Package ‘scoper’

August 10, 2020

Type Package
Version 1.1.0
Date 2020-08-10
Title Spectral Clustering-Based Method for Identifying B Cell Clones
Description Provides a computational framework for identification of B cell clones from Adaptive Immune Receptor Repertoire sequencing (AIRR-Seq) data. Three main functions are included (identicalClones, hierarchicalClones, and spectralClones) that perform clustering among sequences of BCRs/IGs (B cell receptors/immunoglobulins) which share the same V gene, J gene and junction length.
License AGPL-3
URL https://scoper.readthedocs.io
BugReports https://bitbucket.org/kleinstein/scoper/issues
LazyData true
BuildVignettes true
VignetteBuilder knitr
Encoding UTF-8
SystemRequirements C++11
Depends R (>= 3.5.0), ggplot2 (>= 3.2.0)
Imports alakazam (>= 1.0.2), shazam (>= 1.0.1), data.table, doParallel, dplyr (>= 0.8.1), foreach, methods, Rcpp (>= 0.12.12), rlang, scales, stats, stringi, tidyr (>= 1.0.0)
LinkingTo Rcpp
Suggests knitr, rmarkdown, testthat
RoxygenNote 7.1.1
Collate 'Data.R' 'Scoper.R' 'Functions.R' 'RcppExports.R'
NeedsCompilation yes
ExampleDb

Author  Nima Nouri [aut],
        Edel Aron [ctb],
        Jason Vander Heiden [aut, cre],
        Steven Kleinstein [aut, cph]

Maintainer  Jason Vander Heiden <jason.vanderheiden@gmail.com>

Repository  CRAN

Date/Publication  2020-08-10 21:50:02 UTC

R topics documented:

ExampleDb .................................................. 2
hierarchicalClones ........................................ 3
identicalClones ............................................ 5
plotCloneSummary ........................................... 8
scoper .......................................................... 9
ScoperClones-class ......................................... 10
spectralClones ............................................... 11

Index  15

ExampleDb  Example database

Description

A small example database subset from Laserson and Vigneault et al, 2014.

Usage

ExampleDb

Format

A data.frame with the following columns:

- sequence_id: Sequence identifier
- sequence_alignment: IMGT-gapped observed sequence.
- germline_alignment: IMGT-gapped germline sequence.
- germline_alignment_d_mask: IMGT-gapped germline sequence with N, P and D regions masked.
- v_call: V region allele assignments.
- v_call_genotyped: TiGER corrected V region allele assignment.
- d_call: D region allele assignments.
- j_call: J region allele assignments.
- junction: Junction region sequence.
- junction_length: Length of the junction region in nucleotides.
hierarchicalClones

References


hierarchicalClones  Hierarchical clustering method for clonal partitioning

Description

hierarchicalClones provides a hierarchical agglomerative clustering approach to infer clonal relationships in high-throughput Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) data. This approach clusters B or T cell receptor sequences based on junction region sequence similarity within partitions that share the same V gene, J gene, and junction length, allowing for ambiguous V or J gene annotations.

Usage

hierarchicalClones(
  db,
  threshold,
  method = c("nt", "aa"),
  linkage = c("single", "average", "complete"),
  normalize = c("len", "none"),
  junction = "junction",
  v_call = "v_call",
  j_call = "j_call",
  clone = "clone_id",
  cell_id = NULL,
  locus = "locus",
  only_heavy = TRUE,
  split_light = TRUE,
  first = FALSE,
  cdr3 = FALSE,
  mod3 = FALSE,
  max_n = 0,
  nproc = 1,
  verbose = FALSE,
  log = NULL,
  summarize_clones = TRUE
)

Arguments

db         data.frame containing sequence data.
threshold   a numeric scalar where the tree should be cut (the distance threshold for clonal grouping).
method

one of the "nt" for nucleotide based clustering or "aa" for amino acid based clustering.

linkage

available linkage are "single", "average", and "complete".

normalize

method of normalization. The default is "len", which divides the distance by the length of the sequence group. If "none" then no normalization if performed.

junction

color name of the column containing junction sequences. Also used to determine sequence length for grouping.

call

color name of the column containing the V-segment allele calls.

j_call

color name of the column containing the J-segment allele calls.

clone

the output column name containing the clonal cluster identifiers.

cell_id

name of the column containing cell identifiers or barcodes. If specified, grouping will be performed in single-cell mode with the behavior governed by the locus and only_heavy arguments. If set to NULL then the bulk sequencing data is assumed.

locus

name of the column containing locus information. Only applicable to single-cell data. Ignored if cell_id=NULL.

