

# Package ‘Plasmidprofiler’

July 21, 2025

**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

## Contents

|                           |   |
|---------------------------|---|
| amr_positives . . . . .   | 2 |
| amr_presence . . . . .    | 3 |
| blastdata . . . . .       | 3 |
| blast_parser . . . . .    | 4 |
| combine_results . . . . . | 5 |
| create_grob . . . . .     | 5 |
| create_plotly . . . . .   | 6 |
| define_colours . . . . .  | 7 |
| file_cacher . . . . .     | 7 |

|                        |           |
|------------------------|-----------|
| main . . . . .         | 8         |
| minmax . . . . .       | 9         |
| normalize . . . . .    | 10        |
| order_report . . . . . | 10        |
| plot_heatmap . . . . . | 11        |
| read_blast . . . . .   | 12        |
| read_srst2 . . . . .   | 12        |
| report . . . . .       | 13        |
| save_files . . . . .   | 14        |
| srst2data . . . . .    | 14        |
| subsampler . . . . .   | 15        |
| tree_maker . . . . .   | 16        |
| zetner_score . . . . . | 17        |
| <b>Index</b>           | <b>18</b> |

---

|               |                                                                               |
|---------------|-------------------------------------------------------------------------------|
| amr_positives | <i>Identify Antimicrobial Resistance Positive Plasmids from Blast Results</i> |
|---------------|-------------------------------------------------------------------------------|

---

**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 | <i>Adds the AMR_gene column to report</i> |
|--------------|-------------------------------------------|

---

**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 <a href="#">subsampler</a> or <a href="#">combine_results</a> |
| pos.samples | Two column DF of plasmid names and genes present produced by <a href="#">amr_positives</a>     |

**Value**

Report with AMR\_genes added

**See Also**

[subsampler](#), [combine\\_results](#)

**Examples**

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

---

|           |                                       |
|-----------|---------------------------------------|
| blastdata | <i>Example Table of Blast Results</i> |
|-----------|---------------------------------------|

---

**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 | <i>Blast Results Parser Function</i> |
|--------------|--------------------------------------|

---

## 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 | <i>Combines SRST2 and Blast results into a single dataframe</i> |
|-----------------|-----------------------------------------------------------------|

---

**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 | <i>Create Heatmap Graphical Object</i> |
|-------------|----------------------------------------|

---

**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 | <i>Create Plotly Object</i> |
|---------------|-----------------------------|

---

**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 | <i>Defining Colours Based on a Column of Data</i> |
|----------------|---------------------------------------------------|

---

**Description**

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

**Usage**

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

**Arguments**

|        |                                                                                                |
|--------|------------------------------------------------------------------------------------------------|
| report | Dataframe of results produced by <a href="#">subsampler</a> or <a href="#">combine_results</a> |
| 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 | <i>Filecacher</i> |
|-------------|-------------------|

---

**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

|                              |                                                                            |
|------------------------------|----------------------------------------------------------------------------|
| <code>blast.file</code>      | Either system location of blast results (tsv) or dataframe                 |
| <code>srst2.file</code>      | Either system location of srst2 results (tsv) or dataframe                 |
| <code>coverage.filter</code> | Filters results below percent read coverage specified (eg. 80)             |
| <code>sureness.filter</code> | Filters results below sureness specified (eg. 0.75)                        |
| <code>length.filter</code>   | Filters plasmid sequences shorter than length specified (eg. 10000)        |
| <code>combine.inc</code>     | Flag to combine incompatibility sub-groups into their main type (set to 1) |
| <code>plotly.user</code>     | Enter your plotly info to upload to ( <a href="#">Plotly</a> )             |
| <code>plotly.api</code>      | Enter your plotly info to upload to ( <a href="#">Plotly</a> )             |
| <code>post.plotly</code>     | Flag to post to ( <a href="#">Plotly</a> )                                 |
| <code>anonymize</code>       | Flag to post to anonymize plasmids and samples (set to 1)                  |
| <code>main.title</code>      | 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 | <i>Minmax</i> |
|--------|---------------|

---

**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 | <i>Normalize</i> |
|-----------|------------------|

---

**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 | <i>Order the Report</i> |
|--------------|-------------------------|

---

**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 <a href="#">subsampler</a> or <a href="#">combine_results</a> |
| 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 GGLOT 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**

|                            |                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------|
| <code>report</code>        | Dataframe of results                                                                |
| <code>len.highlight</code> | 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 | <i>Blast file import function</i> |
|------------|-----------------------------------|

---

**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 | <i>SRST2 file import function</i> |
|------------|-----------------------------------|

---

**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 | <i>Subsetting Results</i> |
|------------|---------------------------|

---

## 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 <a href="#">subsampler</a> or <a href="#">combine_results</a> |
| 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 <a href="#">subsampler</a> or <a href="#">combine_results</a> |
| 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 | <i>Adds the Zetner Score column to report</i> |
|--------------|-----------------------------------------------|

---

**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

## \* datasets

- blastdata, [3](#)
- report, [13](#)
- srst2data, [14](#)

amr\_positives, [2](#), [3](#), [8](#)  
amr\_presence, [3](#), [8](#)

blast\_parser, [4](#), [8](#)  
blastdata, [3](#)

combine\_results, [3](#), [5](#), [7](#), [8](#), [10](#), [11](#), [15–17](#)  
create\_grob, [5](#)  
create\_plotly, [6](#), [8](#)

define\_colours, [7](#)

file\_cacher, [7](#)

main, [8](#)  
minmax, [9](#)

normalize, [10](#)

order\_report, [8](#), [10](#)

plot\_heatmap, [11](#)

read\_blast, [8](#), [12](#)  
read\_srst2, [8](#), [12](#)  
report, [13](#)

save\_files, [8](#), [14](#)  
srst2data, [14](#)  
subsampler, [3](#), [7](#), [8](#), [10](#), [11](#), [15](#), [15](#), [16](#), [17](#)

tree\_maker, [16](#)

zetner\_score, [8](#), [17](#)