Package ‘rabhit’

October 14, 2022

Type Package

Title Inference Tool for Antibody Haplotype

Version 0.2.4

Description Infers V-D-J haplotypes and gene deletions from AIRR-seq data for Ig and TR chains,
based on J, D, or V genes as anchor, by adapting a Bayesian framework.
It also calculates a Bayes factor, a number that indicates the certainty level of the inference, for each haplotyped gene.
Citation:
Gidoni, et al (2019) <doi:10.1038/s41467-019-08489-3>.
Peres and Gidoni, et al (2019) <doi:10.1093/bioinformatics/btz481>.

License CC BY-SA 4.0

URL https://yaarilab.bitbucket.io/RAbHIT/

BugReports https://bitbucket.org/yaarilab/rabhit/issues

LazyData true

BuildVignettes true

VignetteBuilder knitr

Encoding UTF-8

Depends R (>= 3.5.0), ggplot2 (>= 3.2.0)

Imports dplyr (>= 1.0.0), reshape2 (>= 1.4.3), plotly (>= 4.7.1),
gtools (>= 3.5.0), cowplot (>= 0.9.1), readr (>= 2.1.1),
dendextend (>= 1.9.0), data.table (>= 1.12.2), plyr (>= 1.8.5),
ggdendro (>= 0.1.20), gridExtra (>= 2.3.0), alakazam (>=
1.0.0), tigger (>= 1.0.0), methods (>= 3.4.4), htmlwidgets (>=
1.3.0), gtable (>= 0.3.0), rlang (>= 0.4.0), RColorBrewer (>=
1.1.2), tidyr (>= 1.0.0), stringi (>= 1.4.3), grid (>= 3.4.4),
splitstackshape (>= 1.4.8), fastmatch (>= 1.1.0)

Suggests knitr, rmarkdown, stats, graphics, grDevices

RoxygenNote 7.2.0

NeedsCompilation no

Collate 'Data.R' 'rabhit.R' 'internal_functions.R' 'functions.R'
'graphic_functions.R' 'zzz.R'

Author Ayelet Peres [aut, cre],
Moriah Gidoni [aut],
Gur Yaari [aut, cph]

Maintainer Ayelet Peres <peresay@biu.ac.il>

Repository CRAN

Date/Publication 2022-09-22 15:10:02 UTC

R topics documented:

`.onAttach` *.onAttach start message*

Description

.onAttach start message

Usage

```
.onAttach(libname, pkgname)
```

Arguments

`libname`	defunct
`pkgname`	defunct

Value

invisible()

`createFullHaplotype` *Anchor gene haplotype inference*

Description

The `createFullHaplotype` functions infers haplotype based on an anchor gene.

Usage

```
createFullHaplotype(
  clip_db,
  toHap_col = c("v_call", "d_call"),
  hapBy_col = "j_call",
  hapBy = "IGHJ6",
  toHap_GERM = NULL,
  relative_freq_priors = TRUE,
  kThreshDel = 3,
  rmPseudo = TRUE,
  deleted_genes = c(),
  nonReliable_Vgenes = c(),
  min_minor_fraction = 0.3,
  single_gene = TRUE,
  chain = c("IGH", "IGK", "IGL", "TRB")
)
```

Arguments

`clip_db`	a `data.frame` in AIRR format. See details.
`toHap_col`	a vector of column names for which a haplotype should be inferred. Default is v_call and d_call
`hapBy_col`	column name of the anchor gene. Default is j_call
`hapBy`	a string of the anchor gene name. Default is IGHJ6.
`toHap_GERM`	a vector of named nucleotide germline sequences matching the allele calls in `toHap_col` columns in clip_db.
`relative_freq_priors`	if TRUE, the priors for Bayesian inference are estimated from the relative frequencies in clip_db. Else, priors are set to `c(0.5,0.5)`. Default is TRUE

