Package 'piglet'

February 17, 2026

Title Program for Inferring Immunoglobulin Allele Similarity Clusters and Genotypes

Version 1.2.0

Author Ayelet Peres [aut, cre],
William Lees [aut],
Gur Yaari [aut, cph]

Maintainer Ayelet Peres <ayelet.peres@yale.edu>

Description Improves genotype inference and downstream Adaptive Immune Receptor Repertoire Sequence data analysis. Inference of allele similarity clusters, an alternative naming scheme and genotype inference for immunoglobulin heavy chain repertoires.
The main tools are allele similarity clusters, and allele based genotype. The first tool is designed to reduce the ambiguity within the immunoglobulin heavy chain V alleles. The ambiguity is caused by duplicated or similar alleles which are shared among different genes.
The second tool is an allele based genotype, that determined the presence of an allele based on a threshold derived from a naive population. See Peres et al. (2023) <doi:10.1093/nar/gkad603>.

License CC BY-SA 4.0

Encoding UTF-8

Depends R (>= 3.5.0)

LinkingTo Rcpp

SystemRequirements GNU make

Imports Biostrings (>= 2.62.0), DECIPHER (>= 2.22.0), alakazam (>= 1.2.0), dendextend (>= 1.9.0), data.table (>= 1.12.2), tigger (>= 1.0.0), methods (>= 3.4.4), rlang (>= 0.4.0), zen4R (>= 0.7), RColorBrewer (>= 1.1.2), ggplot2 (>= 3.3.6), circlize (>= 0.4.15), R6 (>= 2.5.1), jsonlite (>= 1.8.3), Rcpp (>= 0.11.0), magrittr, igraph (>= 1.3.0), stringdist (>= 0.9.0), cluster (>= 2.1.0), ape (>= 5.0)

Suggests knitr, rmarkdown, tidyr, htmltools, stringi, bookdown, ComplexHeatmap, dplyr, ggtree (>= 3.0.0), testthat (>= 3.0.0), parallel

RoxygenNote 7.3.3

Collate 'Data.R' 'GermlineCluster-class.R' 'RcppExports.R' 'piglet.R' 'allele_cluster.R' 'utils.R' 'allele_genotype.R' 'community_detection.R' 'piglet-package.R' 'utils-pipe.R' 'visualization.R'

LazyData true

BuildVignettes true

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-02-17 22:30:02 UTC

Contents

`alleleClusterNames` *Allele similarity cluster naming scheme*

Description

For a given cluster the function collapse similar sequences and renames the sequences based on the ASC name scheme

Usage

```
alleleClusterNames(cluster, allele.cluster.table, germ.dist, chain, segment)
```

Arguments

`cluster`	A vector with the cluster identifier - the family and allele cluster number.
`allele.cluster.table`	A data.frame with the list of all germline sequences and their clusters.
`germ.dist`	A matrix with the germline distance between the germline set sequences.
`chain`	A character with the chain identifier: IGH/IGL/IGK/TRB/TRA... (Currently only IGH is supported)
`segment`	A character with the segment identifier: IGHV/IGHD/IGHJ.... (Currently only IGHV is supported)

Value

A data.frame with the clusters renamed alleles based on the ASC scheme.

allele_cluster_table *Allele similarity cluster table*

Description

A `data.table` of the allele similarity cluster table based on the `HVGERM` and `hv_functionality` germlie reference set. This is not the latest version of the allele similarity cluster table. For the latest version please refer either to the zenodo doi or you can use the `recentAlleleClusters`

Usage

```
allele_cluster_table
```

Format

An object of class `data.table` (inherits from `data.frame`) with 286 rows and 5 columns.

References

Peres, et al (2022) doi:10.1101/2022.12.26.521922

allele_diff *Alleles nucleotide position difference*

Description

Compare the sequences of two alleles (reference and sample alleles) and returns the differential nucleotide positions of the sample allele.

Usage

```
allele_diff(
  reference_allele,
  sample_allele,
  position_threshold = 0,
  snps = TRUE
)
```

Arguments

`reference_allele`	The nucleotide sequence of the reference allele, character object.
`sample_allele`	The nucleotide sequence of the sample allele, character object.
`position_threshold`	A position from which to check for differential positions. If zero checks all position. Default to zero.
`snps`	If to return the SNP with the position (e.g., A2G where A is for the reference and G is for the sample.). If false returns just the positions. Default to True

Details

The function utilizes c++ script to optimize the run time for large comparisons.

Value

A `character` vector of the differential nucleotide positions of the sample allele.

Examples

```
{
reference_allele = "AAGG"
sample_allele = "ATGA"

# setting position_threshold = 0 will return all differences
diff <- allele_diff(reference_allele, sample_allele)
# "A2T", "G4A"
print(diff)

# setting position_threshold = 3 will return the differences from position three onward
diff <- allele_diff(reference_allele, sample_allele, position_threshold = 3)
# "G4A"
print(diff)

# setting snps = FALSE will return the differences as indices
diff <- allele_diff(reference_allele, sample_allele, snps = FALSE)
# 2, 4
print(diff)

}
```

`allele_diff_indices` *Calculate differences between characters in columns of germs and return their indices as an int vector.*

Description

Calculate differences between characters in columns of germs and return their indices as an int vector.

Usage

```
allele_diff_indices(germs, X = 0L, non_mismatch_chars_nullable = NULL)
```

Arguments

`germs`	A vector of strings representing germ sequences.
`X`	The threshold index from which to return differences as indices.
`non_mismatch_chars_nullable`	A set of characters that are ignored when comparing sequences (default: 'N', '.', '-').

Value

A vector of integers containing indices of differing columns.

Examples

```
germs = c("ATCG", "ATCC")
X = 3
result = allele_diff_indices(germs, X)
# 1, 2, 3
```

`allele_diff_indices_parallel` *Calculate SNPs or their count for each germline-input sequence pair with optional parallel execution.*

Description

Calculate SNPs or their count for each germline-input sequence pair with optional parallel execution.

Usage

```
allele_diff_indices_parallel(
  germs,
  inputs,
  X = 0L,
  parallel = FALSE,
  return_count = FALSE
)
```

Arguments

`germs`	A vector of strings representing germline sequences.
`inputs`	A vector of strings representing input sequences.
`X`	The threshold index from which to return SNP indices or counts (default: 0).
`parallel`	A boolean flag to enable parallel processing (default: FALSE).
`return_count`	A boolean flag to return the count of mutations instead of their indices (default: FALSE).