only_heavy

use only the IGH (BCR) or TRB/TRD (TCR) sequences for grouping. Only applicable to single-cell data. Ignored if cell_id=NULL.

split_light

split clones by light chains. Ignored if cell_id=NULL.

first

specifies how to handle multiple V(D)J assignments for initial grouping. If TRUE only the first call of the gene assignments is used. If FALSE the union of ambiguous gene assignments is used to group all sequences with any overlapping gene calls.

cdr3

if TRUE removes 3 nucleotides from both ends of "junction" prior to clustering (converts IMGT junction to CDR3 region). If TRUE this will also remove records with a junction length less than 7 nucleotides.

mod3

if TRUE removes records with a junction length that is not divisible by 3 in nucleotide space.

max_n

The maximum number of N characters to permit in the junction sequence before excluding the record from clonal assignment. Note, with linkage="single" non-informative positions can create artifactual links between unrelated sequences. Use with caution. Default is set to be zero. Set it as "NULL" for no action.

nproc

number of cores to distribute the function over.

verbose

if TRUE prints out a summary of each step cloning process. if FALSE (default) process cloning silently.

log

output path and filename to save the verbose log. The input file directory is used if path is not specified. The default is NULL for no action.

summarize_clones

if TRUE performs a series of analysis to assess the clonal landscape and returns a ScoperClones object. If FALSE then a modified input db is returned.
Value

If `summarize_clones=TRUE` (default) a `ScoperClones` object is returned that includes the clonal assignment summary information and a modified input `db` in the `db` slot that contains clonal identifiers in the specified clone column. If `summarize_clones=FALSE` modified `data.frame` is returned with clone identifiers in the specified clone column.

Single-cell data

To invoke single-cell mode the `cell_id` argument must be specified and the `locus` column must be correct. Otherwise, clustering will be performed with bulk sequencing assumptions, using all input sequences regardless of the values in the `locus` column.

Values in the `locus` column must be one of `c("IGH", "IGI", "IGK", "IGL")` for BCR or `c("TRA", "TRB", "TRD", "TRG")` for TCR sequences. Otherwise, the operation will exit and return an error message.

Under single-cell mode with paired-chain sequences, there is a choice of whether grouping should be done by (a) using `IGH` (BCR) or `TRB/TRD` (TCR) sequences only or (b) using `IGH` plus `IGK/IGL` (BCR) or `TRB/TRD` plus `TRA/TRG` (TCR) sequences. This is governed by the `only_heavy` argument. There is also choice as to whether inferred clones should be split by the light/short chain (`IGK`, `IGL`, `TRA`, `TRG`) following heavy/long chain clustering, which is governed by the `split_light` argument.

In single-cell mode, clonal clustering will not be performed on data were cells are assigned multiple heavy/long chain sequences (`IGH`, `TRB`, `TRD`). If observed, the operation will exit and return an error message. Cells that lack a heavy/long chain sequence (i.e., cells with light/short chains only) will be assigned a `clone_id` of `NA`.

See Also

See `plotCloneSummary` for plotting summary results. See `groupGenes` for more details about grouping requirements.

Examples

```r
# Find clonal groups
results <- hierarchicalClones(ExampleDb, threshold=0.15)

# Retrieve modified input data with clonal clustering identifiers
df <- as.data.frame(results)

# Plot clonal summaries
plot(results, binwidth=0.02)
```

---

`identicalClones`  
**Sequence identity method for clonal partitioning**
**Description**

`identicalClones` provides a simple sequence identity based partitioning approach for inferring clonal relationships in high-throughput Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) data. This approach partitions B or T cell receptor sequences into clonal groups based on junction region sequence identity within partitions that share the same V gene, J gene, and junction length, allowing for ambiguous V or J gene annotations.

**Usage**

```r
identicalClones(
  db,
  method = c("nt", "aa"),
  junction = "junction",
  v_call = "v_call",
  j_call = "j_call",
  clone = "clone_id",
  cell_id = NULL,
  locus = "locus",
  only_heavy = TRUE,
  split_light = TRUE,
  first = FALSE,
  cdr3 = FALSE,
  mod3 = FALSE,
  max_n = 0,
  nproc = 1,
  verbose = FALSE,
  log = NULL,
  summarize_clones = TRUE
)
```

**Arguments**

- `db` data.frame containing sequence data.
- `method` one of the "nt" for nucleotide based clustering or "aa" for amino acid based clustering.
- `junction` character name of the column containing junction sequences. Also used to determine sequence length for grouping.
- `v_call` character name of the column containing the V-segment allele calls.
- `j_call` character name of the column containing the J-segment allele calls.
- `clone` character name of the column containing the clonal clustering identifiers.
- `cell_id` name of the column containing cell identifiers or barcodes. If specified, grouping will be performed in single-cell mode with the behavior governed by the `locus` and `only_heavy` arguments. If set to NULL then the bulk sequencing data is assumed.
- `locus` name of the column containing locus information. Only applicable to single-cell data. Ignored if `cell_id=NULL`. 


