Package 'bcRep'

December 19, 2016

Type Package

Title Advanced Analysis of B Cell Receptor Repertoire Data

Version 1.3.6

Date 2016-12-16

Author Julia Bischof

Maintainer Julia Bischof <Julia.Bischof@uksh.de>

Description Methods for advanced analysis of B cell receptor repertoire data, like gene usage, mutations, clones, diversity, distance measures and multidimensional scaling and their visualisation.

License GPL-2

Depends R (>= 3.2.2)

Imports vegan, parallel, doParallel, foreach, gplots, ineq, ape, stringdist, proxy, plotrix

Suggests stats, utils, graphics, grDevices, knitr, rmarkdown, pander

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-19 13:25:03

R topics documented:

---

bcRep-package *Advanced Analysis of B Cell Receptor Repertoire Data*

Description

This package helps to analyze IMGT/HighV-QUEST output data, in more detail. It functions well with B cells, but can also be used for T cell data, in some cases. Using this package IMGT/HighV-QUEST output files can be readed and sequences and clones studied. In special their functionality, junction frames, gene usage and mutations. Functions to analyze clones, but also to compare sequences and clones, are provided. Further distances based on sequence or gene usage data can be calculated and multidimensional scaling can be performed.

Details

Package:	bcRep
Type:	Package
Version:	1.0
Date:	2015-09-09
License:	GPL-2

For many of the functions output data from IMGT/HighV-QUEST is required. In this case the particularly input file is given in the corresponding help file and `readIMGT()` can be used to read these files. IMGT files from different folders can be combined using `combineIMGT()`. For sequence analysis functions like `sequences.functionality()` or `sequences.junctionFrame()` exist, which give an overview about functionality or junction frame usage of sequences. Further gene usage (`geneUsage()`) and gene/gene combinations (`sequences.geneComb()`) can be analyzed. The function `sequences.mutation()` returns an overview of all mutations, replacement and silent mutations, as well as the R/S ratio in several regions like V, FR1-3, CDR1-2. IMGT/HighV-QUEST output can also be used to analyze clones (`clones()`). Therefore criteria can be changed (CDR3 identity, V and J gene usage) and results of different samples be compared (`clones.shared()`). Further more amino acid distributions, as well as Gini index, richness and diversity of sequences, can be studied (`AADistribution()`, `trueDiversity()`, `clones.GiniIndex()`). Principal coordinate analysis on sequences, as well as on gene usage data can be perfomed, using different distances (`sequences.distance()`, `geneUsage.distance()`, `dist.PCoA`). Special plot function for most of the methods are provided.

Author(s)

Julia Bischof

Maintainer: Julia Bischof <Julia.Bischof@uksh.de>

References

Alamyar, E. et al., IMGT/HighV-QUEST: A High-Throughput System and Web Portal for the Analysis of Rearranged Nucleotide Sequences of Antigen Receptors, JOBIM 2010, Paper 63 (2010). Website: http://www.imgt.org/

Brochet, X. et al., Nucl. Acids Res. 36, W503-508 (2008). PMID: 18503082

IMGT/V-QUEST Documentation: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#output3

IMGT Repertoire (IG and TR): http://www.imgt.org/IMGTrepertoire/LocusGenes/

Lou Jost: Entropy and diversity; OIKOS 113:2 (2006)

Examples

```
## Not run:
data(summarytab)
data(aaseqtab)
data(aaseqtab2)
data(mutationtab)
data(clones.ind)
data(clones.allind)
data(vgenes)

## Combine IMGT/HighV-QUEST folders and read data
combineIMGT(folders = c("pathTo/IMGT1a", "pathTo/IMGT1b", "pathTo/IMGT1c"),
name = "NewProject)
tab<-readIMGT("PathTo/file.txt",filterNoResults=TRUE)
```

```
## Get information about functionality and filter for functional sequences
functionality<-sequences.functionality(data = summarytab$Functionality)
ProductiveSequences<-sequences.getProductives(summarytab)

## Gene usage
Vsubgroup.usage<-geneUsage(genes = clones.ind$V_gene,
     functionality = clones.ind$Functionality_all_sequences, level = "subgroup",
     abundance="relative")

Vgenes.comp<-compare.geneUsage(gene.list = list(aaseqtab$V_GENE_and_allele,
     aaseqtab2$V_GENE_and_allele), level = "subgroup", abundance = "relative",
     names = c("IndA", "IndB"), nrCores = 1)
plotCompareGeneUsage(comp.tab = Vgenes.comp, color = c("gray97", "darkblue"),
     PDF = "Example")


## Gene/gene combinations
VDcomb.tab<-sequences.geneComb(family1 = summarytab$V_GENE_and_allele,
     family2 = summarytab$D_GENE_and_allele, level = "subgroup",
     abundance = "relative")
plotGeneComb(geneComb.tab=VDcomb.tab, color="red", withNA=FALSE,PDF="test")


## Mutation analysis
mutation.V<-sequences.mutation(mutationtab = mutationtab, summarytab = summarytab,
     sequence = "V")
mutation.CDR1<-sequences.mutation(mutationtab = mutationtab, sequence = "CDR1",
     functionality = TRUE, junctionFr = TRUE)

## Defining, Filtering and Plotting Clone features
clones.tab<-clones(aaseqtab=aaseqtab,summarytab=summarytab, identity=0.85, useJ=TRUE,
     dispCDR3aa=TRUE, dispFunctionality.ratio=TRUE, dispFunctionality.list=TRUE)
plotClonesCDR3Length(CDR3Length = clones.ind$CDR3_length_AA,
     functionality = clones.ind$Functionality_all_sequences,
     color="gray",abundance="relative", PDF="test")
clones.func<-clones.filterFunctionality(clones.tab = clones.ind,
     filter = "productive")

## Find shared clones between individuals
sharedclones<-clones.shared(clones.tab = clones.allind, identity = 0.85, useJ = TRUE,
     dispD = TRUE, dispCDR3aa = TRUE)
sharedclones.summary<-clones.shared.summary(shared.tab = sharedclones)

## True diversity
trueDiv<-trueDiversity(sequences = aaseqtab$CDR3_IMGT, order = 1)
plotTrueDiversity(trueDiversity.tab=trueDiv,color="red",PDF="test")

trueDiv.comp<-compare.trueDversity(sequence.list = list(aaseqtab$CDR3_IMGT,
     aaseqtab2$CDR3_IMGT), names = c("IndA", "IndB"), order = 1, nrCores = 1)
plotCompareTrueDiversity(comp.tab = trueDiv.comp, PDF = "Example")


## Gini index
```

```
gini<-gini<-clones.giniIndex(clone.size=clones.ind$total_number_of_sequences)

## Dissmilarity/distance indices of gene usage and sequence data

distGeneUsage<-geneUsage.distance(geneUsage.tab = Vgenes, method = "bc")
distSequence<-sequences.distance(sequences = clones.ind$unique_CDR3_sequences_AA,
     method = "levenshtein", divLength=TRUE)

## Principal coordinate analysis of distance matrices + visualization
distpcoa<-dist.PCoA(dist.tab = distGeneUsage, correction = "none")
     # 'groups' data.frame for plot function: in the case, there are no groups:
groups.vec<-unlist(apply(data.frame(clones.ind$unique_CDR3_sequences_AA),1,
          function(x){strsplit(x,split=", ")[[1]]}))
groups.vec<-cbind(groups.vec, 1)
plotDistPCoA(pcoa.tab = distpcoa, groups=groups.vec, axes = c(1,2),
     plotCorrection = FALSE, title = NULL, plotLegend=TRUE, PDF = "TEST")


## End(Not run)
```