Value

A list of integer vectors (if return_count = FALSE) or a vector of integers (if return_count = TRUE).

`allele_diff_indices_parallel2` *Calculate SNPs or their count for each germline-input sequence pair with optional parallel execution.*

Description

This function compares germline sequences (`germs`) and input sequences (`inputs`) and identifies single nucleotide polymorphisms (SNPs) or their counts, with optional parallel execution. The comparison ignores specified non-mismatch characters (e.g., gaps or ambiguous bases).

Usage

```
allele_diff_indices_parallel2(
  germs,
  inputs,
  X = 0L,
  parallel = FALSE,
  return_count = FALSE,
  non_mismatch_chars_nullable = NULL
)
```

Arguments

`germs`	A vector of strings representing germline sequences.
`inputs`	A vector of strings representing input sequences.
`X`	The threshold index from which to return SNP indices or counts (default: 0).
`parallel`	A boolean flag to enable parallel processing (default: FALSE).
`return_count`	A boolean flag to return the count of mutations instead of their indices (default: FALSE).
`non_mismatch_chars_nullable`	A set of characters that are ignored when comparing sequences (default: 'N', '.', '-').

Value

A list of integer vectors (if `return_count = FALSE`) or a vector of integers (if `return_count = TRUE`).

Examples

```
# Example usage
germs <- c("ATCG", "ATCC")
inputs <- c("ATTG", "ATTA")
X <- 0
```

```
# Return indices of SNPs
result_indices <- allele_diff_indices_parallel2(germs, inputs, X,
parallel = TRUE, return_count = FALSE)
print(result_indices)  # list(c(4), c(3, 4))

# Return counts of SNPs
result_counts <- allele_diff_indices_parallel2(germs, inputs, X,
parallel = FALSE, return_count = TRUE)
print(result_counts)  # c(1, 2)
```

`allele_diff_strings` *Calculate differences between characters in columns of germs and return them as a string vector.*

Description

Calculate differences between characters in columns of germs and return them as a string vector.

Usage

```
allele_diff_strings(germs, X = 0L, non_mismatch_chars_nullable = NULL)
```

Arguments

`germs`	A vector of strings representing germ sequences.
`X`	The threshold index from which to return differences as strings.
`non_mismatch_chars_nullable`	A set of characters that are ignored when comparing sequences (default: 'N', '.', '-').

Value

A vector of strings containing differences between characters in columns.

Examples

```
germs = c("ATCG", "ATCC")
X = 3
result = allele_diff_strings(germs, X)
# "A2T", "T3C", "C2G"
```

allele_threshold_table *Allele thresholds table*

Description

A `data.table` of the allele thresholds table. The V alleles are based on the `HVGERM` and `hv_functionality` germline reference set. The D, and the J are based on the AIRR-C reference set (https://zenodo.org/records/10489725). The table contains these columns: allele - the IUIS allele name, asc_allele - the allele name based on allele similarity clusters (only for V), threshold = the genotype threshold for the alleles.

Usage

```
allele_threshold_table
```

Format

An object of class `data.table` (inherits from `data.frame`) with 262 rows and 4 columns.

References

Peres, et al (2022) doi:10.1101/2022.12.26.521922

artificialFRW1Germline *FWR1 artificial dataset generator*

Description

A function to artificially create an IGHV reference set with framework1 (FWR1) primers (see Details).

Usage

```
artificialFRW1Germline(
  germline_set,
  mask_primer = TRUE,
  trimm_primer = FALSE,
  quite = FALSE
)
```

Arguments

germline_set	A germline set distance matrix created by ighvDistance.
mask_primer	Logical (TRUE by default). If to mask with Ns the region of the primer from the germline sequence
trimm_primer	Logical (FALSE by default). If to trim the region of the primer from the germline sequence. If TRUE then, mask_primer is ignored.
quite	Logical (FALSE by default). Do you want to suppress informative messages

Details

The FRW1 primers used in this function were taken from the BIOMED-2 protocol. For more information on the protocol and primer design go to: van Dongen, J., Langerak, A., Brüggemann, M. et al. Design and standardization of PCR primers and protocols for detection of clonal immunoglobulin and T-cell receptor gene recombinations in suspect lymphoproliferations: Report of the BIOMED-2 Concerted Action BMH4-CT98-3936. Leukemia 17, 2257–2317 (2003). https://doi.org/10.1038/sj.leu.2403202Van Dongen, J. J. M., et al. "Design and standardization of PCR primers and protocols for detection of clonal immunoglobulin and T-cell receptor gene recombinations in suspect lymphoproliferations: report of the BIOMED-2 Concerted Action BMH4-CT98-3936." Leukemia 17.12 (2003): 2257-2317.

Value

A list with the input germline set allele and the trimmed/masked sequences.

---

assignAlleleClusters *Assign allele similarity clusters*

Description

assignAlleleClusters uses the allele clusters annotation to change the preliminary allele assignments to the new annotations before inferring a genotype.

Usage

```
assignAlleleClusters(
  data,
  alleleClusterTable,
  v_call = "v_call",
  from_col = "imgt_allele",
  to_col = "new_allele"
)
```

Arguments

data	data.frame in AIRR format, containing V allele calls from a single subject and the sample IMGT-gapped V(D)J sequences under seq.
alleleClusterTable	A data.frame of the allele clusters new annotations relative to the original reference set. See details.
v_call	name of the V allele call column. Default is v_call
from_col	name of the column in alleleClusterTable to use as the source for the dictionary. Default is imgt_allele
to_col	name of the column in alleleClusterTable to use as the target for the dictionary. Default is new_allele

Value

A modified input data.frame with the new assigned

Examples

```
# preferably obtain the latest ASC cluster table
# asc_archive <- recentAlleleClusters(doi="10.5281/zenodo.7429773", get_file = TRUE)

# allele_cluster_table <- extractASCTable(archive_file = asc_archive)

# example allele similarity cluster table
data(allele_cluster_table)

# loading TIgGER AIRR-seq b cell data
data <- tigger::AIRRDb

asc_data <- assignAlleleClusters(data, allele_cluster_table)
```

---

detect_communities_leiden

Leiden community detection

Description

Performs community detection on a weighted graph using the Leiden algorithm with CPM (Constant Potts Model) objective function.

Usage

```
detect_communities_leiden(g, resolution = 1)
```

Arguments

g	An igraph graph object with weighted edges
resolution	Resolution parameter for Leiden algorithm. Higher values produce more communities. Default is 1.0.