**identicalClones**

- **only_heavy** use only the IGH (BCR) or TRB/TRD (TCR) sequences for grouping. Only applicable to single-cell data. Ignored if cell_id=NULL.
- **split_light** split clones by light chains. Ignored if cell_id=NULL.
- **first** specifies how to handle multiple V(D)J assignments for initial grouping. If TRUE only the first call of the gene assignments is used. If FALSE the union of ambiguous gene assignments is used to group all sequences with any overlapping gene calls.
- **cdr3** if TRUE removes 3 nucleotides from both ends of "junction" prior to clustering (converts IMGT junction to CDR3 region). If TRUE this will also remove records with a junction length less than 7 nucleotides.
- **mod3** if TRUE removes records with a junction length that is not divisible by 3 in nucleotide space.
- **max_n** The maximum number of N’s to permit in the junction sequence before excluding the record from clonal assignment. Default is set to be zero. Set it as "NULL" for no action.
- **nproc** number of cores to distribute the function over.
- **verbose** if TRUE prints out a summary of each step cloning process. if FALSE (default) process cloning silently.
- **log** output path and filename to save the verbose log. The input file directory is used if path is not specified. The default is NULL for no action.
- **summarize_clones** if TRUE performs a series of analysis to assess the clonal landscape and returns a ScoperClones object. If FALSE then a modified input db is returned.

**Value**

If summarize_clones=TRUE (default) a ScoperClones object is returned that includes the clonal assignment summary information and a modified input db in the db slot that contains clonal identifiers in the specified clone column. If summarize_clones=FALSE modified data.frame is returned with clone identifiers in the specified clone column.

**Single-cell data**

To invoke single-cell mode the cell_id argument must be specified and the locus column must be correct. Otherwise, clustering will be performed with bulk sequencing assumptions, using all input sequences regardless of the values in the locus column.

Values in the locus column must be one of c("IGH", "IGI", "IGK", "IGL") for BCR or c("TRA", "TRB", "TRD", "TRG") for TCR sequences. Otherwise, the operation will exit and return an error message.

Under single-cell mode with paired-chain sequences, there is a choice of whether grouping should be done by (a) using IGH (BCR) or TRB/TRD (TCR) sequences only or (b) using IGH plus IGK/IGL (BCR) or TRB/TRD plus TRA/TRG (TCR) sequences. This is governed by the only_heavy argument. There is also choice as to whether inferred clones should be split by the light/short chain (IGK, IGL, TRA, TRG) following heavy/long chain clustering, which is governed by the split_light argument.

In single-cell mode, clonal clustering will not be performed on data were cells are assigned multiple heavy/long chain sequences (IGH, TRB, TRD). If observed, the operation will exit and return an...
error message. Cells that lack a heavy/long chain sequence (i.e., cells with light/short chains only) will be assigned a clone_id of NA.

See Also

See plotCloneSummary for plotting summary results. See groupGenes for more details about grouping requirements.

Examples

# Find clonal groups
results <- identicalClones(ExampleDb)

# Retrieve modified input data with clonal clustering identifiers
df <- as.data.frame(results)

# Plot clonal summaries
plot(results, binwidth=0.02)
**xmax** maximum limit for plotting the x-axis. If NULL the limit will be set automatically.

**breaks** number of breaks to show on the x-axis. If NULL the breaks will be set automatically.

**binwidth** binwidth for the histogram. If NULL the binwidth will be set automatically.

**title** string defining the plot title.

**size** numeric value for lines in the plot.

**silent** if TRUE do not draw the plot and just return the ggplot2 object; if FALSE draw the plot.

... additional arguments to pass to ggplot2::theme.

**Value**

A ggplot object defining the plot.

**See Also**

See ScoperClones for the the input object definition. See spectralClones, identicalClones and hierarchicalClones for generating the input object.

**Examples**

```r
# Find clones
results <- hierarchicalClones(ExampleDb, threshold=0.15)

# Plot clonal summaries
plot(results, binwidth=0.02)
```

---

**scoper**  
*The SCOPer package*

**Description**

scoper is a member of the Immcantation framework and provides computational approaches for the identification of B cell clones from adaptive immune receptor repertoire sequencing (AIRR-Seq) datasets. It includes methods for assigning clonal identifiers using sequence identity, hierarchical clustering, and spectral clustering.