`aaDistribution` *Amino acid distribution of sequences*

Description

This function calculates the amino acid distribution of sequences. Distribution is calculated for sequences of the same length and therein for each position.

`aaDistribution` returns a list containing either only amino acid distribution or a list containing amino acid distribution and analyzed number of sequences per length.

`plotAADistribution` visualizes the amino acid distribution of sequences of the same length.

Usage

```
aaDistribution(sequences = NULL, numberSeq = FALSE)

plotAADistribution(aaDistribution.tab=NULL, plotSeqN=FALSE,
     colors=NULL, PDF=NULL, ...)
```

Arguments

`sequences`	A vector containing amino acid sequences.
`numberSeq`	TRUE: table containing number of sequences will be returned, as well (default: FALSE).
`aaDistribution.tab`	Output list of function `aaDistribution()`
`plotSeqN`	TRUE: Number of sequences for each length will be plotted (see Details; default: FALSE).

colors	Colors to be used for figure containing number of sequences (default: rainbow)
PDF	PDF project name (see Details)
...	

Details

The vector containing sequences will be divided in sequences of the same length and then amino acid distribution for each position is analyzed.

If numberSeq = T, the number of sequences used for the analysis of sequences of the same length will be returned, as well. This information is also required for plotAADistribution(...,plotSeqN = T). Sequence numbers equal to 0 are not plotted; the smallest number is 1.

The PDF character string should be only the project name (without ".pdf"). If plotAADistr = T a figure called "PDF"_Amino-acid-distribution.pdf will be saved to the working directory. If plotSeqN = T a figure called "PDF"_Number-of-sequences.pdf will be saved, as well.

Value

Output is a list containing

Amino_acid_distribution
: list contains data frames of amino acid distributions (including stop codons "*") for each length

Number_of_sequences_per_length
: data frame contains the number of sequences for each length, used for analysis (optional)

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

See Also

trueDiversity

Examples

```
data(aaseqtab)
aadistr<-aaDistribution(sequences = aaseqtab$CDR3_IMGT, numberSeq = TRUE)
## Not run: plotAADistribution(aaDistribution.tab=aadistr, plotAADistr=TRUE, plotSeqN=FALSE,
     PDF="test")
## End(Not run)
```

aaseqtab *Summary of amino acid sequences of B cells of one individual*

Description

Data frame represents an output from IMGT/HighV-QUEST, file 5_AA-sequences(...).txt

Usage

```
data("aaseqtab")
```

Format

A data frame with 3000 observations on the following 18 variables.

`Sequence_number` a numeric vector

`Sequence_ID` a character vector

`Functionality` a character vector

`V_GENE_and_allele` a character vector

`J_GENE_and_allele` a character vector

`D_GENE_and_allele` a character vector

`V_D_J_REGION` a character vector

`V_J_REGION` a character vector

`V_REGION` a character vector

`FR1_IMGT` a character vector

`CDR1_IMGT` a character vector

`FR2_IMGT` a character vector

`CDR2_IMGT` a character vector

`FR3_IMGT` a character vector

`CDR3_IMGT` a character vector

`JUNCTION` a character vector

`J_REGION` a character vector

`FR4_IMGT` a character vector

References

Alamyar, E. et al., IMGT/HighV-QUEST: A High-Throughput System and Web Portal for the Analysis of Rearranged Nucleotide Sequences of Antigen Receptors, JOBIM 2010, Paper 63 (2010).

Brochet, X. et al., Nucl. Acids Res. 36, W503-508 (2008). PMID: 18503082

IMGT/HighV-QUEST 5_AA-sequences file description: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#EAA

Examples

```
data(aaseqtab)
str(aaseqtab)
```

---

aaseqtab2 *Summary of amino acid sequences of B cells of another individual*

Description

Data frame represents an output from IMGT/HighV-QUEST, file 5_AA-sequences(...).txt

Usage

```
data("aaseqtab2")
```

Format

A data frame with 3000 observations on the following 18 variables.

`Sequence_number` a numeric vector

`Sequence_ID` a character vector

`Functionality` a character vector

`V_GENE_and_allele` a character vector

`J_GENE_and_allele` a character vector

`D_GENE_and_allele` a character vector

`V_D_J_REGION` a character vector

`V_J_REGION` a character vector

`V_REGION` a character vector

`FR1_IMGT` a character vector

`CDR1_IMGT` a character vector

`FR2_IMGT` a character vector

`CDR2_IMGT` a character vector

`FR3_IMGT` a character vector

`CDR3_IMGT` a character vector

`JUNCTION` a character vector

`J_REGION` a character vector

`FR4_IMGT` a character vector

References

Alamyar, E. et al., IMGT/HighV-QUEST: A High-Throughput System and Web Portal for the Analysis of Rearranged Nucleotide Sequences of Antigen Receptors, JOBIM 2010, Paper 63 (2010).

Brochet, X. et al., Nucl. Acids Res. 36, W503-508 (2008). PMID: 18503082

IMGT/HighV-QUEST 5_AA-sequences file description: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#EAA

Examples

```
data(aaseqtab2)
str(aaseqtab2)
```

`clones` *Grouping sequences into clones*

Description

This function uses IMGT/HighV-QUEST output files to define B cell clones. Therefore criteria using amino acid CDR3 sequences, V genes and J genes (optional) are used. A treshold for CDR3 identity/similarity can be given. Parallel processing is possible.

