_position: Integer, the position middle between 2 fragments with an event
position_1: the first position of TI fragment, if 2 fragments, first position is from the first fragment
position_2: the last position of TI fragment, if 2 fragments, last position is from the second fragment.

Source

https://github.com/CyanolabFreiburg/rifi

summary_minimal *The result of rifi_summary for artificial example data A SummarizedExperiment with the output from rifi_summary as metadata*

Description

The result of rifi_summary for artificial example data A SummarizedExperiment with the output from rifi_summary as metadata

Usage

```
data(summary_minimal)
```

Format

A list of 7 data frames with 290 rows and 11 variables, 36 rows and 11 variables, 57 rows and 18 variables, and 8 rows and 14 variables:

bin_df: all information regarding bins:

ID: The bin/probe specific ID
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
position: The bin/probe specific position
strand: The bin/probe specific strand
segment: The segment the bin/probe belongs to
TU: The overarching transcription unit
delay_fragment: The delay fragment the bin/probe belongs to
delay: The delay of the bin/probe
HL_fragment: The half-life fragment the bin/probe belongs to
half_life: The half-life of the bin/probe
intensity_fragment: The intensity fragment the bin/probe belongs to
intensity: The relative intensity at time point 0
flag: The flag of the bin/probe(TI, PDD)
TI_termination_factor: String, the factor of TI fragment

frag_df: all information regarding fragments:

feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
first_position_frg: The first position of the fragment on the genome
last_position_frg: The last position of the fragment on the genome
strand: The bin/probe specific strand
TU: The overarching transcription unit

segment: The segment the fragment belongs to
delay_fragment: The delay fragment of the fragment
HL_fragment: The half-life fragment of the fragment
half_life: The half-life mean of the fragment
HL_SD: The half-life standard deviation of the fragment
HL_SE: The half-life standard error of the fragment
intensity_fragment: The intensity_fragment of the fragment
intensity: The relative intensity at time point 0
intensity_SD: The intensity standard deviation of the fragment
intensity_SE: The intensity standard error of the fragment
velocity: The velocity value of the respective delay fragment

event_df: all information regarding events:

event: String, event type
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
FC_HL: Integer, the fold change value of 2 HL fragments
FC_intensity: Fold change of intensity
FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment
FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity
event_position: Integer, the position middle between 2 fragments with an event
velocity_ratio: Integer, ratio of velocity between 2 delay fragments
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: The bin/probe specific strand
TU: The overarching transcription unit
segment_1: String, the first fragment of the two of fragments subjected to analysis
segment_2: String, the second fragment of the two of fragments subjected to analysis
event_duration: Integer, the duration between two delay fragments
gap_fragments: Integer, the distance between two delay fragments
features: Integer, number of fragements involved on the event

events_HL_int_df: all information regarding events related to half-life and intensity:

event: String, event type
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
FC_HL: Integer, the fold change value of 2 HL fragments
FC_intensity: Integer, the fold change value of 2 intensity fragments
FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment
FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity
event_position: Integer, the position middle between 2 fragments with an event

feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: The bin/probe specific strand
TU: The overarching transcription unit
segment_1: String, the first fragment of the two of fragments subjected to analysis
segment_2: String, the second fragment of the two of fragments subjected to analysis
event_duration: Integer, the duration between two delay fragments
gap_fragments: Integer, the distance between two delay fragments
features: Integer, number of fragements involved on the event

events_ps_itss_df: all information regarding events related to pausing sites and iTSS_I:

event: String, event type
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
event_position: Integer, the position middle between 2 fragments with an event
velocity_ratio: Integer, ratio of velocity between 2 delay fragments
FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: The bin/probe specific strand
TU: The overarching transcription unit
segment_1: String, the first fragment of the two of fragments subjected to analysis
segment_2: String, the second fragment of the two of fragments subjected to analysis
event_duration: Integer, the duration between two delay fragments
gap_fragments: Integer, the distance between two delay fragments
features: Integer, number of fragements involved on the event

events_velocity_df: all information regarding events related to velocity:

event: String, event type
p_value: Integer, p_value of the event
p_adjusted: Integer, p_value adjusted
event_position: Integer, the position middle between 2 fragments with an event
velocity_ratio: Integer, ratio of velocity between 2 delay fragments
feature_type: String, region annotation covering the fragments
gene: String, gene annotation covering the fragments
locus_tag: String, locus_tag annotation covering the fragments
strand: The bin/probe specific strand
TU: The overarching transcription unit
segment_1: String, the first fragment of the two of fragments subjected to analysis
segment_2: String, the second fragment of the two of fragments subjected to analysis
event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