`kThreshDel`	the minimum lK (log10 of the Bayes factor) to call a deletion. Default is 3.
`rmPseudo`	if TRUE non-functional and pseudo genes are removed. Default is TRUE.
`deleted_genes`	double chromosome deletion summary table. A `data.frame` created by `deletionsByBinom`.
`nonReliable_Vgenes`	a list of known non reliable gene assignments. A `list` created by `nonReliableVGenes`.
`min_minor_fraction`	the minimum minor allele fraction to be used as an anchor gene. Default is 0.3
`single_gene`	if to only consider genes from single assignment. If true then calls where genes appear with others are discarded. If false then the calls are seperated an counted for all genes that appeared. Default is True.
`chain`	the IG/TR chain: IGH,IGK,IGL,TRB. Default is IGH.

Details

Function accepts a `data.frame` in AIRR format (`https://changeo.readthedocs.io/en/stable/standard.html`) containing the following columns:

- `'subject'`: The subject name
- `'v_call'`: V allele call(s) (in an IMGT format)
- `'d_call'`: D allele call(s) (in an IMGT format, only for heavy chains)
- `'j_call'`: J allele call(s) (in an IMGT format)

Value

A `data.frame`, in which each row is the haplotype inference summary of a gene from the column selected in `toHap_col`.

The output containes the following columns:

- `subject`: the subject name.
- `gene`: the gene name.
- Anchor gene allele 1: the haplotype inference for chromosome one. The column name is the anchor gene with the first allele.
- Anchor gene allele 2: the haplotype inference for chromosome two. The column name is the anchor gene with the second allele.
- `alleles`: allele calls for the gene.
- `proirs_row`: priors based on relative allele usage of the anchor gene.
- `proirs_col`: priors based on relative allele usage of the inferred gene.
- `counts1`: the appereance count on each chromosome of the first allele from `alleles`, the counts are seperated by a comma.
- `k1`: the Bayesian factor value for the first allele (from `alleles`) inference.
- `counts2`: the appereance count on each chromosome of the second allele from `alleles`, the counts are seperated by a comma.
- `k2`: the Bayesian factor value for the second allele (from `alleles`) inference.

- `counts3`: the appereance count on each chromosome of the third allele from `alleles`, the counts are seperated by a comma.
- `k3`: the Bayesian factor value for the third allele (from `alleles`) inference.
- `counts4`: the appereance count on each chromosome of the fourth allele from `alleles`, the counts are seperated by a comma.
- `k4`: the Bayesian factor value for the fourth allele (from `alleles`) inference.

Examples

```
# Load example data and germlines
data(samples_db, HVGERM, HDGERM)

# Selecting a single individual
clip_db = samples_db[samples_db$subject=='I5', ]

# Infering haplotype
haplo_db = createFullHaplotype(clip_db,toHap_col=c('v_call','d_call'),
hapBy_col='j_call',hapBy='IGHJ6',toHap_GERM=c(HVGERM,HDGERM))
```

`deletionHeatmap` *Graphical output of single chromosome deletions*

Description

The `deletionHeatmap` function generates a graphical output of the single chromosome deletions in multiple samples.

Usage

```
deletionHeatmap(
  hap_table,
  chain = c("IGH", "IGK", "IGL", "TRB", "TRA"),
  kThreshDel = 3,
  genes_order = NULL,
  html_output = FALSE
)
```

Arguments

`hap_table`	haplotype summary table. See details.
`chain`	the IG chain: IGH,IGK,IGL. Default is IGH.
`kThreshDel`	the minimum lK (log10 of the Bayes factor) used in `createFullHaplotype` to call a deletion. Indicates the color for strong deletion. Default is 3.
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc
`html_output`	If TRUE, a html5 interactive graph is outputed insteaed of the normal plot. Default is FALSE

Details

A `data.frame` created by `createFullHaplotype`.

Value

A single chromosome deletion visualization.