Details

The Leiden algorithm is a community detection method that optimizes a quality function (here CPM). It guarantees connected communities and is generally faster than Louvain while producing better quality partitions.

Value

An igraph communities object

See Also

distance_to_graph, optimize_resolution

Examples

```
data(HVGERM)
d <- igDistance(HVGERM[1:10], method = "hamming")
g <- distance_to_graph(d)
comm <- detect_communities_leiden(g, resolution = 0.5)
```

---

distance_to_graph *Convert distance matrix to weighted graph*

Description

Converts a distance matrix to a weighted igraph object using a log transform that spreads small distances and produces weights in [0,1].

Usage

```
distance_to_graph(distance_matrix)
```

Arguments

distance_matrix	A distance matrix or dist object

Details

The transformation uses a log-based similarity measure:

1. Normalize distances by max distance
2. Apply -log transform to convert to similarity
3. Normalize similarities to [0,1] range
4. Create weighted undirected graph

Value

An igraph object with weighted edges

See Also

detect_communities_leiden, igClust

Examples

```
data(HVGERM)
d <- igDistance(HVGERM[1:10], method = "hamming")
g <- distance_to_graph(d)
```

---

extractASCTable *Extracts the allele cluster table from the archive file.*

Description

Extracts the allele cluster table from the archive file.

Usage

```
extractASCTable(archive_file = NULL)
```

Arguments

archive_file	A path to the asc archive file. Default is null. (see details)

Details

For downloading the latest archive file with the updated allele cluster table, use the function `recentAlleleClusters`.

Value

Returns the allele cluster table.

The table columns: `new_allele` - the ASC given allele name `func_group` - the ASC cluster number `imgt_allele` - the original IUIS/IMGT allele name `thresh` - the allele threshold for ASC-based genotype inference `amplicon_length` - is the original length of the reference set.

Examples

```
asc_archive <- recentAlleleClusters(doi="10.5281/zenodo.7429773", get_file = TRUE)

allele_cluster_table <- extractASCTable(archive_file = asc_archive)
```

`generateReferenceSet` *Generate allele similarity reference set*

Description

Generates the allele clusters reference set based on the clustering from ighvClust. The function collapse similar alleles and assign them into their respective allele clusters and family clusters. See details for naming scheme

Usage

```
generateReferenceSet(
  germline_distance,
  germline_set,
  alleleClusterTable,
  trim_3prime_side = NULL
)
```

Arguments

`germline_distance`
: A germline set distance matrix created by ighvDistance.

`germline_set`
: A character list of the IMGT aligned IGHV allele sequences. See details for curating options.

`alleleClusterTable`
: A data.frame of the alleles and their clusters created by ighvClust.

`trim_3prime_side`
: If a 3' position trim is supplied, duplicated sequences will be checked for differential positions past the trim position. Default NULL; NULL will not activate the check. see @details

Details

Each allele is named by this scheme: IGHVF1-G1*01 - IGH = chain, V = region, F1 = family cluster numbering, G1 - allele cluster numbering, and 01 = allele numbering (given by clustering order, no connection to the expression)

In case there are alleles that are differentiated in a nucleotide position past the trimming position used for the clustering, then the alleles are separated and are annotated with the differentiating

position as so: Say A1*01 and A1*02 are similar up to position 318, and thus collapsed in the clusters to G1*01. Upon checking the sequences past the trim position (318), a differentiating nucleotide was seen in position 319, A1*01 has a G, and A1*02 has a T. Then the alleles will be separated, and the new annotation will be as so: A1*01 = G1*01, and A1*02 = G1*01_G319T. Where the first nucleotide indicate the base, the following number the position, and the last nucleotide the one the base changed into.

Value

A `list` with the re-named germline set, and a table of the allele clusters and thresholds.

`germlineASC` *Converts IGHV germline set to ASC germline set.*

Description

Converts IGHV germline set to ASC germline set.

Usage

```
germlineASC(allele_cluster_table, germline)
```

Arguments

`allele_cluster_table`	The allele cluster table.
`germline`	An IGHV germline set with matching names to the "imgt_allele" column in the allele_cluster_table.

Value

Returns the IGHV germline set with the ASC allele names.

Examples

```
# preferably obtain the latest ASC cluster table
# asc_archive <- recentAlleleClusters(doi="10.5281/zenodo.7429773", get_file = TRUE)

# allele_cluster_table <- extractASCTable(archive_file = asc_archive)

data(HVGERM)

# example allele similarity cluster table
data(allele_cluster_table)

asc_germline <- germlineASC(allele_cluster_table, germline = HVGERM)
```

HVGERM *Human IGHV germlines*

Description

A `character` vector of all 498 human IGHV germline gene segment alleles in IMGT Gene-db release July 2022, with an additional 25 undocumented alleles from VDJbase.

Usage

```
HVGERM
```

Format

Values correspond to IMGT-gaped nuceltoide sequences (with nucleotides capitalized and gaps represented by '.').

References

Xochelli *et al*. (2014) Immunoglobulin heavy variable (IGHV) genes and alleles: new entities, new names and implications for research and prognostication in chronic lymphocytic leukaemia. *Immunogenetics*. 67(1):61-6.

hv_functionality *Human IGHV germlines functionality description*

Description

A `data.table` of all 498 human IGHV germline gene segment alleles in IMGT Gene-db release July 2022, with an additional 25 undocumented alleles from VDJbase. The first column is the allele name, the second column is the functionality annotation, the third column is the nt sequence and the last column is the aa sequence.

Usage

```
hv_functionality
```

Format

An object of class `data.table` (inherits from `data.frame`) with 521 rows and 4 columns.

References

Xochelli *et al*. (2014) Immunoglobulin heavy variable (IGHV) genes and alleles: new entities, new names and implications for research and prognostication in chronic lymphocytic leukaemia. *Immunogenetics*. 67(1):61-6.

igClust *Allele similarity clustering*

Description

Cluster the distance matrix to create allele clusters. Supports both hierarchical clustering (default) and Leiden community detection.