**Clonal clustering**

- **identicalClones**: Clonal assignment using sequence identity partitioning.
- **hierarchicalClones**: Hierarchical clustering approach to clonal assignment.
- **spectralClones**: Spectral clustering approach to clonal assignment.

**Visualization**

- **plotCloneSummary**: Visualize inter- and intra-clone distances.
References


ScoperClones-class

S4 class containing clonal assignments and summary data

Description

ScoperClones stores output from identicalClones, hierarchicalClones and spectralClones functions.

Usage

```r
## S4 method for signature 'ScoperClones'
print(x)
## S4 method for signature 'ScoperClones'
summary(object)
## S4 method for signature 'ScoperClones,missing'
plot(x, y, ...)
## S4 method for signature 'ScoperClones'
as.data.frame(x)
```

Arguments

- `x` ScoperClones object
- `object` ScoperClones object
- `y` ignored.
- `...` arguments to pass to plotCloneSummary.

Slots

db data.frame of repertoire data including with clonal identifiers in the column specified during processing.

vjl_groups data.frame of clonal summary, including sequence count, V gene, J gene, junction length, and clone counts.

ter_inter data.frame containing minimum inter (between) and maximum intra (within) clonal distances.

eff_threshold effective cut-off separating the inter (between) and intra (within) clonal distances.
spectralClones

See Also

identicalClones, hierarchicalClones and spectralClones

Description

spectralClones provides an unsupervised spectral clustering approach to infer clonal relationships in high-throughput Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) data. This approach clusters B or T cell receptor sequences based on junction region sequence similarity and shared mutations within partitions that share the same V gene, J gene, and junction length, allowing for ambiguous V or J gene annotations.

Usage

spectralClones(
  db,
  method = c("novj", "vj"),
  germline = "germline_alignment",
  sequence = "sequence_alignment",
  junction = "junction",
  v_call = "v_call",
  j_call = "j_call",
  clone = "clone_id",
  cell_id = NULL,
  locus = "locus",
  only_heavy = TRUE,
  split_light = TRUE,
  targeting_model = NULL,
  len_limit = NULL,
  first = FALSE,
  cdr3 = FALSE,
  mod3 = FALSE,
  max_n = 0,
  threshold = NULL,
  base_sim = 0.95,
  iter_max = 1000,
  nstart = 1000,
  nproc = 1,
  verbose = FALSE,
  log = NULL,
  summarize_clones = TRUE
)
spectralClones

Arguments

- **db**: data.frame containing sequence data.
- **method**: one of the "novj" or "vj". See Details for description.
- **germline**: character name of the column containing the germline or reference sequence.
- **sequence**: character name of the column containing input sequences.
- **junction**: character name of the column containing junction sequences. Also used to determine sequence length for grouping.
- **v_call**: character name of the column containing the V-segment allele calls.
- **j_call**: character name of the column containing the J-segment allele calls.
- **clone**: the output column name containing the clone ids.
- **cell_id**: name of the column containing cell identifiers or barcodes. If specified, grouping will be performed in single-cell mode with the behavior governed by the **locus** and **only_heavy** arguments. If set to NULL then the bulk sequencing data is assumed.
- **locus**: name of the column containing locus information. Only applicable to single-cell data. Ignored if cell_id=NULL.
- **only_heavy**: use only the IGH (BCR) or TRB/TRD (TCR) sequences for grouping. Only applicable to single-cell data. Ignored if cell_id=NULL.
- **split_light**: split clones by light chains. Ignored if cell_id=NULL.
- **targeting_model**: TargetingModel object. Only applicable if method="vj". See Details for description.
- **len_limit**: IMGT_V object defining the regions and boundaries of the Ig sequences. If NULL, mutations are counted for entire sequence. Only applicable if method = "vj".
- **first**: specifies how to handle multiple V(D)J assignments for initial grouping. If TRUE only the first call of the gene assignments is used. If FALSE the union of ambiguous gene assignments is used to group all sequences with any overlapping gene calls.
- **cdr3**: if TRUE removes 3 nucleotides from both ends of "junction" prior to clustering (converts IMGT junction to CDR3 region). If TRUE this will also remove records with a junction length less than 7 nucleotides.
- **mod3**: if TRUE removes records with a junction length that is not divisible by 3 in nucleotide space.
- **max_n**: the maximum number of N's to permit in the junction sequence before excluding the record from clonal assignment. Default is set to be zero. Set it as "NULL" for no action.
- **threshold**: the supervising cut-off to enforce an upper-limit distance for clonal grouping. A numeric value between (0,1).
- **base_sim**: required similarity cut-off for sequences in equal distances from each other.
- **iter_max**: the maximum number of iterations allowed for kmean clustering step.
- **nstart**: the number of random sets chosen for kmean clustering initialization.
spectralClones