Examples

```
# Plotting single choromosme deletion from haplotype inference
deletionHeatmap(samplesHaplotype)
```

`deletionsByBinom` *Double chromosome deletion by relative gene usage*

Description

The `deletionsByBinom` function inferes double chromosome deletion events by relative gene usage.

Usage

```
deletionsByBinom(
  clip_db,
  chain = c("IGH", "IGK", "IGL"),
  nonReliable_Vgenes = c(),
  genes_order = NULL
)
```

Arguments

`clip_db`	a `data.frame` in AIRR format. See details.
`chain`	the IG/TR chain: IGH,IGK,IGL,TRB. Default is IGH.
`nonReliable_Vgenes`	a list of known non reliable gene assignments. A `list` created by `nonReliableVGenes`.
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc

Details

The function accepts a `data.frame` in AIRR format (`https://changeo.readthedocs.io/en/stable/standard.html`) containing the following columns:

- `'subject'`: The subject name
- `'v_call'`: V allele call(s) (in an IMGT format)
- `'d_call'`: D allele call(s) (in an IMGT format, only for heavy chains)
- `'j_call'`: J allele call(s) (in an IMGT format)

Value

A `data.frame`, in which each row is the double chomosome deletion inference of a gene.

The output containes the following columns:

- `subject`: the subject name.
- `gene`: the gene call
- `frac`: the relative gene usage of the gene
- `cutoff`: the the cutoff of for the binomial test
- `pval`: the p-value of the binomial test
- `deletion`: if a double chromosome deletion event of a gene occured.

Examples

```
# Load example data and germlines
data(samples_db)

# Selecting a single individual
clip_db = samples_db[samples_db$subject=='I5', ]
# Infering haplotype
del_binom_df = deletionsByBinom(clip_db)
head(del_binom_df)
```

`deletionsByVpooled` *Single chromosomal D or J gene deletions inferred by the V pooled method*

Description

The `deletionsByVpooled` function inferes single chromosomal deletion for D and J gene .

Usage

```
deletionsByVpooled(
  clip_db,
  chain = c("IGH", "IGK", "IGL"),
  deletion_col = c("d_call", "j_call"),
  count_thresh = 50,
  deleted_genes = "",
  min_minor_fraction = 0.3,
  kThreshDel = 3,
  nonReliable_Vgenes = c()
)
```

Arguments

`clip_db`	a `data.frame` in AIRR format. See details.
`chain`	the IG chain: IGH,IGK,IGL. Default is IGH.
`deletion_col`	a vector of column names for which single chromosome deletions should be inferred. Default is j_call and d_call.
`count_thresh`	integer, the minimun number of sequences mapped to a specific V gene to be included in the V pooled inference.
`deleted_genes`	double chromosome deletion summary table. A `data.frame` created by `deletionsByBinom`.
`min_minor_fraction`	the minimum minor allele fraction to be used as an anchor gene. Default is 0.3
`kThreshDel`	the minimum lK (log10 of the Bayes factor) to call a deletion. Default is 3.
`nonReliable_Vgenes`	a list of known non reliable gene assignments. A `list` created by `nonReliableVGenes`.

Details

The function accepts a `data.frame` in AIRR format (`https://changeo.readthedocs.io/en/stable/standard.html`) containing the following columns:

- `'subject'`: The subject name
- `'v_call'`: V allele call(s) (in an IMGT format)
- `'d_call'`: D allele call(s) (in an IMGT format, only for heavy chains)
- `'j_call'`: J allele call(s) (in an IMGT format)

Value

A `data.frame`, in which each row is the single chomosome deletion inference of a gene.

The output containes the following columns:

- `subject`: the subject name.
- `gene`: the gene call
- `deletion`: chromosome deletions inferred. Encoded 1 for deletion and 0 for no deletion.
- `k`: the Bayesian factor value for the deletion inference.
- `counts`: the appereance count of the gene on each chromosome, the counts are seperated by a comma.

