Package ‘Plasmidprofiler’

October 12, 2022

Type Package

Title Visualization of Plasmid Profile Results

Version 0.1.6

Maintainer Adrian Zetner <adrian.zetner@phac-aspc.gc.ca>

Description Contains functions developed to combine the results of querying a plasmid database using short-read sequence typing with the results of a blast analysis against the query results.

Depends R (>= 3.1.2)

SystemRequirements Pandoc (>= 1.15)

License Apache License 2.0

LazyData TRUE

RoxygenNote 5.0.1

Suggests lintr

Imports ape, dplyr, gdata, ggdendro, ggplot2, grid, gridExtra, gtable, htmlwidgets, magrittr, plotly, plyr, RColorBrewer, reshape2, stringr

NeedsCompilation no

Author Adrian Zetner [aut, cre]

Repository CRAN

Date/Publication 2017-01-06 01:10:47

R topics documented:

`amr_positives` *Identify Antimicrobial Resistance Positive Plasmids from Blast Results*

Description

This function loads the imported blast results, identifies which plasmids carry AMR genes at highest identity. May have issues with multiple genes per plasmid, currently optimized for identifying one of two genes

Usage

```
amr_positives(blast.results)
```

Arguments

`blast.results`	Blast results loaded from read_blast or from Global Env

Value

Two column DF of plasmid names and genes present

Examples

```
## Not run:
amr_positives(blastdata)

## End(Not run)
```

`amr_presence` *Adds the AMR_gene column to report*

Description

Appends the results of amr_positives to the report in column AMR_gene, missing have "-" instead

Usage

```
amr_presence(report, pos.samples)
```

Arguments

`report`	Dataframe of results produced by `subsampler` or `combine_results`
`pos.samples`	Two column DF of plasmid names and genes present produced by `amr_positives`

Value

Report with AMR_genes added

See Also

`subsampler`, `combine_results`

Examples

```
## Not run:
amr_presence(report, pos.samples)

## End(Not run)
```

`blastdata` *Example Table of Blast Results*

Description

Example Table of Blast Results

Usage

```
data(blastdata)
```

Format

Dataframe.

Source

Strains graciously provided by the authors of the following papers: Complete Genome and Plasmid Sequences of Three Canadian Isolates of Salmonella enterica subsp. enterica Serovar Heidelberg from Human and Food Sources. 2016 Labbe et al. PMID: 26769926

Complete Sequence of Four Multidrug-Resistant MOBQ1 Plasmids Harboring blaGES-5 Isolated from Escherichia coli and Serratia marcescens Persisting in a Hospital in Canada. 2015 Boyd et al. PMID: 25545311

Colistin-Nonsusceptible Pseudomonas aeruginosa Sequence Type 654 with blaNDM-1 Arrives in North America. 2016 Mataseje et al. PMID: 26824951

References

None Yet (PubMed)

Examples

```
data(blastdata)
```

`blast_parser` *Blast Results Parser Function*

Description

Loads the imported blast results, extracts desired columns, Create new column of ratio between hit length to query length - higher as denominator, adjusts pID by this ratio. Any AMR results are removed from the returned df.

Usage

```
blast_parser(blast.results)
```

Arguments

`blast.results`	Blast results loaded from read_blast or Global Env

Value

Blast table with pID adjusted by ratio of hit length to query length (larger as denominator)

Examples

```
## Not run:
blast_parser(blastdata)

## End(Not run)
```

combine_results *Combines SRST2 and Blast results into a single dataframe*

Description

Combines blast and SRST2 results, cuts to desired columns (Sample, Plasmid, Inc_group, Coverage, Divergence, Length, Clusterid), matches plasmids to BR and appends simplified INC names, all future modifications are done to this dataframe

Usage

```
combine_results(sr, br)
```

Arguments

sr	SRST2 results loaded from read_srst2
br	Blast results parsed by blast_parser

Value

Seven column dataframe of SRST2 results now including INC groups

Examples

```
## Not run:
combine_results(example_srst2_results, example_blast_results)

## End(Not run)
```

create_grob *Create Heatmap Graphical Object*

Description

Combines the tree, heatmap, and titles to create final heatmap image.

Usage

```
create_grob(report, grob.title = "Plasmid Profiles")
```

Arguments

report	Dataframe of results
grob.title	Title of heatmap

Value

Composite image

Examples

```
## Not run:
create_grob(report, grob.title="Plasmid Profiles")

## End(Not run)
```

---

create_plotly *Create Plotly Object*

Description

Builds the heatmap, creates final interactive plot.

Usage

```
create_plotly(report, user, api.key, post = NA, title = "Plasmid Profiles",
  len.highlight = NA)
```

Arguments

report	Dataframe of results
user	User ID for plotly web publishing
api.key	API key for plotly web publishing
post	Flag determines whether or not to post to plotly (default NA, no post)
title	Title of heatmap
len.highlight	If anything but NA will highlight the largest plasmid hit per incompatibility group

Value

plotly object

Examples

```
## Not run:
create_plotly(report, title="Plasmid Profiles")

## End(Not run)
```

define_colours *Defining Colours Based on a Column of Data*

Description

```
This function uses RColorBrewer to produce palettes based on the factor levels of the identified column i
```

Usage

```
define_colours(report, column)
```

Arguments

report	Dataframe of results produced by subsampler or combine_results
column	Specify a column by name

Value

Named vector of colours, names are factor levels of column supplied

Examples

```
## Not run:
define_colours(report, "AMR_gene")

## End(Not run)
```

file_cacher *Filecacher*

Description

Creates filecache environment if needed for transferring variables between functions.