Usage

```
igClust(
  germline_distance,
  method = c("hierarchical", "leiden"),
  family_threshold = 75,
  allele_cluster_threshold = 95,
  cluster_method = "complete",
  resolution = NULL,
  target_clusters = NULL,
  optimize_silhouette = TRUE,
  ncores = 1,
  quiet = FALSE
)
```

Arguments

germline_distance	A germline set distance matrix created by igDistance.
method	Clustering method. One of "hierarchical" (default) or "leiden".
family_threshold	The similarity threshold for family level (hierarchical only). Default is 75.
allele_cluster_threshold	The similarity threshold for allele cluster level (hierarchical only). Default is 95.
cluster_method	The hierarchical clustering linkage method. Default is "complete".
resolution	Resolution parameter for Leiden clustering. If NULL, will be optimized.
target_clusters	Target number of clusters for Leiden optimization. Default is NULL.
optimize_silhouette	Logical. Optimize resolution using silhouette score (Leiden only). Default is TRUE.
ncores	Number of cores for parallel processing (Leiden only). Default is 1.
quiet	Logical. Suppress messages. Default is FALSE.

Value

A named list that includes:

- `alleleClusterTable`: data.frame of allele clusters
- `threshold`: list of threshold parameters
- `hclustAlleleCluster`: hierarchical clustering object (hierarchical method)
- `communityObject`: community detection result (Leiden method)
- `graphObject`: igraph object (Leiden method)
- `silhouetteScore`: silhouette score (Leiden method)
- `resolutionParameter`: resolution used (Leiden method)

See Also

`igDistance`, `inferAlleleClusters`

---

igDistance *Germline set alleles distance*

Description

Calculates the distance between pairs of alleles based on their aligned germline sequences. Supports multiple distance methods for different segment types.

Usage

```
igDistance(
  germline_set,
  AA = FALSE,
  method = c("decipher", "hamming", "lv"),
  trim_3prime = NULL,
  return_type = c("matrix", "dist"),
  quiet = TRUE
)
```

Arguments

`germline_set`	A character vector of aligned allele sequences. See details for curating options.
`AA`	Logical (FALSE by default). If TRUE, calculate the distance based on amino acid sequences.
`method`	Distance calculation method. One of: • `"decipher"`: Uses DECIPHER::DistanceMatrix (requires aligned sequences, best for V segments) • `"hamming"`: Hamming distance (requires equal length, sequences padded if needed)

	• `"lv"`: Levenshtein distance (handles variable length, good for D/J segments)
`trim_3prime`	Optional position to trim sequences from 3' end before distance calculation
`return_type`	One of "matrix" (default) or "dist" to return a dist object
`quiet`	Logical (TRUE by default). Suppress informative messages

Details

The aligned IMGT IGHV allele germline set can be downloaded from the IMGT site https://www.imgt.org/ under the section genedb.

For V segments, the "decipher" method is recommended as it handles alignment gaps properly. For D and J segments which may have variable lengths, the "lv" (Levenshtein) method is appropriate.

Value

A `matrix` or `dist` object of the computed distances between allele pairs.

See Also

`ighvDistance` for backward compatibility wrapper

Examples

```
data(HVGERM)
# Using DECIPHER method (default, for V segments)
d1 <- igDistance(HVGERM[1:10], method = "decipher")

# Using Hamming distance
d2 <- igDistance(HVGERM[1:10], method = "hamming")

# Using Levenshtein distance (good for D/J segments)
d3 <- igDistance(HVGERM[1:10], method = "lv")
```

---

ighvClust *Allele similarity clustering (deprecated)*

Description

This function is deprecated. Use `igClust` instead.

Usage

```
ighvClust(
  germline_distance,
  family_threshold = 75,
  allele_cluster_threshold = 95,
  cluster_method = "complete"
)
```

Arguments

germline_distance	A germline set distance matrix created by igDistance.
family_threshold	The similarity threshold for family level (hierarchical only). Default is 75.
allele_cluster_threshold	The similarity threshold for allele cluster level (hierarchical only). Default is 95.
cluster_method	The hierarchical clustering linkage method. Default is "complete".

Value

A named list with clustering results.

See Also

igClust for the current implementation

---

ighvDistance *Germline set alleles distance (deprecated)*

Description

This function is deprecated. Use igDistance instead.

Usage

```
ighvDistance(germline_set, AA = FALSE)
```

Arguments

germline_set	A character list of aligned IGHV allele sequences.
AA	Logical (FALSE by default). If to calculate the distance based on amino acid sequences.

Value

A matrix of computed distances between allele pairs.

See Also

igDistance for the current implementation

`inferAlleleClusters` *Allele similarity cluster*

Description

A wrapper function to infer the allele clusters. Supports both hierarchical clustering (default) and Leiden community detection.

Usage

```
inferAlleleClusters(
  germline_set,
  locus = NULL,
  clustering_method = c("hierarchical", "leiden"),
  distance_method = c("decipher", "hamming", "lv"),
  trim_3prime_side = 318,
  mask_5prime_side = 0,
  family_threshold = 75,
  allele_cluster_threshold = 95,
  cluster_method = "complete",
  resolution = NULL,
  target_clusters = NULL,
  optimize_silhouette = TRUE,
  ncores = 1,
  aa_set = FALSE,
  quiet = FALSE
)
```

Arguments

`germline_set`	A character vector of Ig sequence alleles (must be gapped by IMGT scheme for optimal results).
`locus`	The locus type. One of "IGHV", "IGKV", "IGLV", "IGHD", "IGHJ", "IGKJ", "IGLJ". Default is NULL (auto-detected from sequence names).
`clustering_method`	Clustering method. One of "hierarchical" (default) or "leiden".
`distance_method`	Distance calculation method. One of "decipher" (default), "hamming", or "lv".
`trim_3prime_side`	Position to trim sequences from 3' end. Default is 318; NULL uses full length.
`mask_5prime_side`	Length to mask from 5' side. Default is 0.
`family_threshold`	Similarity threshold for family level (hierarchical only). Default is 75.
`allele_cluster_threshold`	Similarity threshold for allele cluster level (hierarchical only). Default is 95.

`cluster_method` Hierarchical clustering linkage method. Default is "complete".

`resolution` Resolution parameter for Leiden clustering. Default is NULL (auto-optimized).

`target_clusters`
Target number of clusters for Leiden optimization. Default is NULL.

`optimize_silhouette`
Optimize resolution using silhouette score (Leiden only). Default is TRUE.

`ncores` Number of cores for parallel processing (Leiden only). Default is 1.

`aa_set` Logical. Is the sequence set amino acids? Default is FALSE.

`quiet` Logical. Suppress messages. Default is FALSE.