Examples

```
data(samples_db)

# Infering V pooled deletions
del_db <- deletionsByVpooled(samples_db)
head(del_db)
```

`GENE.loc` *Human Gene order on the chromosome*

Description

A list of the chains genes order by their location on the chromosomes

Usage

```
GENE.loc
```

Format

A nested list with three enteries, each a vector of the IG chains (IGH, IGL, and IGK) genes ordered by location.

`geneUsage` *Double chromosome deletion by relative gene usage*

Description

The `geneUsage` function calculates the relative gene usage.

Usage

```
geneUsage(
  clip_db,
  chain = c("IGH", "IGK", "IGL", "TRB"),
  genes_order = NULL,
  rmPseudo = TRUE
)
```

Arguments

`clip_db`	a `data.frame` in AIRR format. See details.
`chain`	the IG/TR chain: IGH,IGK,IGL,TRB. Default is IGH.
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc
`rmPseudo`	if TRUE non-functional and pseudo genes are removed. Default is TRUE.

Details

The function accepts a `data.frame` in AIRR format (`https://changeo.readthedocs.io/en/stable/standard.html`) containing the following columns:

- `'subject'`: The subject name
- `'v_call'`: V allele call(s) (in an IMGT format)
- `'d_call'`: D allele call(s) (in an IMGT format, only for heavy chains)
- `'j_call'`: J allele call(s) (in an IMGT format)

Value

A `data.frame`, in which each row is the relative gene usage value per individual.

The output containes the following columns:

- `subject`: the subject name.
- `gene`: the gene call
- `frac`: the relative gene usage of the gene

`GERM` *Human germlines*

Description

A list of the germline genes from the human immunoglobulin loci

Usage

```
GERM
```

Format

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

`hapDendo` *Hierarchical clustering of haplotypes graphical output*

Description

The `hapDendo` function generates a graphical output of an hierarchical clustering based on the Jaccard distance between multiple samples' haplotypes.

Usage

```
hapDendo(
  hap_table,
  chain = c("IGH", "IGK", "IGL", "TRB", "TRA"),
  genes_order = NULL,
  removeIGH = TRUE,
  mark_low_lk = TRUE,
  lk_cutoff = 1
)
```

Arguments

`hap_table`	haplotype summary table. See details.
`chain`	the IG/TR chain: IGH,IGK,IGL,TRB. Default is IGH.
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc
`removeIGH`	if TRUE, 'IGH'\'IGK'\'IGL' prefix is removed from gene names. Default is TRUE.
`mark_low_lk`	if TRUE, a texture is add for low lK values. Default is TRUE.
`lk_cutoff`	the lK cutoff value to be considerd low for texture layer. Default is lK<1.

Details

A `data.frame` created by `createFullHaplotype`.

Value

A multitple samples visualization of the distances between haplotypes.

Examples

```
# Plotting haplotype hierarchical clustering based on the Jaccard distance
hapDendo(samplesHaplotype)
```

`hapHeatmap` *Graphical output of alleles division by chromosome*

Description

The `hapHeatmap` function generates a graphical output of the alleles per gene in multiple samples.

Usage

```
hapHeatmap(
  hap_table,
  chain = c("IGH", "IGK", "IGL", "TRB", "TRA"),
  genes_order = NULL,
  removeIGH = TRUE,
  lk_cutoff = 1,
  mark_low_lk = TRUE,
  size_annot = 1.5,
  color_y = NULL,
  order_subject = NULL,
  file = NULL,
  size_text = NULL,
  ylabel_size = 1
)
```

Arguments

`hap_table`	haplotype summary table. See details.
`chain`	the IG chain: IGH,IGK,IGL. Default is IGH.
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc
`removeIGH`	if TRUE, ’IGH’\’IGK’\’IGL’\’TRB’ prefix is removed from gene names.
`lk_cutoff`	the lK cutoff value to be considered low for texture layer. Default is lK<1.
`mark_low_lk`	if TRUE, a texture is add for low lK values. Default is TRUE.
`size_annot`	size of bottom annotation text. Default is 1.5 .
`color_y`	named list of the colors for y axis labels.
`order_subject`	order subject by a vector.
`file`	file path for rendering the plot to pdf. If non is supplied than the plot is returned as object. Default is NULL.
`size_text`	text size for annotations.
`ylabel_size`	text size for y axis labels.

