Package 'ogrdbstats'

November 3, 2024

Type Package

Title Analysis of Adaptive Immune Receptor Repertoire Germ Line Statistics

Version 0.5.2

URL https://github.com/airr-community/ogrdbstats

BugReports https://github.com/airr-community/ogrdbstats/issues

Description Multiple tools are now available for inferring the personalised germ line set from an adaptive immune receptor repertoire. Output from these tools is converted to a single format and supplemented with rich data such as usage and characterisation of 'novel' germ line alleles. This data can be particularly useful when considering the validity of novel inferences. Use of the analysis provided is described in <doi:10.3389/fimmu.2019.00435>.

License CC BY-SA 4.0

Encoding UTF-8

Depends R (>= 2.10)

Imports dplyr (>= 0.8.3), ggplot2 (>= 3.2.1), magrittr, tigger (>= 0.4.0), alakazam (>= 0.3.0), stringr (>= 1.4.0), data.table, gridExtra (>= 2.3), tidyr (>= 1.0.0), stringdist (>= 0.9.5.2), RColorBrewer (>= 1.1-2), Biostrings (>= 2.52.0), argparser (>= 0.4), ComplexHeatmap, bookdown, scales,

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

LazyData true

NeedsCompilation no

Author William Lees [aut, cre] (<https://orcid.org/0000-0001-9834-6840>)

Maintainer William Lees <william@lees.org.uk>

Repository CRAN

Date/Publication 2024-11-03 06:10:02 UTC

Contents

`example_rep` *Example repertoire data*

Description

A small example of the analytical datasets created by ogrdbstats from repertoires and reference sets. The dataset can be created by running the example shown for the function read_input_data(). The dataset is created from example files provided with the package. The repertoire data is taken from Rubelt et al. 2016, <doi: 10.1038/ncomms11112>

Usage

```
example_rep
```

Format

'example_rep' - a named list containing the following elements:

ref_genes	named list of IMGT-gapped reference genes
inferred_seqs	named list of IMGT-gapped inferred (novel) sequences.
input_sequences	data frame with one row per annotated read, with CHANGEO-style column names. The column SEG_CAI
genotype_db	named list of gene sequences referenced in the annotated reads (both reference and novel sequences)
haplo_details	data used for haplotype analysis, showing allelic ratios calculated with various potential haplotyping genes
genotype	data frame containing information provided in the OGRDB genotype csv file
calculated_NC	a boolean that is TRUE if mutation counts were calculated by this library, FALSE if they were read from th

Source

<doi: 10.1038/ncomms11112>

`generate_ogrdb_report` *Generate OGRDB reports from specified files.*

Description

This creates the genotype report (suffixed _ogrdb_report.csv) and the plot file (suffixed _ogrdb_plos.pdf). Both are created in the directory holding the annotated read file, and the file names are prefixed by the name of the annotated read file.

Usage

```
generate_ogrdb_report(
  ref_filename,
  inferred_filename,
  species,
  filename,
  chain,
  hap_gene,
  segment,
  chain_type,
  plot_unmutated,
  all_inferred = FALSE,
  format = "pdf"
)
```

Arguments

`ref_filename`	Name of file containing IMGT-aligned reference genes in FASTA format
`inferred_filename`	Name of file containing sequences of inferred novel alleles, or '-' if none
`species`	Species name used in field 3 of the IMGT germline header with spaces omitted, if the reference file is from IMGT. Otherwise "
`filename`	Name of file containing annotated reads in AIRR, CHANGEO or IgDiscover format. The format is detected automatically
`chain`	one of IGHV, IGKV, IGLV, IGHD, IGHJ, IGKJ, IGLJ, TRAV, TRAj, TRBV, TRBD, TRBJ, TRGV, TRGj, TRDV, TRDD, TRDJ
`hap_gene`	The haplotyping columns will be completed based on the usage of the two most frequent alleles of this gene. If NA, the column will be blank
`segment`	one of V, D, J
`chain_type`	one of H, L
`plot_unmutated`	Plot base composition using only unmutated sequences (V-chains only)
`all_inferred`	Treat all alleles as novel
`format`	The format for the plot file ('pdf', 'html' or 'none')

Value

None

Examples

```
# prepare files for example
reference_set = system.file("extdata/ref_gapped.fasta", package = "ogrdbstats")
inferred_set = system.file("extdata/novel_gapped.fasta", package = "ogrdbstats")
repertoire = system.file("extdata/ogrdbstats_example_repertoire.tsv", package = "ogrdbstats")
file.copy(repertoire, tempdir())
repfile = file.path(tempdir(), 'ogrdbstats_example_repertoire.tsv')

generate_ogrdb_report(reference_set, inferred_set, 'Homosapiens',
          repfile, 'IGHV', NA, 'V', 'H', FALSE, format='none')

#clean up
outfile = file.path(tempdir(), 'ogrdbstats_example_repertoire_ogrdb_report.csv')
file.remove(repfile)
file.remove(outfile)
```