Usage

```
file_cacher()
```

main *Main: Run everything*

Description

Run all the interim functions to produce outputs. Can be run in order individually if desired.

1. `read_blast` Import the blast file, add column names
2. `blast_parser` Parse imported file
3. `amr_positives` Detect AMR positive plasmids
4. `read_srst2` Import SRST2 file
5. `combine_results` Combine SRST2 and Blast
6. `zetner_score` Add Sureness value
7. `amr_presence` Add detected AMR to report
8. `subsampler` Apply filters to report
9. `order_report` Arrange report
10. `save_files` Save JPG and CSV
11. `create_plotly` Creates plot
12. `save_files` Save HTML plot

Usage

```
main(blast.file, srst2.file, coverage.filter = NA, sureness.filter = NA,
  length.filter = NA, combine.inc = NA, plotly.user, plotly.api,
  post.plotly = NA, anonymize = NA, main.title = "Plasmid Profiles")
```

Arguments

`blast.file` Either system location of blast results (tsv) or dataframe

`srst2.file` Either system location of srst2 results (tsv) or dataframe

`coverage.filter`
Filters results below percent read coverage specified (eg. 80)

`sureness.filter`
Filters results below sureness specified (eg. 0.75)

`length.filter` Filters plasmid sequences shorter than length specified (eg. 10000)

`combine.inc` Flag to combine incompatibility sub-groups into their main type (set to 1)

`plotly.user` Enter your plotly info to upload to (Plotly)

`plotly.api` Enter your plotly info to upload to (Plotly)

`post.plotly` Flag to post to (Plotly)

`anonymize` Flag to post to anonymize plasmids and samples (set to 1)

`main.title` A title for the figure

Value

Saves output files in working directory

Examples

```
main(blastdata,
srst2data,
coverage.filter=NA,
sureness.filter=0.75,
length.filter=10000,
main.title="Example Results")
```

minmax *Minmax*

Description

Takes two columns of numerical data, normalizes it to ranges from 0 to 1 (0 to -1 for minimums), sums them, arranges by sum, then returns the sorted dataframe

Usage

```
minmax(df, maxcol, mincol)
```

Arguments

`df`	Dataframe
`maxcol`	Column to normalize from 0 to 1
`mincol`	Column to normalize from 0 to -1

Value

Dataframe sorted by sum of maxcol and mincol

Examples

```
## Not run:
 minmax(report, "Length", "Coverage")

## End(Not run)
```

normalize *Normalize*

Description

Normalizes a vector of values to a range of 0-1 x - min(x)) / (max(x) - min(x)

Usage

```
normalize(x)
```

Arguments

x	Vector of values

Value

Normalized vector of values

Examples

```
## Not run:
 normalize(x)

## End(Not run)
```

order_report *Order the Report*

Description

Order the report first by sample order (tree), then by incompatibility group, then by sureness on each plasmid

Usage

```
order_report(report, anonymize = NA)
```

Arguments

report	Dataframe of results produced by subsampler or combine_results
anonymize	Flag to anything other than NA to replace plasmid and sample names with generic names

Value

Ordered report

See Also

subsampler, combine_results

Examples

```
## Not run:
order_report(report)

## End(Not run)
```

plot_heatmap *Create GGPLOT Heatmap*

Description

Using a ggplot2 tile geometry this function will create a heatmap of values in the report coloured by incompatibility group, with alpha values from the sureness score. The order of samples is determined by order_report and plasmids by incompatibility group and sureness score.

Usage

```
plot_heatmap(report, len.highlight = NA)
```

Arguments

report	Dataframe of results
len.highlight	If anything but NA will highlight the largest plasmid hit per incompatibility group

Value

GGPLOT plotted heatmap

Examples

```
## Not run:
plot_heatmap(report)

## End(Not run)
```

read_blast *Blast file import function*

Description

This function imports the 25 column blast file and adds column headers

Usage

```
read_blast(br.file)
```

Arguments

`br.file`	System location of the blast file, no default.

Value

Dataframe of blast data with correct column headers.

Examples

```
## Not run:
read_blast("/data/blast_results.tsv")

## End(Not run)
```

read_srst2 *SRST2 file import function*

Description

This function imports the 14 column SRST2 file. Kind of superfluous

Usage

```
read_srst2(srst2.file)
```

Arguments

`srst2.file`	System location of the srst2 file, no default.

Value

Dataframe of srst2 data with correct column headers.

