Package 'scoper'

May 9, 2026

Type Package

Version 1.5.0

Date 2026-05-05

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://github.com/immcantation/scoper/issues

LazyData true

BuildVignettes true

VignetteBuilder knitr

Encoding UTF-8

LinkingTo Rcpp

Depends R (>= 4.0), ggplot2 (>= 3.4.0)

Imports alakazam (>= 1.4.1), shazam (>= 1.3.0), data.table, doParallel, dplyr (>= 1.0), fastcluster, foreach, methods, Rcpp (>= 0.12.12), rlang, scales, stats, stringi, tidyr (>= 1.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.3.3

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

NeedsCompilation yes

Author Nima Nouri [aut],
Edel Aron [ctb],
Robert Bjornson [ctb],
Gisela Gabernet [ctb],
Cole Jensen [ctb],
Huimin Lyu [ctb],
Susanna Marquez [ctb, cre],
Jason Vander Heiden [aut],
Steven Kleinstein [aut, cph]

Maintainer Susanna Marquez <susanna.marquez@yale.edu>

Repository CRAN

Date/Publication 2026-05-05 15:04:04 UTC

Contents

ExampleDb 2
hierarchicalClones 3
identicalClones 7
plotCloneSummary 10
scoper 11
ScoperClones-class 12
spectralClones 13

Index **17**

`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.
- np1_length: Number of nucleotides between V and D segments
- np2_length: Number of nucleotides between D and J segments
- sample_id: Sample identifier
- c_call: C region assignment.
- duplicate_count: Copy number of the sequence
- locus: Locus of the receptor

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 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"),
  IUPAC = FALSE,
  junction = "junction",
  v_call = "v_call",
  j_call = "j_call",
  clone = "clone_id",
  fields = NULL,
  cell_id = NULL,
  locus = "locus",
```

```
  only_heavy = TRUE,
  split_light = FALSE,
  first = FALSE,
  cdr3 = FALSE,
  mod3 = FALSE,
  max_n = 0,
  nproc = 1,
  verbose = FALSE,
  log = NULL,
  summarize_clones = FALSE,
  seq_id = "sequence_id"
)
```

Arguments

db	data.frame containing sequence data.
threshold	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. Method ‘"aa"‘ still expects nucleotide sequences, which will be translated to amino acids
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.
IUPAC	If TRUE, allows sequences with IUPAC codes to pass validation and be used in clustering with IUPAC-aware distance calculation (via alakazam::pairwiseDist). If FALSE (default), uses fast Hamming distance (via fastDist_rcpp) and only allows standard bases (A, T, C, G), N, and ? in sequences. This parameter controls validation and distance calculation method, not sequence filtering. See max_n for filtering sequences by character content. See the IUPAC and max_n parameters section for more details and examples. Note: This parameter is only available for hierarchicalClones with method="nt".
junction	character name of the column containing junction sequences. Also used to determine sequence length for grouping.
v_call	name of the column containing the V-segment allele calls.
j_call	name of the column containing the J-segment allele calls.
clone	output column name containing the clonal cluster identifiers.
fields	character vector of additional columns to use for grouping. Sequences with disjoint values in the specified fields will be classified as separate clones.
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 — This is deprecated. Only heavy chains will be used in clustering. 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 — This is deprecated. If you desire to split clones by light chains use dowser::resolveLightChains.

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 non-ATCG characters (degenerate positions) to permit in the junction sequence before excluding the record from clonal assignment. Note: max_n operates independently from IUPAC - it controls filtering by character count, while IUPAC controls validation and distance calculation method. With linkage="single", non-informative positions can create artifactual links between unrelated sequences. Use with caution. Default is 0 (ATCG-only). Set to NULL for no filtering.

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 (default) then a modified input db is returned with clone identifiers in the specified 'clone' column. When grouping by fields, summarize_clones should be FALSE.

seq_id — The column containing sequence ids

Value

If summarize_clones=FALSE (default) a modified data.frame is returned with clone identifiers in the specified clone column. If summarize_clones=TRUE 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.

IUPAC and max_n parameters

Note: The IUPAC parameter is only available for hierarchicalClones with method="nt" (nucleotide mode). It is ignored when method="aa" (amino acid mode). The max_n parameter is available for all cloning functions.

The IUPAC and max_n parameters serve complementary but distinct purposes:

IUPAC controls:

- Sequence validation (which characters are allowed)
- Distance calculation method (fast Hamming vs IUPAC-aware scoring)

`max_n` controls:

- Sequence filtering by counting non-ATCG characters in the junction

`hierarchicalClones` with `method="aa"` accepts the full IUPAC DNA alphabet during validation, then `max_n` controls filtering of sequences containing excess non-ATCG characters before translating to amino acids and performing IUPAC-aware clustering.

Example use cases for `hierarchicalClones` with `method="nt"`:

- `IUPAC=FALSE, max_n=0`: Strict ATCG-only mode with fast distance calculation. Will throw an error and exit if sequences with characters not A, T, C, G, N, or ? are detected. max_n=0 will filter out sequences with N or ? characters. Fastest option for high-quality data.
- `IUPAC=FALSE, max_n>0`: Will throw an error and exit if sequences with characters not A, T, C, G, N, or ? are detected. Allows sequences with limited N/? characters in distance calculation, using fast Hamming distance. Note: IUPAC codes are rejected during validation (before max_n filtering), so max_n only controls filtering of sequences with N or ? characters. Useful for data with low-quality or masked positions but no experimental ambiguity codes.
- `IUPAC=TRUE, max_n=0`: Uses IUPAC-aware distance but filters out all non-ATCG characters anyway. Only standard bases remain after filtering. Slower than IUPAC=FALSE but handles any ambiguity codes in the input by filtering them out before clustering.
- `IUPAC=TRUE, max_n>0`: Allows sequences with limited ambiguity codes and uses proper IUPAC-aware distance calculation. Slower but handles biological ambiguity correctly. Set max_n to the maximum number of ambiguous positions per sequence you want to tolerate (counts all non-ATCG: N, ?, and other IUPAC codes).
- `IUPAC=TRUE, max_n=NULL`: Process all sequences with IUPAC codes regardless of the number of ambiguous positions. Uses IUPAC-aware distance calculation with no filtering. Most permissive option.

Note: Validation occurs before filtering. When `IUPAC=FALSE`, sequences containing IUPAC ambiguity codes (R, Y, W, S, M, K, etc.) will fail validation and be rejected before the `max_n` filtering step. Therefore, with `IUPAC=FALSE, max_n > 0`, only sequences with N and ? characters (not IUPAC codes) can pass validation and be filtered by `max_n`. The `max_n` parameter always counts using regex `"[^ATCG]"`, but `IUPAC` determines which non-ATCG characters are allowed to reach the filtering step.

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 where 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 <- hierarchicalClones(ExampleDb, threshold=0.15, summarize_clones=TRUE)

# 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