TI_df: all information regarding TI:

event: String, event type

TI_fragment: String, the fragment with TI

TI_termination_factor: String, the factor of TI fragment

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

features: Integer, number of fragements involved on the event

event_position: Integer, the position middle between 2 fragments with an event

position_1: the first position of TI fragment, if 2 fragments, first position is from the first fragment

position_2: the last position of TI fragment, if 2 fragments, last position is from the second fragment.

Source

https://github.com/CyanolabFreiburg/rifi

`summary_synechocystis_6803`

The result of rifi_summary for Synechocystis 6803 example data A list containing the output from rifi_summary, including the fragment based data frame, bin based data frame, event data frame and the TI dataframe.

Description

The result of rifi_summary for Synechocystis 6803 example data A list containing the output from rifi_summary, including the fragment based data frame, bin based data frame, event data frame and the TI dataframe.

Usage

```
data(summary_synechocystis_6803)
```

Format

A list of 4 data frames with 3000 rows and 11 variables, 297 rows and 11 variables, 486 rows and 18 variables, and 10 rows and 14 variables:

bin_df: all information regarding bins:

- **ID:** The bin/probe specific ID
- **feature_type:** String, region annotation covering the fragments
- **gene:** String, gene annotation covering the fragments
- **locus_tag:** String, locus_tag annotation covering the fragments
- **position:** The bin/probe specific position
- **strand:** The bin/probe specific strand
- **segment:** The segment the bin/probe belongs to
- **TU:** The overarching transcription unit
- **delay_fragment:** The delay fragment the bin/probe belongs to
- **delay:** The delay of the bin/probe
- **HL_fragment:** The half-life fragment the bin/probe belongs to
- **half_life:** The half-life of the bin/probe
- **intensity_fragment:** The intensity fragment the bin/probe belongs to
- **intensity:** The relative intensity at time point 0
- **flag:** The flag of the bin/probe(TI, PDD)
- **TI_termination_factor:** String, the factor of TI fragment

frag_df: all information regarding fragments:

- **feature_type:** String, region annotation covering the fragments
- **gene:** String, gene annotation covering the fragments
- **locus_tag:** String, locus_tag annotation covering the fragments
- **first_position_frg:** The first position of the fragment on the genome
- **last_position_frg:** The last position of the fragment on the genome
- **strand:** The bin/probe specific strand
- **TU:** The overarching transcription unit
- **segment:** The segment the fragment belongs to
- **delay_fragment:** The delay fragment of the fragment
- **HL_fragment:** The half-life fragment of the fragment
- **half_life:** The half-life mean of the fragment
- **HL_SD:** The half-life standard deviation of the fragment
- **HL_SE:** The half-life standard error of the fragment
- **intensity_fragment:** The intensity_fragment of the fragment
- **intensity:** The relative intensity at time point 0
- **intensity_SD:** The intensity standard deviation of the fragment
- **intensity_SE:** The intensity standard error of the fragment
- **velocity:** The velocity value of the respective delay fragment

event_df: all information regarding events:

- **event:** String, event type

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

FC_HL: Integer, the fold change value of 2 HL fragments

FC_intensity: Fold change of intensity

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity

event_position: Integer, the position middle between 2 fragments with an event

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

segment_1: String, the first fragment of the two of fragments subjected to analysis

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

events_HL_int_df: all information regarding events related to half-life and intensity:

event: String, event type

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

FC_HL: Integer, the fold change value of 2 HL fragments

FC_intensity: Integer, the fold change value of 2 intensity fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

FC_HL_FC_intensity: Fold change of half-life/ fold change of intensity

event_position: Integer, the position middle between 2 fragments with an event

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

segment_1: String, the first fragment of the two of fragments subjected to analysis