Examples

```
## Not run:
read_srst2("/data/srst2_results.tsv")

## End(Not run)
```

`report` *Example Complete Report after the following steps. Blast data from attached blastdata table SRST2 data from attached srst2data table*

Description

read_blast Import the blast file, add column names blast_parser Parse imported file amr_positives Detect AMR positive plasmids read_srst2 Import SRST2 file combine_results Combine SRST2 and Blast zetner_score Add Sureness value amr_presence Add detected AMR to report order_report Arrange report

Usage

```
data(report)
```

Format

Dataframe.

Source

Strains graciously provided by the authors of the following papers: Complete Genome and Plasmid Sequences of Three Canadian Isolates of Salmonella enterica subsp. enterica Serovar Heidelberg from Human and Food Sources. 2016 Labbe et al. PMID: 26769926

Complete Sequence of Four Multidrug-Resistant MOBQ1 Plasmids Harboring blaGES-5 Isolated from Escherichia coli and Serratia marcescens Persisting in a Hospital in Canada. 2015 Boyd et al. PMID: 25545311

Colistin-Nonsusceptible Pseudomonas aeruginosa Sequence Type 654 with blaNDM-1 Arrives in North America. 2016 Mataseje et al. PMID: 26824951

References

None Yet (PubMed)

Examples

```
data(report)
```

save_files *Save Files*

Description

Save various files: JPG, CSV, HTML depending on parameters

Usage

```
save_files(report, plot.png = NA, report.csv = NA, webpage = NA,
  title = "Plasmid Profiles")
```

Arguments

`report`	Dataframe of results
`plot.png`	Do you want to save a png? (Anything but NA)
`report.csv`	Do you want to save a text report? (Anything but NA)
`webpage`	Do you want to save an interactive heatmap as html? (Anything but NA)
`title`	Enter a title for the plot

Value

Named vector of colours, names are factor levels of column supplied

Examples

```
## Not run:
 save_files(report, plot.png=1, report.csv=1, webpage=NA)

## End(Not run)
```

srst2data *Example Table of SRST2 Results*

Description

Example Table of SRST2 Results

Usage

```
data(srst2data)
```

Format

Dataframe.

Source

Strains graciously provided by the authors of the following papers: Complete Genome and Plasmid Sequences of Three Canadian Isolates of Salmonella enterica subsp. enterica Serovar Heidelberg from Human and Food Sources. 2016 Labbe et al. PMID: 26769926

Complete Sequence of Four Multidrug-Resistant MOBQ1 Plasmids Harboring blaGES-5 Isolated from Escherichia coli and Serratia marcescens Persisting in a Hospital in Canada. 2015 Boyd et al. PMID: 25545311

Colistin-Nonsusceptible Pseudomonas aeruginosa Sequence Type 654 with blaNDM-1 Arrives in North America. 2016 Mataseje et al. PMID: 26824951

References

None Yet (PubMed)

Examples

```
data(srst2data)
```

`subsampler` *Subsetting Results*

Description

```
Several filters can be applied:
   Coverage: Filters results below percent read coverage specified
          eg. 95.9 cuts results where reads covered less than 95.9% of the total length
   Sureness: Filters results below sureness specified
               eg. 0.9 cuts results where the sureness falls below 0.9
   Length:   Filters plasmid sequences shorter than length specified
               eg. 10000 cuts out results where the plasmid was less than 10kb
  Incompatibility groups can also be combined (eg. Fii(S) and Fii(K) are combined into Fii)
```

Usage

```
subsampler(report, cov.filter = NA, sure.filter = NA, len.filter = NA,
  inc.combine = NA)
```

Arguments

`report`	Dataframe of results produced by `subsampler` or `combine_results`
`cov.filter`	Filters results below percent read coverage specified (eg. 80)
`sure.filter`	Filters results below sureness specified (eg. 0.75)
`len.filter`	Filters plasmid sequences shorter than length specified (eg. 10000)
`inc.combine`	Flag to ombine incompatibility sub-groups into their main type (set to 1)

Value

Report with filters applied

See Also

subsampler, combine_results

Examples

```
## Not run:
subsampler(report, sureness.filter = 0.75, len.filter = 10000)

## End(Not run)
```

tree_maker *Create Dendrogram Based on Plasmid Content*

Description

Reads report, converts to matrix of Sample ~ Plasmid with Sureness as cell values. Performs a hierarchical cluster analysis on a set of dissimilarities derived from the matrix. Creates a dendrogram from this data. Returns either the HC data or the dendrogram plot

Usage

```
tree_maker(report, hc.only = NA)
```

Arguments

report	Dataframe of results produced by subsampler or combine_results
hc.only	Flag to return only hierarchical clustering results instead of dendrogram plot (set to 1)

Value

Dendrogram object or hierarchical clustering results

See Also

subsampler, combine_results

Examples

```
## Not run:
tree_maker(report)

## End(Not run)
```

`zetner_score` *Adds the Zetner Score column to report*

Description

Runs mimmax function on Coverage and Divergence, returns sum of normalized Coverage with negative normalized Divergence a value which is then normalized from 0 to 1.

Usage

```
zetner_score(report)
```

Arguments

`report`	Dataframe of results produced by `subsampler` or `combine_results`

Value

Report with zetner score added

See Also

`subsampler`, `combine_results`

Examples

```
## Not run:
zetner_score(report)

## End(Not run)
```

Index