Details

A `data.frame` created by `createFullHaplotype`.

Value

A list with the following:

- `'p'`: heat-map visualization of the haplotype inference for multiple samples.
- `'width'`: Optimal width value for rendering plot.
- `'height'`: Optimal width value for rendering plot.

When a file is supplied the graph is also rendered to pdf.

Examples

```
# Plotting haplotpe heatmap
p <- hapHeatmap(samplesHaplotype)
p$p
```

`HDGERM` *Human IGHD germlines*

Description

A `character` vector of all 37 human IGHD germline gene segment alleles in IMGT Gene-db release 2018-12-4.

Usage

```
HDGERM
```

Format

Values correspond to IMGT nuceltoide sequences.

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.

`HJGERM` *Human IGHJ germlines*

Description

A `character` vector of all 13 human IGHJ germline gene segment alleles in IMGT Gene-db release 2018-12-4.

Usage

```
HJGERM
```

Format

Values correspond to IMGT nuceltoide sequences.

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.

`HVGERM` *Human IGHV germlines*

Description

A `character` vector of all 342 human IGHV germline gene segment alleles in IMGT Gene-db release 2018-12-4.

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.

`KJGERM` *Human IGKJ germlines*

Description

A `character` vector of all 342 human IGKJ germline gene segment alleles in IMGT Gene-db release 2019-11-18.

Usage

```
KJGERM
```

Format

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

`KVGERM` *Human IGKV germlines*

Description

A `character` vector of all 342 human IGKV germline gene segment alleles in IMGT Gene-db release 2019-11-18.

A `character` vector of all 342 human IGLV germline gene segment alleles in IMGT Gene-db release 2019-11-18.

Usage

```
KVGERM

LVGERM
```

Format

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

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

`LJGERM` *Human IGLJ germlines*

Description

A `character` vector of all 342 human IGLJ germline gene segment alleles in IMGT Gene-db release 2019-11-18.

Usage

```
LJGERM
```

Format

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

`nonReliableVGenes` *Detect non reliable gene assignment*

Description

`nonReliableVGenes` Takes a `data.frame` in AIRR format and detect non reliable IGHV genes. A non reliable gene is when the ratio of the multiple assignments with a gene is below the threshold.

Usage

```
nonReliableVGenes(clip_db, thresh = 0.9, appearance = 0.01)
```

Arguments

`clip_db`	a `data.frame` in AIRR format. See details.
`thresh`	the threshold to consider non reliable gene. Default is 0.9
`appearance`	the minimum fraction of gene appearance to be considered for reliability check. Default is 0.01.

Details

The function accepts a `data.frame` in AIRR format (`https://changeo.readthedocs.io/en/stable/standard.html`) containing the following columns:

- `'subject'`: subject names
- `'v_call'`: V allele call(s) (in an IMGT format)

Value

a nested list of non reliable genes for all subject.

Examples

```
# Example IGHV call data frame
clip_db <- data.frame(subject=rep('S1',6),
v_call=c('IGHV1-69*01','IGHV1-69*01','IGHV1-69*01,IGHV1-69*02',
'IGHV4-59*01,IGHV4-61*01','IGHV4-59*01,IGHV4-31*02','IGHV4-59*01'))
# Detect non reliable genes
nonReliableVGenes(clip_db)
```

`plotDeletionsByBinom` *Graphical output of double chromosome deletions*

Description

The `plotDeletionsByBinom` function generates a graphical output of the double chromosome deletions in multiple samples.