Details

The distance between pairs of allele sequences is calculated, then the alleles are clustered. For hierarchical clustering, two similarity thresholds define family and allele clusters. For Leiden clustering, community detection identifies clusters at a specified resolution.

The allele cluster names follow this scheme: IGHVF1-G1*01 - IGH = chain, V = region, F1 = family cluster numbering, G1 = allele cluster numbering, 01 = allele numbering (by clustering order)

For V segments, the "decipher" distance method is recommended. For D and J segments with variable lengths, "lv" (Levenshtein) is more appropriate.

Value

An object of class GermlineCluster containing:

- germlineSet: Modified germline set (3' trimming and 5' masking)
- alleleClusterSet: Renamed germline set with ASC names
- alleleClusterTable: data.frame of allele similarity clusters
- threshold: List of threshold parameters
- hclustAlleleCluster: hclust object (hierarchical method)
- clusteringMethod: Method used ("hierarchical" or "leiden")
- communityObject: Community object (Leiden method)
- graphObject: igraph object (Leiden method)
- silhouetteScore: Silhouette score (Leiden method)
- resolutionParameter: Resolution used (Leiden method)
- locus: Locus identifier

See Also

`igDistance`, `igClust`, `plot.GermlineCluster`

Examples

```
# load the initial germline set

data(HVGERM)

germline <- HVGERM[!grepl("^[.]", HVGERM)]

# Hierarchical clustering (default)
asc <- inferAlleleClusters(germline)

# Leiden community detection
asc_leiden <- inferAlleleClusters(germline[1:50],
                                  clustering_method = "leiden",
                                  target_clusters = 10)

## plotting the clusters
plot(asc)
```

`inferGenotypeAllele` *Allele based genotype inference*

Description

`inferGenotypeAllele` infer an individual's genotype based on the allele-base method. The method utilize the allele specific threshold to determine the presence of an allele in the genotype. More specifically, based on the allele frequency, repertoire depth, and the specific allele threshold, a confidence level (Z score) is calculated for the presence of the allele in the genotype. The user can select the confidence level for the genotype inference.

Usage

```
inferGenotypeAllele(
  data,
  allele_threshold_table = NULL,
  call = "v_call",
  asc_annotation = FALSE,
  single_assignment = FALSE,
  translate_to_asc = FALSE,
  germline_db = NA,
  find_unmutated = FALSE,
  seq = "sequence_alignment",
  default_allele_threshold = 1e-04,
  quiet = TRUE
)
```

Arguments

`data`	data.frame in AIRR format, containing allele calls from a single subject and the sample IMGT-gapped V(D)J sequences under seq.
`allele_threshold_table`	A data.frame of the alleles and their thresholds.
`call`	name of the V,D, or J allele call column, i.e v_call, d_call, j_call. Default is `v_call`
`asc_annotation`	Logical (FALSE by default). Are the allele calls annotated with the allele similarity clusters.
`single_assignment`	if TRUE, the method only considers sequence with single assignment for the genotype inference.
`translate_to_asc`	For V allele calls, collapse identical allele for the genotype inference. Default is FALSE.
`germline_db`	named vector of sequences containing the germline sequences named in V allele calls and the alleleClusterTable. Only required if find_unmutated is TRUE.
`find_unmutated`	if TRUE, use germline_db to find which samples are unmutated. Not needed if V allele calls only represent unmutated samples.
`seq`	name of the column in data with the aligned, IMGT-numbered, V(D)J nucleotide sequence. Default is sequence_alignment.
`default_allele_threshold`	The default allele threshold for the genotype inference, in case the allele threshold is not in the `allele_threshold_table`. Default is 1e-04.
`quiet`	Logical (TRUE by default). Do you want to suppress informative messages

Details

In naive repertoires, allele calls where more than one assignment is assigned is rare. Hence, in case the data represents the naive repertoire of a subject it is recommended to use the `find_unmutated=TRUE` option, to remove mutated sequences. For non-naive population, the allele calls in cases of multiple assignment are treated as belonging to all groups.

Value

A a data.frame with the inferred V genotype. The table contains the following columns:

- allele: The alleles in the `allele_threshold_table`.
- counts: The number of reads for each alleles.
- depth: The total number of reads in the genotype (Sum of counts).
- threshold: The population driven allele thresholds for genotype presence.
- z_score: The confidence level for the presence of the allele in the genotype.
- asc_allele: If `translate_to_asc` is true, the asc allele value from allele_threshold_table.

See Also

inferAlleleClusters will infer the allele clusters based on a supplied V reference set and set the default allele threshold of 1e-04. See recentAlleleClusters to obtain the latest version of the IGHV allele clusters and the naive population based allele threshold.

Examples

```
# loading TIgGER AIRR-seq b cell data
data <- tigger::AIRRDb

# allele threshold table
data(allele_threshold_table)

data(HVGERM)

# inferring the genotype
genotype <- inferGenotypeAllele(
data = data,
allele_threshold_table = allele_threshold_table,
germline_db = HVGERM, find_unmutated=TRUE)

# filter alleles with z_score >= 0

head(genotype[genotype$z_score >= 0,])
```

`inferGenotypeAllele_asc` *Allele similarity cluster based genotype inference Testing function*

Description

`inferGenotypeAllele_asc` infer an individual's genotype based on the allele-base method. The method utilize the allele specific threshold to determine the presence of an allele in the genotype. More specifically, the absolute frequency of each allele is calculated and checked against the threshold.

Usage

```
inferGenotypeAllele_asc(
  data,
  alleleClusterTable,
  v_call = "v_call",
  single_assignment = FALSE,
  germline_db = NA,
  find_unmutated = FALSE,
  seq = "sequence_alignment",
```

```
  confidence_level = NULL,
  default_allele_threshold = 1e-04
)
```

Arguments

`data`	data.frame in AIRR format, containing V allele calls from a single subject and the sample IMGT-gapped V(D)J sequences under seq.
`alleleClusterTable`	A data.frame of the allele similarity clusters thresholds.
`v_call`	name of the V allele call column. Default is `v_call`
`single_assignment`	if TRUE, the method only considers sequence with single assignment for the genotype inference.
`germline_db`	named vector of sequences containing the germline sequences named in V allele calls and the alleleClusterTable. Only required if find_unmutated is TRUE.
`find_unmutated`	if TRUE, use germline_db to find which samples are unmutated. Not needed if V allele calls only represent unmutated samples.
`seq`	name of the column in data with the aligned, IMGT-numbered, V(D)J nucleotide sequence. Default is sequence_alignment.
`confidence_level`	The confidence level on which to filter the inferred genotype alleles. Default is NULL, meaning filtering only based on allele threshold.
`default_allele_threshold`	The default allele threshold for the genotype inference, in case the allele threshold is not in the `alleleClusterTable`. Default is 1e-04.