Usage

```
clones(aaseqtab = NULL, summarytab = NULL, ntseqtab = NULL, identity = 0.85,
     useJ = TRUE,dispD = FALSE, dispSeqID = FALSE, dispCDR3aa = FALSE,
     dispCDR3nt = FALSE, dispJunctionFr.ratio = FALSE,
     dispJunctionFr.list = FALSE, dispFunctionality.ratio = FALSE,
     dispFunctionality.list = FALSE, dispTotalSeq = FALSE, nrCores=1)
```

Arguments

`aaseqtab`	IMGT/HighV-QUEST output, file 5_AA-sequences(...).txt
`summarytab`	IMGT/HighV-QUEST output, file 1_Summary(...).txt
`ntseqtab`	IMGT/HighV-QUEST output, file 3_Nt-sequences(...).txt (optional)
`identity`	Treshold of CDR3 identity. A value between 0 and 1.
`useJ`	Shall J genes be included into analysis? default: TRUE
`dispD`	Shall D genes and alleles be returned? default: FALSE
`dispSeqID`	Shall sequence ID's be returned? default: FALSE
`dispCDR3aa`	Shall amino acid CDR3 sequences be returned? default: FALSE
`dispCDR3nt`	Shall nucleotide amino acid sequences be returned? default: FALSE
`dispJunctionFr.ratio`	Shall ratios of in-frame, out-of-frame and unknown junctions be returned? default: FALSE

`dispJunctionFr.list`
: Shall a list of all junction frames be returned? default: FALSE

`dispFunctionality.ratio`
: Shall ratios of productive, unproductive and unknown functionality sequences be returned? default: FALSE

`dispFunctionality.list`
: Shall a list of all functionalities be returned? default: FALSE

`dispTotalSeq`
: Shall all total nucleotide sequences be returned? default: FALSE

`nrCores`
: Number of cores used for parallel processing (default: 1)

Details

This function uses IMGT/HighV-QUEST output to define clones. Therefore amino acid CDR3 sequences, V genes and J genes (optional) are used. Criteria for clone groups are 1) same CDR3 length, 2) CDR3 identity of a given treshold, 3) same V gene and 4) same J gene (optional). A treshold for CDR3 identity has to be between 0 and 1. A cutoff of 0.85 means CDR3 identity of 85%. For example for a CDR3 length of 15 amino acids 85% identity would mean that at least 11 of 15 positions have to be identical (0.85*15 = 10.75; values are rounded).

`useJ=T` includes also the criteria of same J genes for clone defintion.

Important to know: - if `useJ=T`, sequences having no J information are ignored

Value

Output of clones() is a data frame containing

`unique_CDR3_sequences_[AA]`
: unique CDR3 sequences belonging to this clone

`CDR3_length_AA`
: CDR3 length in amino acids

`number_of_unique_sequences`
: number of unique CDR3 sequences belonging to this clone

`total_number_of_sequences`
: number of all sequences belonging to this clone (one sequence can appear several times)

`sequence_count_per_CDR3`
: sequence count for each of the unique CDR3 sequences

`V_gene`
: V gene belonging to this clone

`V_gene_and_allele`
: original IMGT V gene nomenclature

`J_gene`
: J gene(s) belonging to this clone (if useJ=F, there can be several J genes)

`J_gene_and_allele`
: original IMGT J gene nomenclature

`optional arguments`
: D_gene;_all_CDR3_sequences_AA; all_CDR3_sequences_nt; Funct_all_sequences; Funct_productive/unproductive/unknown sequences; Junction_frame_all_sequences; JF_in-frame/out-of-frame/unknown sequences; Sequence_IDs; Total_sequences_nt

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

See Also

clones.CDR3Length, plotClonesCDR3Length, plotClonesCopyNumber, geneUsage, plotGeneUsage, clones.shared

Examples

```
## Not run:
data(summarytab)
data(aaseqtab)

clones.tab<-clones(aaseqtab=aaseqtab,summarytab=summarytab, identity=0.85, useJ=TRUE,
     dispCDR3aa=TRUE, dispFunctionality.ratio=TRUE, dispFunctionality.list=TRUE)

## End(Not run)
```

---

clones.allind *B cell clones of 8 individuals*

Description

This data frame represents the combined ouput of the function clones() for 7 individuals (in total 2948 clones).

Usage

```
data("clones.allind")
```

Format

A data frame with 5286 observations on the following 15 variables.

`samples` a factor with 8 levels

`unique_CDR3_sequences_AA` a character vector

`CDR3_length_AA` a numeric vector

`number_of_unique_sequences` a numeric vector

`total_number_of_sequences` a numeric vector

`sequence_count_per_CDR3` a character vector

`V_gene` a character vector

V_gene_and_allele a character vector

J_gene a character vector

J_gene_and_allele a character vector

D_gene a character vector

all_CDR3_sequences_AA a character vector

all_CDR3_sequences_nt a character vector

Functionality_all_sequences a character vector

Junction_frame_all_sequences a character vector

Examples

```
data(clones.allind)
str(clones.allind)
boxplot(clones.allind$CDR3_length_AA ~ clones.allind$samples,xlab="sample",
     ylab="CDR3 length [AA]",xaxt="n",main="CDR3 length distribution of clones", las=3)
axis(1,at=seq(1,length(unique(clones.allind$samples)),1),
     unique(clones.allind$samples))
```

---

clones.CDR3Length *CDR3 length distribution of clones*

Description

This function gives information about the CDR3 length distribution of clones. Results can be returned as relative or absolute values and be visualized as a barplot, using plotClonesCDR3Length.

Usage

```
clones.CDR3Length(CDR3Length = NULL, functionality = NULL, junctionFr = NULL,
     abundance=c("relative","absolute"), ...)

plotClonesCDR3Length(CDR3Length=NULL,functionality=NULL, junctionFr=NULL,
     color=c("orange","darkblue","gray"), abundance=c("relative","absolute"),
     title=NULL, PDF=NULL,...)
```

Arguments

CDR3Length	Vector of CDR3 length of clones (amino acids or nucleotides)
functionality	Vector of functionality of clones (same order than CDR3Length)
junctionFr	Vector of junction frame usage of clones (same order than CDR3Length)
abundance	Shall relative or absolute values be returned? (default: relative)
color	color used for plots (default: c("orange","darkblue","gray"))
title	title of the plot (optional)
PDF	PDF project name (see Details)
...	

Details

The PDF character string should be only the project name (without ".pdf"). For simple analysis of the CDR3 length, a plot called "PDF"_CDR3-length.pdf will be saved to the working direktory.

If CDR3Length.Func = T a figure containing CDR3 length vs. functionality is saved (called "PDF"_CDR3-length_vs_Functionality.pdf) and if CDR3Length.JunctionFr = T a figure containing CDR3 length vs. junction fram usage (called "PDF"_CDR3-length_vs_Junction-frame.pdf) is saved.

Value

Output is a list containing

CDR3_length	relative or absolute abundances of CDR3 length
CDR3_length_vs_functionality	(optional)
CDR3_length_vs_junction_frame	(optional)

Author(s)

Julia Bischof

See Also

clones.CDR3Length, plotClonesCDR3Length, plotClonesCopyNumber, geneUsage

Examples

```
## Not run: data(clones.ind)
clones.CDR3Length(CDR3Length = clones.ind$CDR3_length_AA,
     functionality = clones.ind$Functionality_all_sequences)
plotClonesCDR3Length(CDR3Length = clones.ind$CDR3_length_AA,
     functionality = clones.ind$Functionality_all_sequences)
## End(Not run)
```

---

clones.filterFunctionality

Filter for productive/unproductive clones

Description

Clones containing only productive or only unproductive sequences are filtered.