---

genotype_statistics_cmd

Collect parameters from the command line and use them to create a report and CSV file

---

Description

Collect parameters from the command line and use them to create a report and CSV file

Usage

```
genotype_statistics_cmd(args = NULL)
```

Arguments

args	A string vector containing the command line arguments. If NULL, will take them from the command line

Value

Nothing

Examples

```
# Prepare files for example
reference_set = system.file("extdata/ref_gapped.fasta", package = "ogrdbstats")
inferred_set = system.file("extdata/novel_gapped.fasta", package = "ogrdbstats")
repertoire = system.file("extdata/ogrdbstats_example_repertoire.tsv", package = "ogrdbstats")
file.copy(repertoire, tempdir())
```

```
repfile = file.path(tempdir(), 'repertoire.tsv')

genotype_statistics_cmd(c(
              reference_set,
              'Homosapiens',
              repfile,
              'IGHV',
              '--inf_file', inferred_set,
              '--format', 'none'))

# clean up
outfile = file.path(tempdir(), 'repertoire_ogrdb_report.csv')
plotdir = file.path(tempdir(), 'repertoire_ogrdb_plots')
file.remove(repfile)
file.remove(outfile)
unlink(plotdir, recursive=TRUE)
```

`make_barplot_grobs` *Create a barplot for each allele, showing number of reads distributed by mutation count*

Description

Create a barplot for each allele, showing number of reads distributed by mutation count

Usage

```
make_barplot_grobs(
  input_sequences,
  genotype_db,
  inferred_seqs,
  genotype,
  segment,
  calculated_NC
)
```

Arguments

`input_sequences`	the input_sequences data frame
`genotype_db`	named list of gene sequences in the personalised genotype
`inferred_seqs`	named list of novel gene sequences
`genotype`	data frame created by calc_genotype
`segment`	one of V, D, J
`calculated_NC`	a boolean, TRUE if mutation counts had to be calculated, FALSE otherwise

Value

list of grobs

Examples

```
barplot_grobs = make_barplot_grobs(
                      example_rep$input_sequences,
                      example_rep$genotype_db,
                      example_rep$inferred_seqs,
                      example_rep$genotype,
                      'V',
                      example_rep$calculated_NC
                )
```

`make_haplo_grobs` *Create haplotyping plots*

Description

Create haplotyping plots

Usage

```
make_haplo_grobs(segment, haplo_details)
```

Arguments

`segment`	one of V, D, J
`haplo_details`	Data structure created by create_haplo_details

Value

named list containing the following elements:

a_allele_plot	plot showing allele usage for each potential haplotyping gene
haplo_grobs	differential plot of allele usage for each usable haplotyping gene

Examples

```
haplo_grobs = make_haplo_grobs('V', example_rep$haplo_details)
```

`make_novel_base_grobs` *Create plots showing base usage at selected locations in sequences based on novel alleles*

Description

Create plots showing base usage at selected locations in sequences based on novel alleles

Usage

```
make_novel_base_grobs(inferred_seqs, input_sequences, segment, all_inferred)
```

Arguments

`inferred_seqs`	named list of novel gene sequences
`input_sequences`	the input_sequences data frame
`segment`	one of V, D, J
`all_inferred`	true if user has requested all alleles in reference set plotted - will suppress some warnings

Value

named list containing the following elements:

cdr3_dist	cdr3 length distribution plots
whole	whole-length usage plots
end	3' end usage plots
conc	3' end consensus composition plots
triplet	3' end triplet usage plots

Examples

```
base_grobs = make_novel_base_grobs(
                example_rep$inferred_seqs,
                example_rep$input_sequences,
                'V',
                FALSE
            )
```

`read_input_files` *Read input files into memory*

Description

Read input files into memory

Usage

```
read_input_files(
  ref_filename,
  inferred_filename,
  species,
  filename,
  chain,
  hap_gene,
  segment,
  chain_type,
  all_inferred
)
```