Details

In naive repertoires, allele calls where more than one assignment is assigned is rare. Hence, in case the data represents the naive repertoire of a subject it is recommended to use the `find_unmutated=TRUE` option, to remove mutated sequences. For non-naive population, the allele calls in cases of multiple assignment are treated as belonging to all groups.

Value

A a data.frame with the inferred V genotype. The table contains the following columns:

gene allele cluster	alleles the present alleles in the repertoire	imgt_alleles the imgt nomenclature of the alleles	counts the number of reads for each alleles	absolute_fraction the absolute fraction of the alleles	absolute_threshol the population dri thresholds for gen

See Also

inferAlleleClusters will infer the allele clusters based on a supplied V reference set and set the default allele threshold of 1e-04. See recentAlleleClusters to obtain the latest version of the IGHV allele clusters and the naive population based allele threshold.

Examples

```
# loading TIgGER AIRR-seq b cell data
data <- tigger::AIRRDb

# preferably obtain the latest ASC cluster table
# asc_archive <- recentAlleleClusters(doi="10.5281/zenodo.7429773", get_file = TRUE)

# allele_cluster_table <- extractASCTable(archive_file = asc_archive)

# example allele similarity cluster table
data(allele_cluster_table)

data(HVGERM)

# reforming the germline set
asc_germline <- germlineASC(allele_cluster_table, germline = HVGERM)

# assigning the ASC alleles
asc_data <- assignAlleleClusters(data, allele_cluster_table)

# inferring the genotype
asc_genotype <- inferGenotypeAllele_asc(
data = asc_data,
alleleClusterTable = allele_cluster_table,
germline_db = asc_germline, find_unmutated=TRUE)
```

---

insert_gaps2_vec *Insert gaps into an ungapped sequence based on a gapped reference sequence.*

Description

This function inserts gaps (e.g., . or -) into an ungapped sequence (ungapped) to match the positions of gaps in a reference sequence (gapped). It ensures that the aligned sequence has the same gap structure as the reference.

Usage

```
insert_gaps2_vec(gapped, ungapped, parallel = FALSE)
```

Arguments

gapped	A vector of strings representing the reference sequences with gaps.
ungapped	A vector of strings representing the sequences without gaps.
parallel	A boolean flag to enable parallel processing (default: FALSE).

Value

A vector of strings with gaps inserted to match the gapped reference.

Examples

```
# Example usage
gapped <- c("caggtc..aact", "caggtc---aact")
ungapped <- c("caggtcaact", "caggtcaact")

# Sequential execution
result <- insert_gaps2_vec(gapped, ungapped, parallel = FALSE)
print(result)  # "caggtc..aact", "caggtc---aact"

# Parallel execution
result_parallel <- insert_gaps2_vec(gapped, ungapped, parallel = TRUE)
print(result_parallel)
```

---

new_germline_cluster *Create a GermlineCluster object*

Description

`GermlineCluster` is an S3 class that stores the output of `inferAlleleClusters`. It contains the allele cluster table, clustering objects, and threshold parameters used for inference.

Usage

```
new_germline_cluster(
  germlineSet,
  alleleClusterSet,
  alleleClusterTable,
  threshold,
  hclustAlleleCluster = NULL,
  clusteringMethod = "hierarchical",
  communityObject = NULL,
  graphObject = NULL,
  distanceMatrix = NULL,
  silhouetteScore = NA_real_,
  resolutionParameter = NA_real_,
  locus = "IGHV"
)
```

Arguments

`germlineSet`	The original germline set provided.
`alleleClusterSet`	The renamed germline set with allele clusters.

`alleleClusterTable`
: The allele cluster table.

`threshold`
: The threshold used for family and allele clusters.

`hclustAlleleCluster`
: A hierarchical clustering object for the germline set, or `NULL`.

`clusteringMethod`
: The clustering method used, either `"hierarchical"` or `"leiden"`.

`communityObject`
: A community detection object for Leiden clustering, or `NULL`.

`graphObject`
: An **igraph** graph object for Leiden clustering, or `NULL`.

`distanceMatrix`
: The distance matrix used for clustering, or `NULL`.

`silhouetteScore`
: The silhouette score for community detection.

`resolutionParameter`
: The resolution parameter used for Leiden clustering.

`locus`
: The locus identifier, for example `"IGHV"`, `"IGHD"`, `"IGHJ"`.

Value

An object of class `"GermlineCluster"`.

See Also

`inferAlleleClusters`

`GermlineCluster`

---

optimize_resolution *Optimize resolution parameter using silhouette score*

Description

Performs a grid search over resolution parameters and selects the one that maximizes the silhouette score.

Usage

```
optimize_resolution(
  g,
  distance_matrix,
  target_clusters = 80,
  resolution_range_low = 0.1,
  resolution_range_high = 0.5,
  max_steps = 20,
  ncores = 1
)
```

Arguments

g	An igraph graph object with weighted edges
distance_matrix	The distance matrix (as dist object) used for silhouette calculation
target_clusters	Target number of clusters for initial tuning. Default is 80.
resolution_range_low	Fractional range below tuned resolution. Default is 0.1.
resolution_range_high	Fractional range above tuned resolution. Default is 0.5.
max_steps	Maximum steps for initial tuning. Default is 20.
ncores	Number of cores for parallel processing. Default is 1.

Value

A list containing:

- results: data.frame with Resolution, ClusterCount, Silhouette
- partitions: list of membership vectors for each resolution
- best_resolution: optimal resolution parameter
- best_partition: membership vector at optimal resolution
- best_clusters: number of clusters at optimal resolution

See Also

detect_communities_leiden, igClust

---

piglet *The Program for Ig clusters (PIgLET) package*

Description

PIgLET is a suite of computational tools that improves genotype inference and downstream AIRR-seq data analysis. The package as two main tools. The first is Allele Clusters, this tool is designed to reduce the ambiguity within the IGHV alleles. The ambiguity is caused by duplicated or similar alleles which are shared among different genes. The second tool is an allele based genotype, that determined the presence of an allele based on a threshold derived from a naive population.

