Package ‘scoper’

May 25, 2020

Type Package

Version 1.0.1

Date 2020-05-24

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.
Nouri N and Kleinstein SH (2018) <doi: 10.1093/bioinformatics/bty235>.
Nouri N and Kleinstein SH (2019) <doi: 10.1101/788620>.
Gupta NT, et al. (2017) <doi: 10.4049/jimmunol.1601850>.

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 (>= 1.0.1), shazam (>= 1.0.0), data.table,
doParallel, dplyr (>= 0.8.1), foreach, magrittr, methods, Rcpp
(>= 0.12.12), rlang, scales, stats, stringi

LinkingTo Rcpp

Suggests knitr, rmarkdown, testthat

RoxygenNote 7.1.0

Collate 'Data.R' 'Scoper.R' 'Functions.R' 'RcppExports.R'

NeedsCompilation yes

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 2020-05-25 19:40:02 UTC

R topics documented:

`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`: TIgGER 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.

References

1. Laserson U and Vigneault F, et al. High-resolution antibody dynamics of vaccine-induced immune responses. Proc Natl Acad Sci USA. 2014 111:4928-33.

`hierarchicalClones` *Hierarchical clustering-based method for partitioning Ig sequences into clones.*

Description

The `hierarchicalClones` function provides a computational pipline for assigning Ig sequences into clonal groups sharing same V gene, J gene, and junction length, based on the junction sequence similarity.

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",
  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` 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 clonal cluster identifiers.

`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.

Details

`hierarchicalClones` provides a computational platform to explore the B cell clonal relationships in high-throughput Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) data sets. This function performs hierarchical clustering among sequences of B cell receptors (BCRs, immunoglobulins, Ig) that share the same V gene, J gene, and junction length based on the junction sequence similarity:

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.

See Also

See plotCloneSummary plotting summary results.

Examples

```
# 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 *Identical clustering-based method for partitioning Ig sequences into clones.*

Description

The identicalClones function provides a computational pipline for assigning Ig sequences into clonal groups sharing same V gene, J gene, and identical junction.

Usage

```
identicalClones(
  db,
  method = c("nt", "aa"),
  junction = "junction",
  v_call = "v_call",
  j_call = "j_call",
  clone = "clone_id",
  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` the output column name containing the clonal clustering identifiers.

`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.

Details

`identicalClones` provides a computational platform to explore the B cell clonal relationships in high-throughput Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) data sets. This function performs clustering among sequences of B cell receptors (BCRs, immunoglobulins, Ig) that share the same V gene, J gene, and identical junction:

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.

See Also

See plotCloneSummary plotting summary results.

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)
```

plotCloneSummary *Plot clonal clustering summary*

Description

`plotCloneSummary` plots the results in a `ScoperClones` object returned by `spectralClones`, `identicalClones` or `hierarchicalClones`. Includes the minimum inter (between) and maximum intra (within) clonal distances and the calculated efective threshold.

Usage

```
plotCloneSummary(
  data,
  xmin = NULL,
  xmax = NULL,
  breaks = NULL,
  binwidth = NULL,
  title = NULL,
  size = 0.75,
  silent = FALSE,
  ...
)
```

Arguments

`data`	ScoperClones object output by the spectralClones, identicalClones or hierarchicalClones.
`xmin`	minimum limit for plotting the x-axis. If `NULL` the limit will be set automatically.
`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

```
# 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

1. Nouri N and Kleinstein SH (2018). A spectral clustering-based method for identifying clones from high-throughput B cell repertoire sequencing data. Bioinformatics, 34(13):i341-i349.
2. Nouri N and Kleinstein SH (2019). Somatic hypermutation analysis for improved identification of B cell clonal families from next-generation sequencing data. bioRxiv, 10.1101/788620.
3. Gupta NT, et al. (2017). Hierarchical clustering can identify B cell clones with high confidence in Ig repertoire sequencing data. The Journal of Immunology, 198(6):2489-2499.

ScoperClones-class — *S4 class containing clonal assignments and summary data*

Description

ScoperClones stores output from identicalClones, hierarchicalClones and spectralClones functions.

Usage

```
## 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.
- inter_intra 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.

See Also

identicalClones, hierarchicalClones and spectralClones

spectralClones *Spectral clustering-based method for partitioning Ig sequences into clones.*

Description

The `spectralClones` function provides an unsupervised computational pipline for assigning Ig sequences into clonal groups sharing same V gene, J gene, and junction length, based on the junction sequence similarity and shared mutations in V and J segments.

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",
  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
)
```

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.

`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.

`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

`spectralClones` provides a computational platform to explore the B cell clonal relationships in high-throughput Adaptive Immune Receptor Repertoire sequencing (AIRR-seq) data sets. Two methods are included to perform clustering among sequences of B cell receptors (BCRs, immunoglobulins, Ig) that share the same V gene, J gene and junction length:

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

- If method = "vj": clonal relationships are inferred not only based on the junction region homology, but also takes 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).
- Not mandatory, but the influence of SHM hot- and cold-spot biases in the clonal inference process will be noted if a SHM targeting model is provided through argument targeting_model (see createTargetingModel for more technical details).
- Not mandatory, but the upper-limit cut-off for clonal grouping can be provided to prevent sequences with disimilarity above the threshold group together. Using this argument any sequence with distances above the threshold value from other sequences, will become a singleton.

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.

See Also

See plotCloneSummary plotting summary results.

Examples

```
# 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