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

events_ps_itss_df: all information regarding events related to pausing sites and iTSS_I:

event: String, event type

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

event_position: Integer, the position middle between 2 fragments with an event

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

FC_HL_adapted: Integer, the fold change of half-life/ fold change of intensity, position of the half-life fragment is adapted to intensity fragment

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

segment_1: String, the first fragment of the two of fragments subjected to analysis

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

events_velocity_df: all information regarding events related to velocity:

event: String, event type

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

event_position: Integer, the position middle between 2 fragments with an event

velocity_ratio: Integer, ratio of velocity between 2 delay fragments

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

segment_1: String, the first fragment of the two of fragments subjected to analysis

segment_2: String, the second fragment of the two of fragments subjected to analysis

event_duration: Integer, the duration between two delay fragments

gap_fragments: Integer, the distance between two delay fragments

features: Integer, number of fragements involved on the event

TI_df: all information regarding TI:

event: String, event type

TI_fragment: String, the fragment with TI

TI_termination_factor: String, the factor of TI fragment

p_value: Integer, p_value of the event

p_adjusted: Integer, p_value adjusted

feature_type: String, region annotation covering the fragments

gene: String, gene annotation covering the fragments

locus_tag: String, locus_tag annotation covering the fragments

strand: The bin/probe specific strand

TU: The overarching transcription unit

features: Integer, number of fragements involved on the event

event_position: Integer, the position middle between 2 fragments with an event

position_1: the first position of TI fragment, if 2 fragments, first position is from the first fragment

position_2: the last position of TI fragment, if 2 fragments, last position is from the second fragment.

Source

https://github.com/CyanolabFreiburg/rifi

TI_fit %

==========

TI_fit

TI_fit estimates transcription interference and termination factor using nls function for probe or bin flagged as "TI".

Description

TI_fit uses nls2 function to fit the flagged probes or bins with "TI" found using finding_TI.r. It estimates the transcription interference level (referred later to TI) as well as the transcription factor fitting the probes/bins with nls function looping into several starting values.

Usage

```
TI_fit(
  inp,
  cores = 1,
  restr = 0.2,
  k = seq(0, 1, by = 0.5),
  decay = c(0.05, 0.1, 0.2, 0.5, 0.6),
  ti = seq(0, 1, by = 0.5),
  ti_delay = seq(0, 2, by = 0.5),
  rest_delay = seq(0, 2, by = 0.5),
  bg = 0
)
```

Arguments

inp	SummarizedExperiment: the input with correct format.
cores	integer: the number of assigned cores for the task.
restr	numeric: a parameter that restricts the freedom of the fit to avoid wrong TI-term_factors, ranges from 0 to 0.2.

k	numeric vector: A sequence of starting values for the synthesis rate. Default is seq(0, 1, by = 0.5).
decay	numeric vector: A sequence of starting values for the decay Default is c(0.05, 0.1, 0.2, 0.5, 0.6).
ti	numeric vector: A sequence of starting values for the delay. Default is seq(0, 1, by = 0.5).
ti_delay	numeric vector: A sequence of starting values for the delay. Default is seq(0, 2, by = 0.5).
rest_delay	numeric vector: A sequence of starting values. Default is seq(0, 2, by = 0.5).
bg	numeric vector: A sequence of starting values. Default is 0.

Details

To determine TI and termination factor, TI_fit function is applied to the flagged probes and to the probes localized 1000 nucleotides upstream. Before applying TI_fit function, some probes/bins are filtered out if they are below the background using generic_filter_BG. The model loops into a dataframe containing sequences of starting values and the coefficients are extracted from the fit with the lowest residuals. When many residuals are equal to 0, the lowest residual can not be determined and the coefficients extracted could be wrong. Therefore, a second filter was developed. First we loop into all starting values, we collect nls objects and the corresponding residuals. They are sorted and residuals non equal to 0 are collected in a vector. If the first residuals are not equal to 0, 20 % of the best residuals are collected in tmp_r_min vector and the minimum termination factor is selected. In case the first residuals are equal to 0 then values between 0 to 20% of the values collected in tmp_r_min vector are gathered. The minimum termination factor coefficient is determined and saved. The coefficients are gathered in res vector and saved as an object.