Usage

```
clones.filterFunctionality(clones.tab = NULL,
     filter = c("productive", "unproductive"))
```

Arguments

clones.tab	A data frame containing clones and their characteristics
filter	Filter feature

Value

Output is a data frame, that contains clones with only productive or unproductive sequences.

Author(s)

Julia Bischof

References

IMGT/LIGM-DB labels: http://www.imgt.org/ligmdb/label

IMGT/HighV-QUEST definition of functionality: http://www.imgt.org/IMGTScientificChart/SequenceDescription/IMGTfunctionality.html#func

See Also

clones.filterJunctionFrame, clones.filterSize

Examples

```
data(clones.ind)

productiveClones<-clones.filterFunctionality(clones.tab = clones.ind,
     filter = "productive")
```

---

clones.filterJunctionFrame

Filter for in-frame/out-of-frame clones

Description

Clones containing only in-frame or only out-of-frame sequences can be filtered.

Usage

```
clones.filterJunctionFrame(clones.tab = NULL,
     filter = c("in-frame", "out-of-frame"))
```

Arguments

clones.tab	A data frame containing clones and their characteristics
filter	Filter feature

Value

Output is a data frame, that contains clones with only in-frame or out-of-frame sequences.

Author(s)

Julia Bischof

References

IMGT/LIGM-DB labels: http://www.imgt.org/ligmdb/label

IMGT/HighV-QUEST definition of functionality (and junction frame): http://www.imgt.org/IMGTScientificChart/SequenceDescription/IMGTfunctionality.html#func

See Also

clones.filterFunctionality, clones.filterSize

Examples

```
data(clones.ind)

inFrameClones<-clones.filterJunctionFrame(clones.tab = clones.ind,
     filter = "in-frame")
```

---

clones.filterSize *Filter clones for their size*

Description

This function filters clones for their size (clone copy number). It can be filtered for different kinds of tresholds (see Details).

Usage

```
clones.filterSize(clones.tab = NULL, column = NULL, number = NULL,
     propOfClones = NULL, propOfSequences = NULL,
     method = c("two.tailed", "upper.tail", "lower.tail"))
```

Arguments

clones.tab	A data frame containing clones and their characteristics
column	Name or number of column to be filtered
number	An integer, giving treshold for size
propOfClones	A proportion between 0 and 1, giving proportion of all clones (see Details)
propOfSequences	A proportion between 0 and 1, giving proportion of all sequences (see Details)
method	Filter method, either smallest and biggest numbers, only smallest or only biggest numbers (see Details)

Details

This function filters clones for their size (clone copy number; total number of sequences belonging to a clone). It can be filtered for

1) a given number of sequences, e.g. `number=20`: the 20 biggest and/or 20 smallest clones,

2) a proportion of all clones, e.g. `propOfClones=0.2`: the 20% biggest and/or smallest clones,

3) a proportion of all sequences, e.g. `propOfSequences=0.01`: Clones, were more and/or less than 1% of all sequences are included.

Only of these criteria is required.

`Method` determines which clones shall be returned: 1) biggest and smallest clones (`two.tailed`), 2) biggest clones (`upper.tail`) or 3) smallest clones (`lower.tail`). In case of `two.tailed` a list containing upper and lower tail will be returned.

Value

Ouput is a data frame (`upper.tail, lower.tail`) or a list (`two.tailed`), containing filtered clones.

Author(s)

Julia Bischof

See Also

`clones.filterFunctionality`, `clones.filterJunctionFrame`

Examples

```
data(clones.ind)

clones.filtered1<-clones.filterSize(clones.tab=clones.ind,
     column="total_number_of_sequences", number=20, method="two.tailed")
clones.filtered2<-clones.filterSize(clones.tab=clones.ind, column=4, propOfClones=0.1,
     method="upper.tail")
clones.filtered3<-clones.filterSize(clones.tab=clones.ind, column=4,
     propOfSequences=0.02, method="lower.tail")
```

---

clones.giniIndex *Gini index of clones*

Description

This function calculates the Gini index of clones.

Usage

```
clones.giniIndex(clone.size = NULL, PDF = NULL)
```

Arguments

clone.size	Vector containing clone sizes
PDF	PDF project name (see Details)

Details

This function calculates the Gini index of clones. Input is a vector containing clone sizes (copy number). The Gini index measures the inequality of clone size distribution. It's between 0 and 1. An index of 0 represents a polyclonal distribution, where all clones have same size. An index of 1 represents a perfect monoclonal distribution.

The PDF character string should be only the project name (without ".pdf"). A figure called "PDF"_Lorenz-curve.pdf will be saved to the working directory. The Lorenz curve represents the clone distribution.

Author(s)

Julia Bischof

References

Cowell, F. A. (2000): Measurement of Inequality. in A B Atkinson/F Bourguignon (Eds): Handbook of Income Distribution, Amsterdam

Gastwirth, J. L. (1972): The Estimation of the Lorenz Curve and Gini Index. The Review of Economics and Statistics 54 (3): 306-316

Lorenz, M. O. (1905): Methods of measuring the concentration of wealth. Publications of the American Statistical Association 9 (70): 209-219

Zeileis, A. (2014): ineq: Measuring Inequality, Concentration, and Poverty. R package version 0.2-13. http://CRAN.R-project.org/package=ineq

Examples

```
data(clones.ind)

gini<-clones.giniIndex(clone.size=clones.ind$total_number_of_sequences, PDF = "Example")
```

---

clones.IDlist *Match sequence ID's and clone ID's*

Description

Get information about which sequences belong to the same clone (see details). This analysis can only be performed, if sequence ID's are returned in table containing clones (clones(..., dispSeqID = T)).

Usage

```
clones.IDlist(clones.seqID = NULL, summarytab.seqID = NULL)
```

Arguments

clones.seqID	A vector containing the column "Sequence_IDs" of the output of function clones()
summarytab.seqID	A vector containing the column "Sequence_ID" of 1_Summary(...).txt IMGT table

Details

This function returns information about each sequence given in den IMGT tables and the corresponding clone. This analysis can only be performed, if sequence ID's are returned in table containing clones (clones(..., dispSeqID = T)).

There are two columns: Sequence_ID and Clone_ID, where Sequence_ID is the same ID given in IMGT tables and Clone_ID is a ID representing all the given clones (clone_1, ..., clone_n, where n is the number of clones; "no_clone" represents the sequences, which belong to none of the clones [probably due to too small abundances]).

