Package 'tripr'

May 29, 2024

Type Package

Title T-cell Receptor/Immunoglobulin Profiler (TRIP)

Version 1.10.0

Description TRIP is a software framework that provides analytics services on antigen receptor (B cell receptor immunoglobulin, BcR IG | T cell receptor, TR) gene sequence data. It is a web application written in R Shiny. It takes as input the output files of the IMGT/HighV-Quest tool. Users can select to analyze the data from each of the input samples separately, or the combined data files from all samples and visualize the results accordingly.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

biocViews BatchEffect, MultipleComparison, GeneExpression, ImmunoOncology, TargetedResequencing

Imports shinyjs, shinyFiles, plyr, data.table, DT, stringr, stringdist, plot3D, gridExtra, RColorBrewer, plotly, dplyr, config (>= 0.3.1), golem (>= 0.3.1), methods, grDevices, graphics, stats, utils

Enhances parallel

Suggests BiocGenerics, shinycssloaders, tidyverse, BiocManager, Biostrings, xtable, rlist, motifStack, knitr, rmarkdown, testthat (>= 3.0.0), fs, BiocStyle, RefManageR, biocthis, pryr

Depends shiny (>= 1.6.0), shinyBS

Collate ``tripr-package.R'' ``global.R'' ``helpers.R'' ``run_TRIP_without_ui.R'' ``app_config.R'' ``app_server.R'' ``app_ui.R'' ``run_app.R'' ``zzz.R''

URL https://github.com/BiodataAnalysisGroup/tripr

BugReports https://github.com/BiodataAnalysisGroup/tripr/issues

BiocType Software

RoxygenNote 7.2.0

VignetteBuilder knitr

Config/testthat/edition 3

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

git_branch RELEASE_3_19

git_last_commit 7bea195

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-28

Author Maria Th. Kotouza [aut],
Katerina Gemenetzi [aut],
Chrysi Galigalidou [aut],
Elisavet Vlachonikola [aut],
Nikolaos Pechlivanis [cre],
Andreas Agathangelidis [aut],
Raphael Sandaltzopoulos [aut],
Pericles A. Mitkas [aut],
Kostas Stamatopoulos [aut],
Anastasia Chatzidimitriou [aut],
Fotis E. Psomopoulos [aut],
Iason Ofeidis [aut],
Aspasia Orfanou [aut]

Maintainer Nikolaos Pechlivanis <inab.bioinformatics@lists.certh.gr>

Contents

run_app 2
run_TRIP 3
tripr 8

Index **9**

`run_app` *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_app(
  onStart = NULL,
  options = list(launch.browser = TRUE),
  enableBookmarking = NULL,
```

```
  uiPattern = "/",
  ...
)
```

Arguments

onStart	A function that will be called before the app is actually run. This is only needed for shinyAppObj, since in the shinyAppDir case, a global.R file can be used for this purpose.
options	Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app.
enableBookmarking	Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to enableBookmarking(). See enableBookmarking() for more information on bookmarking your app.
uiPattern	A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
...	arguments to pass to golem_opts. See '?golem::get_golem_options' for more details.

Value

None

Examples

```
if (interactive()) {
    run_app(options = list(launch.browser = FALSE))
}
```

---

run_TRIP *Run tripr analysis via R command line*

Description

run_TRIP() is a wrapper of {tripr} shiny analysis tool for use via R command line. Output of analysis is saved in *tripr/extdata/output* folder, where R libraries are saved (typically *R/library*).

Usage