Arguments

`ref_filename`	Name of file containing IMGT-aligned reference genes in FASTA format
`inferred_filename`	Name of file containing sequences of inferred novel alleles, or '-' if none
`species`	Species name used in field 3 of the IMGT germline header with spaces omitted, if the reference file is from IMGT. Otherwise ''
`filename`	Name of file containing annotated reads in AIRR, CHANGEO or IgDiscover format. The format is detected automatically
`chain`	one of IGHV, IGKV, IGLV, IGHD, IGHJ, IGKJ, IGLJ, TRAV, TRAj, TRBV, TRBD, TRBJ, TRGV, TRGj, TRDV, TRDD, TRDJ
`hap_gene`	The haplotyping columns will be completed based on the usage of the two most frequent alleles of this gene. If NA, the column will be blank
`segment`	one of V, D, J
`chain_type`	one of H, L
`all_inferred`	Treat all alleles as novel

Value

A named list containing the following elements:

ref_genes	named list of IMGT-gapped reference genes
inferred_seqs	named list of IMGT-gapped inferred (novel) sequences.

input_sequences	data frame with one row per annotated read, with CHANGEO-style column names One key point: the colu
genotype_db	named list of gene sequences referenced in the annotated reads (both reference and novel sequences)
haplo_details	data used for haplotype analysis, showing allelic ratios calculated with various potential haplotyping genes
genotype	data frame containing information provided in the OGRDB genotype csv file
calculated_NC	a boolean that is TRUE if mutation counts were calculated by this library, FALSE if they were read from th

Examples

```
# Create the analysis data set from example files provided with the package
#(this dataset is also provided in the package as example_rep)
reference_set = system.file("extdata/ref_gapped.fasta", package = "ogrdbstats")
inferred_set = system.file("extdata/novel_gapped.fasta", package = "ogrdbstats")
repertoire = system.file("extdata/ogrdbstats_example_repertoire.tsv", package = "ogrdbstats")

example_data = read_input_files(reference_set, inferred_set, 'Homosapiens',
       repertoire, 'IGHV', NA, 'V', 'H', FALSE)
```

`write_genotype_file` *Write the genotype file required by OGRDB*

Description

Write the genotype file required by OGRDB

Usage

```
write_genotype_file(filename, segment, chain_type, genotype)
```

Arguments

`filename`	name of file to create (csv)
`segment`	one of V, D, J
`chain_type`	one of H, L
`genotype`	genotype data frame

Value

None

Examples

```
genotype_file = tempfile("ogrdb_genotype")
write_genotype_file(genotype_file, 'V', 'H', example_rep$genotype)
file.remove(genotype_file)
```

`write_plot_file` *Create the OGRDB style plot file*

Description

Create the OGRDB style plot file

Usage

```
write_plot_file(
  filename,
  input_sequences,
  cdr3_dist_grobs,
  end_composition_grobs,
  cons_composition_grobs,
  whole_composition_grobs,
  triplet_composition_grobs,
  barplot_grobs,
  a_allele_plot,
  haplo_grobs,
  message,
  format
)
```

Arguments

`filename`	name of file to create (pdf)
`input_sequences`	the input_sequences data frame
`cdr3_dist_grobs`	cdr3 length distribution grobs created by make_novel_base_grob
`end_composition_grobs`	end composition grobs created by make_novel_base_grobs
`cons_composition_grobs`	consensus composition grobs created by make_novel_base_grobs
`whole_composition_grobs`	whole composition grobs created by make_novel_base_grobs
`triplet_composition_grobs`	triplet composition grobs created by make_novel_base_grobs
`barplot_grobs`	barplot grobs created by make_barplot_grons
`a_allele_plot`	a_allele_plot grob created by make_haplo_grobs
`haplo_grobs`	haplo_grobs created by make_haplo_grobs
`message`	text message to display at end of report
`format`	Format of report ('pdf', 'html' or 'none')

Value

None

Examples

```
plot_file = tempfile(pattern = 'ogrdb_plots')

base_grobs = make_novel_base_grobs(
                 example_rep$inferred_seqs,
                 example_rep$input_sequences,
                 'V',
                 FALSE
             )
barplot_grobs = make_barplot_grobs(
                      example_rep$input_sequences,
                      example_rep$genotype_db,
                      example_rep$inferred_seqs,
                      example_rep$genotype,
                      'V',
                      example_rep$calculated_NC
                )
haplo_grobs = make_haplo_grobs('V', example_rep$haplo_details)

write_plot_file(
    plot_file,
    example_rep$input_sequences,
    base_grobs$cdr3_dist,
    base_grobs$end,
    base_grobs$conc,
    base_grobs$whole,
    base_grobs$triplet,
    barplot_grobs,
    haplo_grobs$aplot,
    haplo_grobs$haplo,
    "Notes on this analysis",
    'none'
)

file.remove(plot_file)
```

Index