Allele Similarity Cluster

This section provides the functions that support the main tool of creating the allele similarity cluster form an IGHV germline set.

- inferAlleleClusters: The main function of the section to create the allele clusters based on a germline set.
- ighvDistance: Calculate the distance between IGHV aligned germline sequences.
- ighvClust: Hierarchical clustering of the distance matrix from `ighvDistance`.
- generateReferenceSet: Generate the allele clusters reference set.
- plotAlleleCluster: Plots the Hierarchical clustering.
- artificialFRW1Germline: Artificially create an IGHV reference set with framework1 (FWR1) primers.

Allele based genotype

This section provides the functions to infer the IGHV genotype using the allele based method and the allele clusters thresholds

- inferGenotypeAllele: Infer the IGHV genotype using the allele based method.
- assignAlleleClusters: Renames the v allele calls based on the new allele clusters.
- germlineASC: Converts IGHV germline set to ASC germline set.
- recentAlleleClusters: Download the most recent version of the allele clusters table archive from zenodo.
- extractASCTable: Extracts the allele cluster table from the zenodo archive file.
- zenodoArchive: An R6 object to query the zenodo api.

References

1. ##

`plot.GermlineCluster` *Plot method for GermlineCluster*

Description

Plot method for GermlineCluster

Usage

```
## S3 method for class 'GermlineCluster'
plot(x, y = NULL, cex = 1, seed = 9999, ...)
```

Arguments

x	GermlineCluster object
y	Not used
cex	Controls the size of the allele label. Default is 1.
seed	Set a seed number for drawing the dendrogram. Default 9999.
...	Additional arguments passed to plotting functions

Value

A plot of the allele clusters dendrogram

---

plotAlleleCluster *Plotting the dendrogram of the clusters*

Description

Plotting the dendrogram of the clusters

Usage

```
plotAlleleCluster(x, y = NULL, cex = 1, seed = 9999)
```

Arguments

x	The GermlineCluster object. See inferAlleleClusters
y	NULL. not in use.
cex	Controls the size of the allele label. Default is 1.
seed	Set a seed number for drawing the dendrogram. Default 9999.

Value

A plot of the allele clusters dendrogram

plotClusterComparison *Compare hierarchical and Leiden clustering*

Description

Creates a comparison visualization showing cluster assignments from both methods.

Usage

```
plotClusterComparison(hierarchical_result, leiden_result, ...)
```

Arguments

hierarchical_result	GermlineCluster object from hierarchical clustering
leiden_result	GermlineCluster object from Leiden clustering
...	Additional arguments

Value

A ggplot object showing cluster agreement

See Also

inferAlleleClusters

plotCommunityNetwork *Plot community network*

Description

Creates a network visualization of allele clusters from community detection.

Usage

```
plotCommunityNetwork(
  x,
  layout = c("fr", "kk", "circle"),
  node_color = "cluster",
  node_size = "degree",
  edge_alpha = 0.3,
  show_labels = TRUE,
  label_size = 3,
  ...
)
```

Arguments

x	A GermlineCluster object with Leiden clustering
layout	Network layout: "fr" (Fruchterman-Reingold, default), "kk" (Kamada-Kawai), or "circle"
node_color	Variable for node color: "cluster" (default), "family", or a color value
node_size	Variable for node size: "degree" (default), "fixed", or a numeric value
edge_alpha	Alpha transparency for edges. Default is 0.3.
show_labels	Logical. Show node labels. Default is TRUE.
label_size	Size of node labels. Default is 3.
...	Additional arguments

Details

This function creates a network visualization showing:

- Nodes representing alleles, colored by cluster
- Edges weighted by sequence similarity
- Layout optimized by specified algorithm

Value

A ggplot object

See Also

inferAlleleClusters, detect_communities_leiden

Examples

```
data(HVGERM)
asc <- inferAlleleClusters(HVGERM[1:30],
                           clustering_method = "leiden",
                           target_clusters = 5)
plotCommunityNetwork(asc)
```

plotSilhouetteOptimization *Plot silhouette optimization results*

Description

Creates a plot showing silhouette score and cluster count across resolution values.

Usage

```
plotSilhouetteOptimization(optimization_result, highlight_best = TRUE, ...)
```

Arguments

optimization_result	Result from optimize_resolution
highlight_best	Logical. Highlight optimal resolution. Default is TRUE.
...	Additional arguments

Value

A ggplot object

See Also

optimize_resolution, igClust

Examples

```
data(HVGERM)
d <- igDistance(HVGERM[1:30], method = "hamming")
g <- distance_to_graph(d)
opt <- optimize_resolution(g, d, target_clusters = 5)
plotSilhouetteOptimization(opt)
```

`plotTruncatedTree` *Plot truncated tree visualization*

Description

Creates a circular or dendrogram tree visualization collapsed to ASC subgroup level, with optional heatmap annotations showing family assignments.

Usage

```
plotTruncatedTree(
  x,
  layout = c("circular", "dendrogram"),
  collapse_to = c("asc_subgroup", "iuis_subgroup", "family"),
  label_style = c("asc", "iuis", "both"),
  show_threshold_line = TRUE,
  threshold = 0.25,
  tip_size_by = "n_alleles",
  tip_color_by = "present",
  show_heatmap = TRUE,
  label_size = 7,
  ...
)
```

Arguments

`x`	A GermlineCluster object from `inferAlleleClusters`
`layout`	Tree layout: "circular" (default) or "dendrogram"
`collapse_to`	Level to collapse tree: "asc_subgroup" (default, based on ASC names), "iuis_subgroup" (based on original IUIS gene names), or "family"
`label_style`	Label style for tips: "asc" (default, show ASC names like IGHVF1-G1), "iuis" (show IUIS names with superscript markers if ASC splits IUIS group), or "both" (show both names)
`show_threshold_line`	Logical. Show threshold line on tree. Default is TRUE.
`threshold`	Threshold height for threshold line (0-1 scale). Default is 0.25.
`tip_size_by`	Variable for tip point size: "n_alleles" (default), "fixed", or NULL
`tip_color_by`	Variable for tip point color: "present" (default), "fraction_novel", or NULL
`show_heatmap`	Logical. Show heatmap annotation for IUIS vs ASC families. Default is TRUE.
`label_size`	Size of tip labels. Default is 7.
`...`	Additional arguments passed to ggtree

Details

This function creates a publication-quality tree visualization that:

- Renames tree tips from original allele names to ASC names (new_allele)
- Collapses alleles to ASC subgroup level (single representative per ASC group)
- Shows tip point size by number of alleles in cluster
- Adds optional heatmap track showing IUIS vs ASC family assignments
- Draws threshold line at specified height

When using `label_style = "iuis"`, if multiple ASC groups split a single IUIS subgroup, the labels are marked with superscript letters (e.g., IGHV1-2^A, IGHV1-2^B) to distinguish them.