Usage

```
plotDeletionsByBinom(
  GENE.usage.df,
  chain = c("IGH", "IGK", "IGL", "TRB", "TRA"),
  genes.low.cer = c("IGHV3-43", "IGHV3-20"),
  genes.dup = c("IGHD4-11", "IGHD5-18"),
  genes_order = NULL
)
```

Arguments

`GENE.usage.df`	double chromosome deletion summary table. See details.
`chain`	the IG chain: IGH,IGK,IGL. Default is IGH.
`genes.low.cer`	a vector of IGH genes known to be with low certantiny in the binomial test. Default is IGHV3-43 and IGHV3-20
`genes.dup`	a vector of IGH genes known to have a duplicated gene. Default is IGHD4-11 that his duplicate is IGHD4-4 and IGHD5-18 that his duplicate is IGHD5-5
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc

Details

A `data.frame` created by `binom_test_deletion`.

Value

A double chromosome deletion visualization.

Examples

```
# Load example data and germlines
data(samples_db)

# Infering haplotype
deletions_db = deletionsByBinom(samples_db);
plotDeletionsByBinom(deletions_db)
```

`plotDeletionsByVpooled` *Graphical output for single chromosome D or J gene deletions according to V pooled method*

Description

The `plotDeletionsByVpooled` function generates a graphical output for single chromosome D or J gene deletions (for heavy chain only).

Usage

```
plotDeletionsByVpooled(
  del.df,
  chain = c("IGH", "IGK", "IGL", "TRB", "TRA"),
  K_ranges = c(3, 7)
)
```

Arguments

`del.df`	a `data.frame` created by `deletionsByVpooled`
`chain`	the IG chain: IGH,IGK,IGL. Default is IGH..
`K_ranges`	vector of one or two integers for log(K) certainty level thresholds

Details

A `data.frame` created by `deletionsByVpooled`.

Value

A single chromosome deletion visualization.

Examples

```
# Load example data and germlines
data(samples_db)
del_db <- deletionsByVpooled(samples_db)
plotDeletionsByVpooled(del_db)
```

`plotHaplotype` *Graphical output of an inferred haplotype*

Description

The `plotHaplotype` functions visualizes an inferred haplotype.

Usage

```
plotHaplotype(
  hap_table,
  html_output = FALSE,
  genes_order = NULL,
  text_size = 14,
  removeIGH = TRUE,
  plotYaxis = TRUE,
  chain = c("IGH", "IGK", "IGL", "TRB"),
  dir
)
```

Arguments

`hap_table`	haplotype summary table. See details.
`html_output`	if TRUE, a html5 interactive graph is outputed. Default is FALSE.
`genes_order`	A vector of the genes by the desired order. Default is by GENE.loc
`text_size`	the size of graph labels. Default is 14 (pts).
`removeIGH`	if TRUE, ’IGH’\’IGK’\’IGL’\’TRB’ prefix is removed from gene names.
`plotYaxis`	if TRUE, Y axis labels (gene names) are plotted on the middle and right plots. Default is TRUE.
`chain`	the Ig/TR chain: IGH,IGK,IGL,TRB. Default is IGH.
`dir`	The output folder for saving the haplotype map for multiple individuals.

Details

A `data.frame` in a haplotype format created by `createFullHaplotype` function.

Value

A haplotype map visualization. If more than one subject is visualized, a pdf is created. If html_output is TRUE, a folder named html_output is created with individual graphs.

Examples

```
# Selecting a single individual from the haplotype samples data
haplo_db = samplesHaplotype[samplesHaplotype$subject=='I5', ]

# plot haplotype
plotHaplotype(haplo_db)
```

`rabhit` *The RAbHIT package*

Description

The `rabhit` package provides a robust novel method for determining antibody heavy and light chain haplotypes by adapting a Bayesian framework. The key functions in `rabhit`, broken down by topic, are described below.