If summarytab.seqID is provided, the output table will be ordered like the sequences of the IMGT tables; including those sequences, which belong to no clone. If not, sequences are ordered for sequence ID of the clones.seqID vector.

Author(s)

Julia Bischof

See Also

clones

Examples

```
## Not run:
data(summarytab)
data(aaseqtab)
clone.table<-clones(summarytab = summarytab, aaseqtab = aaseqtab, useJ = T,
                        dispSeqID = T)

clone.ID<-clones.IDlist(clones.seqID = clone.table$Sequence_IDs,
                        summarytab.seqID = summarytab$Sequence_ID)

## End(Not run)
```

---

clones.ind *Data frame: clones of one individual*

Description

This data frame represents the ouput of the function clones() for one individual (1000 clones).

Usage

```
data("clones.ind")
```

Format

A data frame with 1000 observations on the following 14 variables.

`unique_CDR3_sequences_AA` a character vector

`CDR3_length_AA` a numeric vector

`number_of_unique_sequences` a numeric vector

`total_number_of_sequences` a numeric vector

`sequence_count_per_CDR3` a character vector

`V_gene` a character vector

`V_gene_and_allele` a character vector

`J_gene` a character vector

`J_gene_and_allele` a character vector

`D_gene` a character vector

`all_CDR3_sequences_AA` a character vector

`all_CDR3_sequences_nt` a character vector

`Functionality_all_sequences` a character vector

`Junction_frame_all_sequences` a character vector

Examples

```
data(clones.ind)
str(clones.ind)
boxplot(clones.ind$CDR3_length_AA,ylab="CDR3 length [AA]",xaxt="n",
     main="CDR3 length distribution of clones")
```

---

clones.shared *Shared clones between samples*

Description

`clones.shared` compares clones of different samples to find shared clones with identical or similar CDR3 sequences. Criteria for same/similar clones are 1) same CDR3 length, 2) CDR3 identity of a given treshold, 3) same V gene, 4) same J gene (optional). Parallel processing is possible.

`clones.shared.summary` provides information about the number of shared clones between two or more samples.

Usage

```
clones.shared(clones.tab = NULL, identity = 0.85, useJ = TRUE, dispD = TRUE,
     dispCDR3aa = FALSE, dispCDR3nt = FALSE, dispFunctionality.list = FALSE,
     dispFunctionality.ratio = FALSE, dispJunctionFr.list = FALSE,
     dispJunctionFr.ratio = FALSE, dispTotalSeq = FALSE, nrCores=1)

clones.shared.summary(shared.tab = NULL, clones.tab = NULL)
```

Arguments

clones.tab	A dataframe, which includes all individual clones. Therefore data frames, as output from clones() has to be combined, via rbind(). The first column needs to include individual identifiers. Example: data(ClonesAllInd)
identity	A Value between 0 and 1, indicating proportion of identity of sequences.
useJ	TRUE: J gene shall be included as criteria (default: TRUE)
dispD	TRUE: return D gene and allele in output data frame (only, if they are included in input; default: TRUE)
dispCDR3aa	TRUE: return CDR3 amino acid sequences in output data frame (only, if they are included in input; default: FALSE)
dispCDR3nt	TRUE: return CDR3 nucleotide sequences in output data frame (only, if they are included in input; default: FALSE)
dispFunctionality.list	TRUE: return functionality list in output data frame (only, if they are included in input; default: FALSE)
dispFunctionality.ratio	TRUE: return functionality ratios for prosuctive and unproductive sequences in output data frame (only, if they are included in input; default: FALSE)
dispJunctionFr.list	TRUE: return junction frame list in output data frame (only, if they are included in input; default: FALSE)
dispJunctionFr.ratio	TRUE: return junction frame ratios for in-frame and out-of-frame sequences output data frame (only, if they are included in input; default: FALSE)
dispTotalSeq	TRUE: return total nucleotide sequences output data frame (only, if they are included in input; default: FALSE)
nrCores	Number of cores used for parallel processing (default: 1)
shared.tab	Output from function clones.shared(). Either whole data frame or individual column.

Details

clones.shared: Criteria for clone groups are 1) same CDR3 length, 2) CDR3 identity of a given treshold, 3) same V gene and 4) same J gene (optional). Same or similar CDR3 sequences has to be shared between at least two samples. A treshold for CDR3 identity has to be between 0 and 1. A cutoff of 0.85 means CDR3 identity of 85%. For example for a CDR3 length of 15 amino acids

85% identity would mean that at least 11 of 15 positions have to be identical (0.85*15 = 10.75; values are rounded).

`useJ=T` includes also the criteria of same J genes for clone defintion.

`clones.shared.summary` summarizes information about shared clones. If `clonestab.individual` is also provided, number of clones, appearing in only one of these individuals is also returned.

Value

Output of `clones.shared` is a data frame including

`number_samples` The number of samples, that share this clones

`samples` Sample ID's, seperated by ";"

`CDR3_length_AA` length of CDR3 amino acid sequence

`shared_CDR3` shared (100% identity) or similar (<100% identity) CDR3 sequence

`number_shared_CDR3`
Number of shared CDR3 sequences

`unique_CDR3_sequences_AA_per_individual`
CDR3 sequences per clone, individuals are seperated by ";"

`sequence_count_per_CDR3`
Quantity how often clones of "unique all CDR3 sequences [AA] per individual" appaer, individuals are seperated by ";"

`V_gene` V gene

`J_gene` J gene(s)

`optional output`
D_gene; all_CDR3_sequences_AA; al_CDR3_sequences_nt; Funct_all_sequences; Funct_productive/unproductive; Junction_frame_all_sequences; JF_in-frame/out-of-frame; Total_sequences_nt (individuals are seperated by ";")

Output of `clones.shared.summary` is a data frame containing all possible groups and the quantity of appearance.

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

See Also

`clones`

Examples

```
## Not run:
data(clones.allind)
sharedclones<-clones.shared(clones.tab = clones.allind, identity = 0.85, useJ = TRUE,
     dispD = TRUE, dispCDR3aa = TRUE)
sharedclones.summary<-clones.shared.summary(shared.tab = sharedclones)

## End(Not run)
```

`combineIMGT` *Combination of several IMGT output folders*

Description

This function combines several IMGT ouput folders. IMGT/HighV-QUEST can analyse data with up to 500.000 sequences. In case one wants to analyse more than 500.000 sequences, FASTA file have to be splitted into smaller files and have to be analysed individually. Afterwards IMGT output folders can be combined using this function.

Usage

```
combineIMGT(folders = NULL, name = NULL)
```

Arguments

`folders`	A list containing folder names
`name`	Name of the new (combined) project

Value

Output is a folder containing the 10 combined output files (no 11_Parameters.txt file).