Requires the ggtree package to be installed.

Value

A ggplot/ggtree object

See Also

`inferAlleleClusters`, `plot.GermlineCluster`

Examples

```
data(HVGERM)
asc <- inferAlleleClusters(HVGERM[1:50])

# Basic truncated tree with ASC labels
if (requireNamespace("ggtree", quietly = TRUE)) {
  plotTruncatedTree(asc, show_heatmap = FALSE)

  # With IUIS labels (marked if ASC splits IUIS group)
  plotTruncatedTree(asc, label_style = "iuis", show_heatmap = FALSE)
}
```

`print.GermlineCluster` *Print method for GermlineCluster*

Description

Print method for GermlineCluster

Usage

```
## S3 method for class 'GermlineCluster'
print(x, ...)
```

Arguments

x	A GermlineCluster object
...	Additional arguments (ignored)

Value

Invisibly returns x

---

recentAlleleClusters *Retrieving allele similarity clusters Zenodo archive*

Description

A wrapper function for zenodoArchive, download the most recent allele similarity clusters and thresholds from the zenodo archive. The clusters and thresholds are based on https://yaarilab.github.io/IGHV_reference_book/ At the moment only available for human IGHV reference set.

Usage

```
recentAlleleClusters(
  doi = "10.5281/zenodo.7401189",
  path,
  get_file = FALSE,
  quite = FALSE
)
```

Arguments

doi	The doi for the archive to download. Default is the IGHV set.
path	The output folder for saving the archive files. Default is to a temporary directory.
get_file	Logical (FALSE by default). Do you want to return the path for the file downloaded.
quite	Logical (FALSE by default). Do you want to suppress informative messages

Value

If get_file is TRUE, the function returns the path to the archive file

Examples

```
recentAlleleClusters(doi="10.5281/zenodo.7401189")
```

summary.GermlineCluster *Summary method for GermlineCluster*

Description

Summary method for GermlineCluster

Usage

```
## S3 method for class 'GermlineCluster'
summary(object, ...)
```

Arguments

object	A GermlineCluster object
...	Additional arguments (ignored)

Value

A list with summary statistics

zenodoArchive *zenodoArchive*

Description

zenodoArchive

zenodoArchive

Format

R6Class object.

Value

Object of R6Class for modelling an zenodoArchive for ASC cluster files

Public fields

`doi` zenodoArchive doi, NULL is not supplied

`all_versions` zenodoArchive if to return all versions, `true` when not specified

`sort` zenodoArchive how to sort the records, `mostrecent` when not specified

`page` zenodoArchive which page to pull in query, `1` when not specified

`size` zenodoArchive how many records per page, `20` when not specified

`zenodoVersions` zenodoArchive doi available version, a storing variable.

`zenodoQuery` zenodoArchive doi version query, a storing variable.

`download_file` zenodoArchive doi downloads files, a storing variable.

`download_url` zenodoArchive doi downloads urls, a storing variable.

Methods

Public methods:

- `zenodoArchive$new()`
- `zenodoArchive$clean_doi()`
- `zenodoArchive$zenodo_query()`
- `zenodoArchive$get_versions()`
- `zenodoArchive$get_version_files()`
- `zenodoArchive$download_zenodo_files()`
- `zenodoArchive$clone()`

Method `new()`**:** initializes the zenodoArchive

Usage:

```
zenodoArchive$new(
  doi,
  page = 1,
  size = 20,
  all_versions = "true",
  sort = "mostrecent"
)
```

Arguments:

`doi` A zenodo doi. To retrieve all records supply a concept doi (a generic doi common to all versions).

`page` Which page to query. Default is 1

`size` How many records per page. Default is 20

`all_versions` If to return all concept doi versions. If `true` returns all, if `false` returns the latest. Default is `ture`

`sort` Which sorting to apply on the records. Default is `mostrecent`. Possible sortings "best-match", "mostrecent", "-mostrecent" (ascending), "version", "-version" (ascending).

Method `clean_doi()`**:** cleans the doi record for query

Usage:

```
zenodoArchive$clean_doi(doi = self$doi)
```

Arguments:

`doi` The zenodo archive doi

Returns: the clean doi

Method `zenodo_query()`**:** Query the zenodo archive according to the initial parameters.

Usage:

```
zenodoArchive$zenodo_query(...)
```

Arguments:

`...` Excepts the self created by `initialize`

Returns: a list with the query values.

Method `get_versions()`**:** Extract all concept doi available versions.

Usage:

```
zenodoArchive$get_versions(...)
```

Arguments:

`...` Excepts the self created by `initialize`

Returns: a data.frame of the available versions.

Method `get_version_files()`**:** get the chosen doi archive version available files

Usage:

```
zenodoArchive$get_version_files(version = "latest")
```

Arguments:

`version` which archive version files to get. Default to latest. To see all available version use `get_versions`

Returns: a list of the available files in the archive version.

Method `download_zenodo_files()`**:** get the chosen doi archive version available files

Usage:

```
zenodoArchive$download_zenodo_files(
  file = NULL,
  path = tempdir(),
  version = "latest",
  all_files = F,
  get_file_path = F,
  quite = F
)
```

Arguments:

`file` If supplied, downloads the specific file from the archive.

`path` The output folder for saving the archive files. Default is to a temporary directory.

`version` which archive version files to get. Default to latest. To see all available version use `get_versions`

`all_files` Logical (FALSE by default). Do you want to download all files in the archive.

`get_file_path` Logical (FALSE by default). Do you want to return the path for the file downloaded.

`quite` Logical (FALSE by default). Do you want to suppress informative messages

Returns: If `get_file_path` is TRUE, the function returns the path to the archive file

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
zenodoArchive$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
zenodo_archive <- zenodoArchive$new(
   doi = "10.5281/zenodo.7401189"
)

# view available version ins the archive
archive_versions <- zenodo_archive$get_versions()

# Getting the available files in the latest zenodo archive version
files <- zenodo_archive$get_version_files()

# downloading the first file from the latest archive version
zenodo_archive$download_zenodo_files()
```

Index