Haplotype and deletions inference

`rabhit` provides tools to infer haplotypes based on given anchor genes, deletion detection based on relative gene usage, pooling v genes, and a single anchor gene.

- createFullHaplotype: Haplotypes inference and single chromosome deletions based on an anchor gene.
- deletionsByVpooled: Single chromosomal deletion detection by pooling V genes.
- deletionsByBinom: Double chromosomal deletion detection by relative gene usage.
- geneUsage: Relative gene usage.
- nonReliableVGenes: Non reliable gene assignment detection.

Haplotype and deletions visualization

Functions for visualization of the inferred haplotypes and deletions

- plotHaplotype: Haplotype inference map.
- deletionHeatmap: Single chromosome deletions heatmap.
- hapHeatmap: Chromosome comparison of multiple samples.
- hapDendo: Hierarchical clustering of multiple haplotypes based on Jaccard distance.
- plotDeletionsByVpooled: V pooled based single chromosome deletions heatmap.
- plotDeletionsByBinom: Double chromosome deletions heatmap.

References

1. Gidoni, M., Snir, O., Peres, A., Polak, P., Lindeman, I., Mikocziova, I., . . . Yaari, G. (2019). Mosaic deletion patterns of the human antibody heavy chain gene locus shown by Bayesian haplotyping. Nature Communications, 10(1). doi:10.1038/s41467-019-08489-3

`readHaplotypeDb` *Read a Change-O tab-delimited database file*

Description

`readHaplotypeDb` reads a tab-delimited haplotype file created by a createFullHaplotype into a data.frame. Based on readChangeoDb function from alakazam.

Usage

```
readHaplotypeDb(file)
```

Arguments

`file`	tab-delimited database file output by a Change-O tool.

Value

A data.frame of the haplotype file. Columns will be imported as is, except for the following columns which will be explicitly converted into character values:

- alleles
- subject

`samplesHaplotype` *Example haplotype inference results*

Description

A `data.frame` of example haplotype infrence results from createFullHaplotype after double chromosome deletion inference via deletionsByBinom and non reliable V genes detection via nonReliableVGenes. Source data is a colletion of IGH human naive b-cell repertiore data from five individuals (see references). Overall, the data set includes 6 samples. A single individual has two samples (Individual I5), one is short read sequences from BIOMED-2 protocol primers for framework 2 region (The sample is annotated I5_FR2).

Usage

```
samplesHaplotype
```

Format

A `data.frame`, in which each row is the haplotype inference summary of a gene of an individual, from the column selected to prefrom the haplptype infrence on.

References

Gidoni, Moriah, *et al*. Mosaic deletion patterns of the human antibody heavy chain gene locus shown by Bayesian haplotyping. *Nature Communications*. 10.1 (2019): 628.

See Also

See createFullHaplotype for detailed column descriptions.

`samples_db` *Example IGH human naive b-cell repertiore*

Description

A `data.frame` of example IGH human naive b-cell repertiore data from five individuals (see references). Overall, the data set includes 6 samples. A single individual has two samples (Individual I5), one is short read sequences from BIOMED-2 protocol primers for framework 2 region (The sample is annotated I5_FR2).

Usage

```
samples_db
```

Format

A `data.frame` in Change-O format (`https://changeo.readthedocs.io/en/stable/standard.html`) containing the following columns:

- `'SUBJECT'`: subject names
- `'V_CALL'`: V allele call(s) (in an IMGT format)
- `'D_CALL'`: D allele call(s) (in an IMGT format, only for heavy chains)
- `'J_CALL'`: J allele call(s) (in an IMGT format)

References

Gidoni, Moriah, *et al*. Mosaic deletion patterns of the human antibody heavy chain gene locus shown by Bayesian haplotyping. *Nature Communications*. 10.1 (2019): 628.

Index