Value

the SummarizedExperiment object: with delay, decay and TI_termination_factor added to the rowRanges. The full fit data is saved in the metadata as "fit_TI".

Examples

```
data(preprocess_minimal)
TI_fit(inp = preprocess_minimal, cores=2, restr=0.01)
```

TUgether %

==========

TUgether

TUgether combines delay fragments into TUs

Description

TUgether combines delay fragments into TUs. The column "TU" is added. It uses score fun_increasing on the start and end points of delay_fragments.

Usage

```
TUgether(inp, cores = 1, pen = -0.75)
```

Arguments

inp	SummarizedExperiment: the input data frame with correct format.
cores	cores: integer: the number of assigned cores for the task.
pen	numeric: an internal parameter for the dynamic programming. Higher values result in fewer fragments. Default -0.75.

Details

The function used is: .score_fun_increasing

The input is the SummarizedExperiment object. pen is the penalty for new fragments in the dynamic programming. Since high scores are aimed, pen is negative.

Value

The SummarizedExperiment with the columns regarding the TU:

ID: The bin/probe specific ID.
position: The bin/probe specific position.
intensity: The relative intensity at time point 0.
probe_TI: An internal value to determine which fitting model is applied.
flag: Information on which fitting model is applied.
position_segment: The position based segment.
delay: The delay value of the bin/probe.
half_life: The half-life of the bin/probe.
TI_termination_factor: String, the factor of TI fragment.
delay_fragment: The delay fragment the bin belongs to.
velocity_fragment: The velocity value of the respective delay fragment.
intercept: The vintercept of fit through the respective delay fragment.
slope: The slope of the fit through the respective delay fragment.
HL_fragment: The half-life fragment the bin belongs to.
HL_mean_fragment: The mean half-life value of the respective half-life fragment.
intensity_fragment: The intensity fragment the bin belongs to.
intensity_mean_fragment: The mean intensity value of the respective intensity fragment.
TU: The overarching transcription unit.
TI_termination_fragment: The TI fragment the bin belongs to.
TI_mean_termination_factor: The mean termination factor of the respective TI fragment.
seg_ID: The combined ID of the fragment.

Examples

```
data(fragmentation_minimal)
TUgether(inp = fragmentation_minimal, cores = 2, pen = -0.75)
```

---

viz_pen_obj %

==========

viz_pen_obj

viz_pen_obj visualizes penalty objects

Description

viz_pen_obj provides an optional visualization of any penalty object created by make_pen. the function aan be customized to show only the n = top_i top results.

Usage

```
viz_pen_obj(obj, top_i = nrow(obj[[3]][[1]]) * ncol(obj[[3]][[1]]))
```

Arguments

obj	object: penalty object(make_pen output)
top_i	integer: the number of top results visualized. Default is all.

Value

A visualization of the penalty object

Examples

```
data(penalties_e_coli)
viz_pen_obj(penalties_e_coli$pen_obj_delay,25)
```

wrapper_e_coli — *The result of rifi_wrapper for E.coli example data A list of SummarizedExperiment containing the output of rifi_wrapper. The list contains 6 elements of SummarizedExperiment output of rifi_preprocess, rifi_fit, rifi_penalties, rifi_fragmentation, rifi_stats and rifi_summary. The plot is generated from rifi_visualization. for more detail, please refer to each function separately.*

Description

The result of rifi_wrapper for E.coli example data A list of SummarizedExperiment containing the output of rifi_wrapper. The list contains 6 elements of SummarizedExperiment output of rifi_preprocess, rifi_fit, rifi_penalties, rifi_fragmentation, rifi_stats and rifi_summary. The plot is generated from rifi_visualization. for more detail, please refer to each function separately.

Usage

```
data(wrapper_e_coli)
```

Format

An object of class `list` of length 6.

Source

https://github.com/CyanolabFreiburg/rifi

wrapper_minimal — *The result of rifi_wrapper for E.coli artificial example. A list of SummarizedExperiment containing the output of rifi_wrapper. The list contains 6 elements of SummarizedExperiment output of rifi_preprocess, rifi_fit, rifi_penalties, rifi_fragmentation, rifi_stats and rifi_summary. The plot is generated from rifi_visualization. for more detail, please refer to each function separately.*

Description

The result of rifi_wrapper for E.coli artificial example. A list of SummarizedExperiment containing the output of rifi_wrapper. The list contains 6 elements of SummarizedExperiment output of rifi_preprocess, rifi_fit, rifi_penalties, rifi_fragmentation, rifi_stats and rifi_summary. The plot is generated from rifi_visualization. for more detail, please refer to each function separately.