Author(s)

Julia Bischof

Examples

```
## Not run:
## Combine folders IMGT1a, IMGT1b, IMGT1c to one folder named "NewProject"
combineIMGT(folders = c("pathTo/IMGT1a", "pathTo/IMGT1b", "pathTo/IMGT1c"),
name = "NewProject)

## End(Not run)
```

compare.aaDistribution *Compare amino acid distribution of different samples*

Description

This function compares the amino acid distribution of different samples. Sequences of the same length are clustered and analyzed. Additionally the number of sequences for each sample can be returned. Parallel processing is possible.

Usage

```
compare.aaDistribution(sequence.list = NULL, names = NULL, numberSeq = FALSE,
     nrCores = 1)

plotCompareAADistribution(comp.tab = NULL, plotSeqN = FALSE, colors = NULL,
     title = NULL, PDF = NULL)
```

Arguments

sequence.list	A list containing vectors of amino acid sequences of each sample
names	A vector containing names for the samples (default: Sample1, Sample2, ...)
numberSeq	Shall number of sequences used for analysis be returned? (default: FALSE)
nrCores	Number of cores used for parallel processing
comp.tab	Output tab from compare.aaDistribution()
plotSeqN	Shall number of sequences used for analysis be plotted? (only possible, if in compare.aaDistribution(...,numberSeq=TRUE,...) is used; default: FALSE)
colors	colors used for individuals in sequence number plot (default: rainbow)
title	Title of plot
PDF	PDF project name (see Details)

Details

Amino acid distribution for each individual is analyzed. Sequenzes of the same length are clustered and amino acid distribution for each position is measured.

The PDF character string should be only the project name (without ".pdf").

A figure called "PDF"_Comparison_Amino-acid-distribution.pdf will be saved to the working directory.

Optional another figure called PDF"_Comparison_Number-of-sequences.pdf" will be returned.

Value

Output is a list containing amino acid distributions for each sequence length and each sample.

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

See Also

compare.aaDistribution, plotCompareAADistribution, aaDistribution

Examples

```
data(aaseqtab)
data(aaseqtab2)

AAdistr.comp<-compare.aaDistribution(sequence.list = list(aaseqtab$CDR3_IMGT,
     aaseqtab2$CDR3_IMGT), names = c("IndA", "IndB"), numberSeq = FALSE, nrCores = 1)
## Not run:
plotCompareAADistribution(comp.tab = AAdistr.comp, plotSeqN = FALSE, PDF = "Example")

## End(Not run)
```

---

compare.geneUsage *Compare gene usage of different samples*

Description

This function compares the gene usage of different samples (see details!). Analysis can be done for subgroups, genes and alleles (see Details). Values can be returned as relative or absolute abundance. Parallel processing is possible.

Usage

```
compare.geneUsage(gene.list = NULL, level = c("subgroup", "gene", "allele"),
     abundance = c("relative", "absolute"), names = NULL, nrCores = 1)

plotCompareGeneUsage(comp.tab = NULL, color = c("gray97", "darkblue"),
     title = NULL, PDF = NULL)
```

Arguments

gene.list	A list containing vectors of genes of each sample
level	Gene level used for gene usage analysis: subgroup, gene, allele
abundance	Shall relative or absolute values be returned? (default: relative)
names	A vector containing names for the samples (default: like Sample1, Sample2, ...)
nrCores	Number of cores used for parallel processing

comp.tab	Output tab from compare.geneUsage()
color	Colors used for heatmap (default: gray and darkblue)
title	Title of plot
PDF	PDF project name (see Details)

Details

Gene usage analysis will be done for each sample. Vector of genes will be analyzed for one of the levels subgroup (e.g. IGHV1), gene (e.g. IGHV1-1) or allele (e.g. IGHV1-1*2). Proportions (abundance = "relative") are always based on the number of all alleles found in list: the number of the subgroup/gene/allele is divided by the number of all alleles mentioned for all sequences (in the case there are more alleles/genes mentioned for one sequence).

The PDF character string should be only the project name (without ".pdf").

A figure called "PDF"_Comparison_Gene-usage.pdf will be saved to the working directory.

Value

Output is a data frame containing absolue or relative values of gene usage of each sample.

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

See Also

geneUsage, compare.geneUsage, plotCompareGeneUsage

Examples

```
data(aaseqtab)
data(aaseqtab2)

Vgenes.comp<-compare.geneUsage(gene.list = list(aaseqtab$V_GENE_and_allele,
     aaseqtab2$V_GENE_and_allele), level = "subgroup", abundance = "relative",
     names = c("IndA", "IndB"), nrCores = 1)
## Not run:
plotCompareGeneUsage(comp.tab = Vgenes.comp, color = c("gray97", "darkblue"), PDF = "Example")

## End(Not run)
```

`compare.trueDiversity` *Compare richness or diversity of different samples*

Description

This function compares the richness or diversity of different samples. Analysis can be done for order 0 (richness), 1 (Shannon) and 2 (Simpson) (see Details). Parallel processing is possible.

Usage

```
compare.trueDiversity(sequence.list = NULL, comp.aaDistribution.tab = NULL,
     order = c(0, 1, 2), names = NULL, nrCores = 1)

plotCompareTrueDiversity(comp.tab = NULL, mean.plot=T, colors = NULL,
    title = NULL, PDF = NULL)
```

Arguments

`sequence.list`	A list containing vectors of amino acid sequences of each sample (see Details)
`comp.aaDistribution.tab`	Output from `compare.aaDistribution()` (see Details)
`order`	True diversity order (q). Values: 0, 1, 2
`names`	A vector containing names for the samples (default: like Sample1, Sample2, ...)
`nrCores`	Number of cores used for parallel processing
`comp.tab`	Output tab from `compare.trueDiversity()`
`mean.plot`	Includes only one figure with mean diversities (default = T)
`colors`	Colors used for individuals (default: rainbow)
`title`	Title of plot
`PDF`	PDF project name (see Details)

Details

This functions needs either a list containing vectors of sequences or the output of

`compare.aaDistribution()` as input. In first case `compare.aaDistribution()` is first applied to data set. Richness or diversity is calculated for sequences of the same length, for each position. Analysis of true diversity of order `0`, `1` and `2` is possible. Order 0: Richness (in this case it represents number of different amino acids per position). Order 1: Exponential function of Shannon entropy using the natural logarithm (weights all amino acids by their frequency). Order 2: Inverse Simpson entropy (weights all amino acids by their frequency, but weights are given more to abundant amino acids). These indices are very similar (Hill, 1973). For example the exponential function of Shannon index is linearly related to inverse Simpson.

`plotCompareTrueDiversity` returns an image with diversity plots for each length, if `mean.plot = F`. In the case of `mean.plot = T`, only one figure is returned, where mean diversity values for each

sequence length are plotted. Each plot contains the richness or diversity (y-axis) for each position (x-axis). Individuals are color coded.