```
run_TRIP(
  datapath = fs::path_package("extdata", "dataset", package = "tripr"),
  output_path = fs::path_home("Documents/tripr_output"),
  filelist = c("1_Summary.txt", "2_IMGT-gapped-nt-sequences.txt",
    "4_IMGT-gapped-AA-sequences.txt", "6_Junction.txt"),
  cell = "Bcell",
  throughput = "High Throughput",
  preselection = "1,4C:W",
  selection = "5",
  identity_range = "85:100",
  vgenes = "",
  dgenes = "",
  jgenes = "",
  cdr3_length_range = "",
  aminoacid = "",
  pipeline = "1",
  select_clonotype = "V Gene + CDR3 Amino Acids",
  highly_sim_params = paste0("1-1 2-1 3-1 4-1 5-1 6-1 7-1 8-1 9-1 10-1 11-1 ",
    "12-1 13-1 14-1 15-2 16-2 17-2 18-2 19-2 20-2 21-2 23-2 24-2 25-2 ",
    "26-2 27-2 28-2 29-3 30-3 31-3 32-3 33-3 34-3 35-3 36-3 37-3 38-3 ",
    "39-3 40-3 41-3 42-3 43-3 44-3 45-3 46-3 47-3 48-3 49-3 50-3,1,Yes"),
  shared_clonotypes_params = "reads,1,Yes",
  highly_shared_clonotypes_params = "reads,1,Yes",
  repertoires_params = "1,4,6",
  identity_groups = "85:97,97:99,99:100,100:100",
  multiple_values_params = "2:7,2:3,2:5,2:11",
  alignment_params = "1,both,1,2:20",
  mutations_params = "both,0.5,0.5,2:20"
)
```

Arguments

`datapath`	(character) The directory where the folders of the data is located. Note that every sample of the dataset must have **its own individual folder** and every sample folder must be in **one root folder**. Note that **every** file in the root folder will be used in the analysis. Supposedly the dataset is in user's *Documents/* folder, one could use: `fs::path_home("Documents", "dataset")`, with the help of path_home function. See the package vignette for more.
`output_path`	(character) The directory where the output data will be stored. Please provide a valid path, ideally the same way as datapath by using the path_home function. The default value points to *Documents/tripr_output* directory.
`filelist`	(character vector) The character vector of files of the IMGT output that will be used through the analysis from each sample.
`cell`	(character) 'Bcell' (default) or 'Tcell'.
`throughput`	(character) 'High Throughput' (default) or 'Low Throughput'.

`preselection` (character) Preselection options:
1 == Only take into account Functional V-Gene,
2 == Only take into account CDR3 with no Special Characters (X,*,#,.),
3 == Only take into account Productive Sequences,
4 == Only take into account CDR3 with valid start/end landmarks.,
For Preselection option 4, select start/end landmarks.,
Use the vertical line '|' to add more than one start or end landmarks,
Use comma ',' to seperate the list of options, use semicolon ':' to seperate start and end landmarks.

`selection` (character) Selection options:
5 == V-REGION identity 6 == Select Specific V Gene ,
7 == Select Specific J Gene ,
8 == Select Specific D Gene ,
9 == Select CDR3 length range ,
10 == Only select CDR3 containing specific amino-acid sequence.
Use comma ',' to seperate the list of options.

`identity_range` (character) V-REGION identity Use colon ':' to seperate identity low and high

`vgenes` (character) Filter in specific V Genes,
Separate the different V-Gene names with '|' e.g. TRBV11-2|TRBV29-1*03 (F)

`dgenes` (character) Filter in specific D Genes,
Separate the different D-Gene names with | e.g. TRBD2|TRBD1

`jgenes` (character) Filter in specific J Genes,
Separate the different J-Gene names with | e.g. TRBJ2-6|TRBJ2-2

`cdr3_length_range` (character) Filter in rows with CDR3 lengths within a range,
Use colon ':' to seperate identity low and high

`aminoacid` (character) Filter in rows with CDR3 containing specific amino-acid sequence

`pipeline` (character) Pipeline options:
1 == Clonotypes Computation,
2 == Highly Similar Clonotypes computation,
3 == Shared Clonotypes Computation,
4 == Highly Similar Shared Clonotypes Computation,
5 == Repertoires Extraction,
6 == Repertoires Comparison,
7 == Highly Similar Repertoires Extraction,
8 == Insert Identity groups,
9 == Somatic hypermutation status,
10 == CDR3 Distribution,