Usage

```
data(wrapper_minimal)
```

Format

An object of class `list` of length 6.

Source

https://github.com/CyanolabFreiburg/rifi

`wrapper_summary_synechocystis_6803`

The result of rifi_wrapper for summary_synechocystis_6803 example data A list of SummarizedExperiment containing the output of rifi_wrapper. The list contains 6 elements of SummarizedExperiment output of rifi_preprocess, rifi_fit, rifi_penalties, rifi_fragmentation, rifi_stats and rifi_summary. The plot is generated from rifi_visualization. for more detail, please refer to each function separately.

Description

The result of rifi_wrapper for summary_synechocystis_6803 example data A list of SummarizedExperiment containing the output of rifi_wrapper. The list contains 6 elements of SummarizedExperiment output of rifi_preprocess, rifi_fit, rifi_penalties, rifi_fragmentation, rifi_stats and rifi_summary. The plot is generated from rifi_visualization. for more detail, please refer to each function separately.

Usage

```
data(wrapper_summary_synechocystis_6803)
```

Format

An object of class `list` of length 6.

Source

https://github.com/CyanolabFreiburg/rifi

Index

∗ **datasets**
example_input_e_coli, 25
example_input_minimal, 26
example_input_synechocystis_6803, 26
fit_e_coli, 29
fit_minimal, 30
fit_synechocystis_6803, 31
fragmentation_e_coli, 34
fragmentation_minimal, 35
fragmentation_synechocystis_6803, 36
penalties_e_coli, 48
penalties_minimal, 49
penalties_synechocystis_6803, 50
preprocess_e_coli, 53
preprocess_minimal, 54
preprocess_synechocystis_6803, 54
res_minimal, 55
stats_e_coli, 74
stats_minimal, 76
stats_synechocystis_6803, 77
summary_e_coli, 79
summary_minimal, 83
summary_synechocystis_6803, 86
wrapper_e_coli, 94
wrapper_minimal, 94
wrapper_summary_synechocystis_6803, 95

apply_ancova, 3
apply_event_position, 5
apply_manova, 7
apply_t_test, 10
apply_t_test_ti, 12
apply_Ttest_delay, 9

check_input, 14

dataframe_summary, 15
dataframe_summary_events, 16
dataframe_summary_events_HL_int, 18
dataframe_summary_events_ps_itss, 19
dataframe_summary_events_velocity, 21
dataframe_summary_TI, 22

event_dataframe, 23
example_input_e_coli, 25
example_input_minimal, 26
example_input_synechocystis_6803, 26

finding_PDD, 27
finding_TI, 28
fit_e_coli, 29
fit_minimal, 30
fit_synechocystis_6803, 31
fold_change, 32
fragment_delay, 37
fragment_HL, 38
fragment_inty, 40
fragment_TI, 41
fragmentation_e_coli, 34
fragmentation_minimal, 35
fragmentation_synechocystis_6803, 36

gff3_preprocess, 43

make_df, 44
make_pen, 45

nls2_fit, 47

penalties_e_coli, 48
penalties_minimal, 49
penalties_synechocystis_6803, 50
predict_ps_itss, 51
preprocess_e_coli, 53
preprocess_minimal, 54
preprocess_synechocystis_6803, 54

res_minimal, 55

rifi_fit, 57
rifi_fragmentation, 59
rifi_penalties, 60
rifi_preprocess, 62
rifi_stats, 64
rifi_summary, 65
rifi_visualization, 67
rifi_wrapper, 72

segment_pos, 73
stats_e_coli, 74
stats_minimal, 76
stats_synechocystis_6803, 77
summary_e_coli, 79
summary_minimal, 83
summary_synechocystis_6803, 86

TI_fit, 90
TUgether, 91

viz_pen_obj, 93

wrapper_e_coli, 94
wrapper_minimal, 94
wrapper_summary_synechocystis_6803, 95