The PDF character string should be only the project name (without ".pdf").

A figure called "PDF"_Comparison_True-diversity_q"order".pdf will be saved to the working directory.

Value

Output is a list containing 1) the diversity order, and 2) diversity values for each individual and each sequence length.

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

References

M. O. Hill: Diversity and Evenness: A Unifying Notation and Its Consequences; Ecology 54:2, p 427-432 (1973)

Lou Jost: Entropy and diversity; OIKOS 113:2 (2006)

Jari Oksanen, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens and Helene Wagner (2015). vegan: Community Ecology Package. R package version 2.3-0. http://CRAN.R-project.org/package=vegan

See Also

compare.trueDiversity, plotCompareTrueDiversity, trueDiversity, compare.aaDistribution

Examples

```
data(aaseqtab)
data(aaseqtab2)

trueDiv.comp<-compare.trueDiversity(sequence.list = list(aaseqtab$CDR3_IMGT,
     aaseqtab2$CDR3_IMGT), names = c("IndA", "IndB"), order = 1, nrCores = 1)
## Not run:
plotCompareTrueDiversity(comp.tab = trueDiv.comp, PDF = "Example")

## End(Not run)
```

`dist.PCoA` *Principal coordinate analysis (PCoA; multidimensional scaling [MDS]) of dissimilarity/distance indices*

Description

Performs and plots a principal coordinate analysis (PCoA) of dissimilarity/distance indices. Correction methods can be used. Merging of samples to groups is possible in the plot function.

Usage

```
dist.PCoA(dist.tab = NULL, correction = c("lingoes", "cailliez", "none"))

plotDistPCoA(pcoa.tab = NULL, groups = NULL, names = NULL, axes = NULL,
     plotCorrection = FALSE, title = NULL, plotLegend=FALSE, PDF = NULL)
```

Arguments

`dist.tab`	Dissimilarity/distance matrix (e.g. from `sequences.distance()`)
`correction`	Correction method of PCoA: Lingoes, Cailliez or none
`pcoa.tab`	PCoA outout from `dist.PCoA()`
`groups`	data.frame containing sequences (1. column) and groups (2. column)
`names`	Names of samples/axes
`axes`	Which axes shall be plotted? e.g. c(1,2) for axes 1 and 2
`plotCorrection`	Shall corrected or uncorrected eigenvalues be plotted?
`title`	Title of the plot
`plotLegend`	Shall legend be plotted?
`PDF`	PDF project name (see Details)

Details

This function provides a PCoA object for dissimilarity indices/distances as input (e.g. from functions `sequences.distance` or `geneUsage.distance()`). For further details of pcoa see `pcoa`.

The plot function provides a figure with the principal coordinates with positive eigenvalues (in the case of no correction) or the principal coordinates with positive eigenvalues from the distance matrix corrected using the specified correction method. The principal coordinates correspond to the specified axes.

A figure called "PDF"_PCoA.pdf will be saved to the working directory.

Value

Output is an PCoA object, see `pcoa`.

Author(s)

Julia Bischof

References

Paradis E., Claude J. & Strimmer K. 2004. APE: analyses of phylogenetics and evolution in R language. Bioinformatics 20: 289-290.

See Also

dist.PCoA, plotDistPCoA, sequences.distance, geneUsage.distance, pcoa

Examples

```
## Not run:
data(clones.ind)
seq.dist<-sequences.distance(sequences = clones.ind$unique_CDR3_sequences_AA,
     method = "levenshtein", divLength=F)
distpcoa<-dist.PCoA(dist.tab = seq.dist, correction = "none")

# 'groups' data.frame for plot function: in the case, there are no groups:
groups.vec<-unlist(apply(data.frame(clones.ind$unique_CDR3_sequences_AA),1,
          function(x){strsplit(x,split=", ")[[1]]}))
groups.vec<-cbind(groups.vec, 1)

plotDistPCoA(pcoa.tab = distpcoa, groups=groups.vec, axes = c(1,2),
     plotCorrection = FALSE, title = NULL, plotLegend=T, PDF = "TEST")

## End(Not run)
```

---

geneUsage *Gene usage statistics*

Description

This function gives information about the gene usage distribution (see detail!). Results can returned as relative or absolute values and be visualized as a barplot, using `plotGeneUsage`. Further gene usage can be analyzed in relation to functionality or junction frame usage.

Usage

```
geneUsage(genes = NULL, level = c("subgroup", "gene", "allele"),
     functionality = NULL, junctionFr = NULL,
     abundance=c("relative","absolute"), ...)

plotGeneUsage(geneUsage.tab=NULL,plotFunctionality=FALSE,plotJunctionFr=FALSE,
     color=c("orange","darkblue","gray"), title=NULL, PDF=NULL,...)
```

Arguments

genes	Vector containing gene information (IMGT nomenclature; see Details)
level	Gene level used for gene usage analysis: subgroup, gene, allele
functionality	Vector containing functionality information
junctionFr	Vector containing functionality information
abundance	Shall relative or absolute values be returned? (default: relative)
geneUsage.tab	Output of geneUsage()
plotFunctionality	Shall gene usage vs. functionality be plotted? (default: FALSE)
plotJunctionFr	Shall gene usage vs. junction frame be plotted? (default: FALSE)
color	color used for plots (default: c("orange","darkblue","gray"))
title	title of the plot (optional)
PDF	PDF project name (see Details)
...	

Details

Vector of genes will be analyzed for one of the levels subgroup (e.g. IGHV1), gene (e.g. IGHV1-1) or allele (e.g. IGHV1-1*2). Proportions (abundance = "relative") are always based on the number of all alleles found in list: the number of the subgroup/gene/allele is divided by the number of all alleles mentioned for all sequences (in the case there are more alleles/genes mentioned for one sequence).

The PDF character string should be only the project name (without ".pdf"). For an analysis of the gene usage distribution, a plot called "PDF"_Gene-usage.pdf will be saved to the working direktory.

If geneUsage.Func = T gene usage vs. functionality will be analyzed (and a figure called "PDF"_Gene-usage_vs_Functionality.pdf will be saved) and if geneUsage.JunctionFr = T gene usage vs. junction frame usage will be analyzed (and figure called "PDF"_Gene-usage_vs_Junction-frame.pdf will be saved).

Value

Output is a list containing

gene_usage	relative or absolute abundances of subgroups/genes
gene_usage_vs_functionality	(optional)
gene_usage_vs_junction_frame	(optional)

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

See Also

geneUsage, plotGeneUsage

Examples

```
data(clones.ind)
geneUsage(genes = clones.ind$V_gene, level = "subgroup",
     functionality = clones.ind$Functionality_all_sequences)
## Not run: plotGeneUsage(geneUsage.tab = clones.ind$V_gene,
     plotFunctionality = TRUE)
## End(Not run)
```

---

geneUsage.distance *Dissimilarity/distance indices for gene usage data*

Description

This function calculates Bray-Curtis, Jaccard or cosine indices for gene usage data of different samples.