11 == Pi Distribution,
12 == Multiple value comparison,
13 == CDR3 with 1 length difference,
14 == Alignment,
15 == Somatic hypermutations,
16 == Logo,
17 == SHM normal,
18 == SHM High similarity,

19 == Diagnosis,
Use comma ',' to seperate the list of options

`select_clonotype`
(character) Compute clonotypes.
Select one the following options:
"V Gene + CDR3 Amino Acids",
"V Gene and Allele + CDR3 Amino Acids",
"V Gene + CDR3 Nucleotide",
"V Gene and Allele + CDR3 Nucleotide",
"J Gene + CDR3 Amino Acids",
"J Gene and Allele + CDR3 Amino Acids",
"J Gene + CDR3 Nucleotide",
"J Gene and Allele + CDR3 Nucleotide",
"CDR3 Amino Acids",
"CDR3 Nucleotide",
"Sequence

`highly_sim_params`
(character) Select number of missmatches, the threshold of the clonotype frequency and whether you want to take gene into account. Use dashes '-' to show the length of the CDR3 sequences and the number of allowed missmatches and spaces ' ' to separate. For the CDR3 lengths with not specified number of missmatches the default value is 1. Use comma ',' to separate the three options.

`shared_clonotypes_params`
(character) Shared clonotypes computation.
Select 'reads' of 'threshold' for clonotypes, the number of reads or the threshold percentage accordingly, and whether you want to take gene into account. Use comma ',' to seperate the 3 options

`highly_shared_clonotypes_params`
(character) Highly Similar Shared Clonotypes Computation
Select 'reads' of 'threshold' for clonotypes, the number of reads or the threshold percentage accordingly, and whether you want to take gene into account. Use comma ',' to seperate the 3 options

`repertoires_params`
(character) Repertoires Extraction
Options:
1 == V Gene
2 == V Gene and allele
3 == J Gene
4 == J Gene and allele
5 == D Gene
6 == D Gene and allele
Use comma ',' to seperate the selected options

`identity_groups`
(character) Insert identity groups
Insert low and high values as follows:
low_values:high_values
Seperate low_values and high_values using comma ','.

`multiple_values_params`
: (character) Multiple value comparison
Options:
1 == V GENE
2 == V GENE and allele
3 == J GENE
4 == J GENE and allele
5 == D GENE
6 == D GENE and allele
7 == CDR3-IMGT length
8 == D-REGION reading frame
9 == Molecular mass
10 == pI
11 == V-REGION identity Use colon ':' to indicate combinations of 2 values, use comma "," to seperate the selected options

`alignment_params`
: (character) Alignment parameters:
Region for Alignment: 1 == V.D.J.REGION or 2 == V.J.REGION
AA or Nt: Select 'aa' or 'nt' or 'both'
Germline: 1 == Use Allele's germline or 2 == Use Gene's germline
Use: 1 == All clonotypes or 2 == Select top N clonotypes or 3 == Select threshold for clonotypes
Use comma ',' to seperate the 4 parameters. If you select option 2 or 3 at the 4th parameter you have to set the N or the threshold as well using colon ':'.

`mutations_params`
: (character) Somatic hypermutations parameters:
AA or Nt: Select 'aa' or 'nt' or 'both'
Set threshold for AA
Set threshold for Nt
Use: 1 == All clonotypes or 2 == Select top N clonotypes or 3 == Select threshold for clonotypes
Use comma ',' to seperate the 3 parameters. If you select option 2 or 3 at the 3rd parameter you have to set the N or the threshold as well using colon ':'.

Value

None

Examples

```
## Do not run

run_TRIP(
   output_path=tools::R_user_dir("tripr", which="cache"),
   filelist=c("1_Summary.txt", "2_IMGT-gapped-nt-sequences.txt",
       "4_IMGT-gapped-AA-sequences.txt", "6_Junction.txt"),
   cell="Bcell",
   throughput="High Throughput",
   preselection="1,2,3,4C:W",
```

```
    selection="5",
    identity_range="88:100",
    cdr3_length_range="",
    pipeline="1",
    select_clonotype="V Gene + CDR3 Amino Acids")
```

`tripr` *tripr*

Description

T-cell Receptor/Immunoglobulin Profiler (TRIP)

Details

The only function you're likely to need from tripr is [run_app()]. Otherwise refer to the vignettes for using tripr.

Index

enableBookmarking(), *3*

path_home, *4*

run_app, 2
run_TRIP, 3

tripr, 8