```
identicalClones(
  db,
  method = c("nt", "aa"),
  junction = "junction",
  v_call = "v_call",
  j_call = "j_call",
  clone = "clone_id",
  fields = NULL,
  cell_id = NULL,
```

```
  locus = "locus",
  only_heavy = TRUE,
  split_light = FALSE,
  first = FALSE,
  cdr3 = FALSE,
  mod3 = FALSE,
  max_n = 0,
  nproc = 1,
  verbose = FALSE,
  log = NULL,
  summarize_clones = FALSE
)
```

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	name of the column containing the V-segment allele calls.
j_call	name of the column containing the J-segment allele calls.
clone	output column name containing the clonal cluster identifiers.
fields	character vector of additional columns to use for grouping. Sequences with disjoint values in the specified fields will be classified as separate clones.
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	This is deprecated. Only heavy chains will be used in clustering. 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	This is deprecated. If you desire to split clones by light chains use dowser::resolveLightChains.
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 non-ATCG characters to permit in the junction sequence before excluding the record from clonal assignment. Counts non-ATCG characters using regex `"[^ATCG]"`, which includes N, ?, and IUPAC ambiguity codes. With the default value of 0, all sequences containing any non-ATCG character (including IUPAC codes) in the junction are removed before clustering. Set to `NULL` for no filtering.

`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` (default) then a modified input `db` is returned. When grouping by `fields`, `summarize_clones` should be `FALSE`.

Value

If `summarize_clones=FALSE` (default) a modified `data.frame` is returned with clone identifiers in the specified `clone` column. If `summarize_clones=TRUE` 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.

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 where 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, summarize_clones=TRUE)

# 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, summarize_clones=TRUE)

# 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 method for clonal partitioning*

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",
  fields = NULL,
  cell_id = NULL,
  locus = "locus",
  only_heavy = TRUE,
  split_light = FALSE,
  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 = FALSE
)
```

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	name of the column containing the V-segment allele calls.
j_call	name of the column containing the J-segment allele calls.
clone	output column name containing the clonal cluster identifiers.
fields	character vector of additional columns to use for grouping. Sequences with disjoint values in the specified fields will be classified as separate clones.
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	This is deprecated. Only heavy chains will be used in clustering. 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	This is deprecated. If you desire to split clones by light chains use dowser::resolveLightChains.
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 non-ATCG characters to permit in the junction sequence before excluding the record from clonal assignment. Counts non-ATCG characters using regex "[^ATCG]", which includes N, ?, and IUPAC ambiguity codes. With the default value of 0, all sequences containing any non-ATCG character are removed before clustering. Set to NULL for no filtering.

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 (default) then a modified input db is returned. When grouping by fields, summarize_clones should be FALSE.

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=FALSE (default) a modified data.frame is returned with clone identifiers in the specified clone column. If summarize_clones=TRUE 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.

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

```
# Subset example data
db <- subset(ExampleDb, c_call == "IGHG")

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

# 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*)*, 12

ExampleDb, 2

groupGenes, *7*, *9*, *16*

hierarchicalClones, 3, *10–13*

identicalClones, 7, *10–13*
IMGT_V, *14*

plot,ScoperClones,missing-method
*(*ScoperClones-class*)*, 12
plotCloneSummary, *7*, *9*, 10, *11*, *12*, *16*
print,ScoperClones-method
*(*ScoperClones-class*)*, 12

scoper, 11
ScoperClones, *5*, *9–11*, *15*
ScoperClones *(*ScoperClones-class*)*, 12
ScoperClones-class, 12
ScoperClones-method
*(*ScoperClones-class*)*, 12
spectralClones, *10–13*, 13
summary,ScoperClones-method
*(*ScoperClones-class*)*, 12

TargetingModel, *14*, *15*