**nproc**
number of cores to distribute the function over.

**verbose**
if TRUE prints out a summary of each step cloning process. if FALSE (default) process cloning silently.

**log**
output path and filename to save the verbose log. The input file directory is used if path is not specified. The default is NULL for no action.

**summarize_clones**
if TRUE performs a series of analysis to assess the clonal landscape and returns a `ScoperClones` object. If FALSE then a modified input db is returned.

**Details**

If method="novj", then clonal relationships are inferred using an adaptive threshold that indicates the level of similarity among junction sequences in a local neighborhood.

If method="vj", then clonal relationships are inferred not only on junction region homology, but also taking into account the mutation profiles in the V and J segments. Mutation counts are determined by comparing the input sequences (in the column specified by `sequence`) to the effective germline sequence (IUPAC representation of sequences in the column specified by `germline`).

While not mandatory, the influence of SHM hot-/cold-spot biases in the clonal inference process will be noted if a SHM targeting model is provided through the `targeting_model` argument. See `TargetingModel` for more technical details.

If the `threshold` argument is specified, then an upper limit for clonal grouping will be imposed to prevent sequences with dissimilarity above the threshold from grouping together. Any sequence with a distance greater than the `threshold` value from the other sequences, will be assigned to a singleton group.

**Value**

If `summarize_clones=TRUE` (default) a `ScoperClones` object is returned that includes the clonal assignment summary information and a modified input db in the db slot that contains clonal identifiers in the specified clone column. If `summarize_clones=FALSE` modified data.frame is returned with clone identifiers in the specified clone column.

**Single-cell data**

To invoke single-cell mode the `cell_id` argument must be specified and the `locus` column must be correct. Otherwise, clustering will be performed with bulk sequencing assumptions, using all input sequences regardless of the values in the `locus` column.

Values in the `locus` column must be one of c("IGH", "IGI", "IGK", "IGL") for BCR or c("TRA", "TRB", "TRD", "TRG") for TCR sequences. Otherwise, the operation will exit and return an error message.

Under single-cell mode with paired-chain sequences, there is a choice of whether grouping should be done by (a) using IGH (BCR) or TRB/TRD (TCR) sequences only or (b) using IGH plus IGK/IGL (BCR) or TRB/TRD plus TRA/TRG (TCR) sequences. This is governed by the `only_heavy` argument. There is also choice as to whether inferred clones should be split by the light/short chain (IGK, IGL, TRA, TRG) following heavy/long chain clustering, which is governed by the `split_light` argument.

In single-cell mode, clonal clustering will not be performed on data were cells are assigned multiple heavy/long chain sequences (IGH, TRB, TRD). If observed, the operation will exit and return an
error message. Cells that lack a heavy/long chain sequence (i.e., cells with light/short chains only) will be assigned a clone_id of NA.

**See Also**

See `plotCloneSummary` for plotting summary results. See `groupGenes` for more details about grouping requirements.

**Examples**

```r
# Subset example data
db <- subset(ExampleDb, sample_id == "-1h")

# Find clonal groups
results <- spectralClones(db, method="novj", germline="germline_alignment_d_mask")

# Retrieve modified input data with clonal clustering identifiers
df <- as.data.frame(results)

# Plot clonal summaries
plot(results, binwidth=0.02)
```
Index

* datasets
  ExampleDb, 2

as.data.frame, ScoperClones-method
  (ScoperClones-class), 10

ExampleDb, 2

groupGenes, 5, 8, 14

hierarchicalClones, 3, 8–11

identicalClones, 5, 8–11

IMGT_V, 12

plot, ScoperClones, missing-method
  (ScoperClones-class), 10

plotCloneSummary, 5, 8, 9, 10, 14

print, ScoperClones-method
  (ScoperClones-class), 10

scoper, 9

ScoperClones, 4, 5, 7–9, 13

ScoperClones (ScoperClones-class), 10

ScoperClones-class, 10

ScoperClones-method
  (ScoperClones-class), 10

spectralClones, 8–11, 11

summary, ScoperClones-method
  (ScoperClones-class), 10

TargetingModel, 12, 13