Package 'rnalab'

March 23, 2020

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Author Emily Dong [aut, cre],
Akshita Batra [aut]

Maintainer Emily Dong <erd5077@stanford.edu>

Depends R (>= 3.5.0)

Description This package is intended for biotechnology companies who manufacture DNA or RNA for use as therapeutic agents (e.g. for gene therapy applications in personalized medicine). This package provides a user-friendly interface (via RShiny) as well as easy-to-use functions to analyze nucleic acid data for biotech users who may not be familiar with interacting with data directly with R.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Imports Biostrings (>= 2.54.0), dplyr, ggplot2 (>= 3.2.1), tidyr (>= 1.0.2), readr (>= 1.3.1), tibble (>= 2.1.3), stringr (>= 1.4.0), shiny (>= 1.4.0), rlist (>= 0.4.6.1), grid, gridExtra, shinyjs, rlang

NeedsCompilation no

R topics documented:

`add_gc_content` *Adding the GC content*

Description

Calculate GC content of the sequence column

Usage

```
add_gc_content(df_sequence, name_seq_col)
```

Arguments

df_sequence	dataframe
name_seq_col	Name of the sequence column

Value

A new dataframe with existing columns and added column named gc_content

Examples

```
df <- data.frame("id" = c('ID1','ID2') , "Sequnce_DNA" = c('AAAGGGCTTCCC','AGGGGGTTTCCC'))
df_new <- add_gc_content(df, 'Sequnce_DNA')
```

`add_mono_nucleotide_length` *Adding longest mono nucleotide length*

Description

Calculate the length of the longest consecutive mono nucleotide

Usage

```
add_mono_nucleotide_length(df_sequence, name_seq_col)
```

Arguments

df_sequence	Dataframe
name_seq_col	Name of the sequence column

Value

A new dataframe with existing columns and four added columns named - long_mono_A, long_mono_C, long_mono_G,long_mono_T

Examples

```
df <- data.frame("id" = c('id1','id2') , "Sequnce_DNA" = c('AAAGGGCTTCCC','AGGGGGTTTCCC'))
df_new <- add_gc_content(df, 'Sequnce_DNA')
```

`add_sequence_length` *Add a column to the existing dataframe*

Description

Calculate length of the sequence column

Usage

```
add_sequence_length(df_sequence, name_seq_col)
```

Arguments

`df_sequence`	global dataframe
`name_seq_col`	Name of the Sequence Column

Value

df_updated

Examples

```
df <- data.frame("id" = c('ID1','ID2') , "Sequnce_DNA" = c('AAAGGGCTTCCC','AGGGGGTTTCCC'))
df_new <- add_sequence_length(df, 'Sequnce_DNA')
```

`available_seq_prop` *Internal function to check which properties users have chosen*

Description

Internal function to check which properties users have chosen

Usage

```
available_seq_prop(df_sequence)
```

Arguments

`df_sequence`	dataframe

Value

df_sequence

`dnaseqs` *Sample DNA Sequence Data*

Description

Sample data set of 500 human cDNA sequences from the Ensembl genome browser. This data set represents an example output of manufacturing data from a biotech company (e.g. if this company were manufacturing these DNA sequences for research or therapeutic purposes). Manufacturing information (e.g. date, yield, purity, impurity1) was generated as toy values for the purposes of demonstrating the utility of the rnalab package.

Usage

```
data(dnaseqs)
```

Format

A data frame with 500 observations of 8 variables.
[,1] ensembl_id
[,2] gene_symbol
[,3] length
[,4] sequence
[,5] date
[,6] yield
[,7] purity
[,8] impurity1

Source

Ensembl H. sapiens cDNA

Examples

```
data(dnaseqs)
rnalab::rnalab_hist_plot(data = dnaseqs, vars = list('length', 'yield'), nbins = 100)
rnalab::rnalab_scatterplot(data = dnaseqs, x = 'length', y = 'yield', fit=TRUE)
```

`filter_seq` *Internal function to filter the data*

Description

#' Internal function to filter the data

Usage

```
filter_seq(inside_data, col, max, min)
```

Arguments

`inside_data`	dataframe
`col`	name of the column
`max`	integer showing the max to filter
`min`	integer showing the min to filter

Value

inside_data

`remove_added_columns` *remove_added_columns*

Description

Internal function to remove any new column added by the App

Usage

```
remove_added_columns(df_sequence)
```

Arguments

`df_sequence`	dataframe

Value

df_sequence

`rename_column` *Rename the columns to the usable names for the internal functions*

Description

Internal function to rename exisiting columns

Usage

```
rename_column(df_sequence, column_length, new_name)
```

Arguments

`df_sequence`	dataframe
`column_length`	Name of the column
`new_name`	New name of the column

Value

df_sequence

rnalab_hist_plot *Plot Histograms for input data*

Description

Use ggplot2 to generate Histograms input data and display the summary statistics next to the plot.

Usage

```
rnalab_hist_plot(data, vars, nbins)
```

Arguments

data	Dataset to use for ggplot2 plots
vars	A list or vector of variables to plot as the x variable in individual histograms
nbins	An integer number of bins to use for all histograms

Value

A grid graphical object consisting of a ggplot2 histogram with printed summary statistics.

Examples

```
# Plot the distribution and summary stats for a single variable in the dnaseqs data set
rnalab_hist_plot(dnaseqs, 'length', 100)
# Plot the distribution and summary stats for multiple variables in the dnaseqs data set
rnalab_hist_plot(dnaseqs, c('length', 'yield'), 100)
```

rnalab_scatterplot *Plot Scatterplots for input data*

Description

Use ggplot2 to generate Scatterplots for input data. Allows the option to plot a linear regression fit.

Usage

```
rnalab_scatterplot(data, x, y, fit)
```

Arguments

data	Dataset to use for ggplot2 plots
x, y	The x and y variables to be used in the aes() mapping for ggplot scatterplots
fit	A boolean indicating whether a linear regression fit should be added to the ggplot scatterplot

Value

A grid graphical object consisting of a ggplot2 jitter plot with or without a linear regression fit.

Examples

```
# Plot a scatter plot without linear regression added (using data from the dnaseqs data set)
rnalab_scatterplot(dnaseqs, x = 'length', y = 'yield', fit = FALSE)
# Plot a scatter plot with linear regression added (using data from the dnaseqs data set)
rnalab_scatterplot(dnaseqs, x = 'length', y = 'yield', fit = TRUE)
```

---

runRNAapp *Run the rnalab Shiny App*

Description

Run the rnalab Shiny App

Usage

```
runRNAapp()
```

Index