Usage

```
geneUsage.distance(geneUsage.tab=NULL, names=NULL,
     method=c("bc","jaccard", "cosine"), cutoff=0)
```

Arguments

geneUsage.tab	gene usage table with genes as columns and samples as rows
names	Names of samples (default: Sample1...n)
method	Distance/dissimilarity index to be used for calculation. On of Bray-Curtis (bc), Jaccard (jaccard) or cosine (cosine)
cutoff	Cutoff for gene proportions (default: 0; see details)

Details

This function calculates dissmiliarity indices based on gene usage data of different samples (columns = genes, rows = samples). Bray-Curtis, Jaccard or cosine indices can be chosen.

For explanation of Bray-Curtis and Jaccard index see vegdist.

For explanation of cosine index see dist.

When using Jaccard index, a cutoff for gene proportions can be given. Proportions will be transformed into absence/presence data (<= cutoff; > cutoff) and afterwards Jaccard indices are calculated.

Value

Output is a matrix containing dissimilarity/distance indices between samples, based on gene usages.

Author(s)

Julia Bischof

References

Bray, J. R. and J. T. Curtis. 1957. An ordination of upland forest communities of southern Wisconsin. Ecological Monographs 27:325-349.

Levandowsky, Michael; Winter, David (1971), "Distance between sets", Nature 234 (5): 34-35

Graham L. Giller (2012). "The Statistical Properties of Random Bitstreams and the Sampling Distribution of Cosine Similarity". Giller Investments Research Notes (20121024/1)

Jari Oksanen, F. Guillaume Blanchet, et al. (2015). vegan: Community Ecology Package. R package version 2.3-1. https://CRAN.R-project.org/package=vegan

David Meyer and Christian Buchta (2015). proxy: Distance and Similarity Measures. R package version 0.4-15. https://CRAN.R-project.org/package=proxy

See Also

dist.PCoA, plotDistPCoA, sequences.distance

Examples

```
data(vgenes) # VH gene proportions of 10 samples (rows) and 30 VH genes (columns)
disttab<-geneUsage.distance(geneUsage.tab = vgenes, method = "bc")
```

---

mutationtab *Summary of mutations cells*

Description

Data frame represents an output from IMGT/HighV-QUEST, file 7_V-REGION-mutation-and-AA-change-table(...).txt

Usage

```
data("mutationtab")
```

Format

A data frame with 3000 observations on the following 11 variables.

`Sequence_number` a numeric vector

`Sequence_ID` a character vector

`Functionality` a character vector

`V_GENE_and_allele` a character vector

V_REGION a character vector

FR1_IMGT a character vector

CDR1_IMGT a character vector

FR2_IMGT a character vector

CDR2_IMGT a character vector

FR3_IMGT a character vector

CDR3_IMGT a character vector

References

Alamyar, E. et al., IMGT/HighV-QUEST: A High-Throughput System and Web Portal for the Analysis of Rearranged Nucleotide Sequences of Antigen Receptors, JOBIM 2010, Paper 63 (2010).

Brochet, X. et al., Nucl. Acids Res. 36, W503-508 (2008). PMID: 18503082

IMGT/HighV-QUEST 7_V-REGION-mutation-and-AA-change-table file description: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#Emuttable

Examples

```
data(mutationtab)
str(mutationtab)
```

---

ntseqtab *Summary of nucleotide sequences of B cells of one individual*

Description

Data frame represents an output from IMGT/HighV-QUEST, file 3_Nt-sequences(...).txt

Usage

```
data("ntseqtab")
```

Format

A data frame with 3000 observations on the following 114 variables.

Sequence_number a numeric vector

Sequence_ID a character vector

Functionality a character vector

V_GENE_and_allele a character vector

J_GENE_and_allele a character vector

D_GENE_and_allele a character vector

V_D_J_REGION a character vector

V_J_REGION a character vector

V_REGION a character vector

FR1_IMGT a character vector

CDR1_IMGT a character vector

FR2_IMGT a character vector

CDR2_IMGT a character vector

FR3_IMGT a character vector

CDR3_IMGT a character vector

JUNCTION a character vector

3V_REGION a character vector

(N_D)_J_REGION a character vector

(N_D)_REGION a character vector

P3V a character vector

N_REGION a character vector

N1_REGION a character vector

P5D a character vector

D_REGION a character vector

P3D a character vector

P5D1 a logical vector

D1_REGION a character vector

P3D1 a character vector

N2_REGION a character vector

P5D2 a character vector

D2_REGION a character vector

P3D2 a character vector

N3_REGION a character vector

P5D3 a character vector

D3_REGION a character vector

P3D3 a character vector

N4_REGION a character vector

P5J a character vector

5J_REGION a character vector

D_J_REGION a character vector

J_REGION a character vector

FR4_IMGT a character vector

V_D_J_REGION_start a numeric vector

V_D_J_REGION_end a numeric vector

V_J_REGION_start a numeric vector

V_J_REGION_end a logical vector

V_REGION_start a numeric vector

V_REGION_end a numeric vector

FR1_IMGT_start a numeric vector

FR1_IMGT_end a numeric vector

CDR1_IMGT_start a numeric vector

CDR1_IMGT_end a numeric vector

FR2_IMGT_start a numeric vector

FR2_IMGT_end a numeric vector

CDR2_IMGT_start a numeric vector

CDR2_IMGT_end a numeric vector

FR3_IMGT_start a numeric vector

FR3_IMGT_end a numeric vector

CDR3_IMGT_start a numeric vector

CDR3_IMGT_end a numeric vector

JUNCTION_start a numeric vector

JUNCTION_end a numeric vector

3V_REGION_start a numeric vector

3V_REGION_end a numeric vector

(N_D)_J_REGION_start a numeric vector

(N_D)_J_REGION_end a numeric vector

(N_D)_REGION_start a numeric vector

(N_D)_REGION_end a numeric vector

P3V_start a numeric vector

P3V_end a numeric vector

N_REGION_start a numeric vector

N_REGION_end a numeric vector

N1_REGION_start a numeric vector

N1_REGION_end a numeric vector

P5D_start a numeric vector

P5D_end a numeric vector

D_REGION_start a numeric vector

D_REGION_end a numeric vector

P3D_start a numeric vector

P3D_end a numeric vector

P5D1_start a numeric vector

P5D1_end a numeric vector

D1_REGION_start a numeric vector

D1_REGION_end a numeric vector

P3D1_start a numeric vector

P3D1_end a numeric vector

N2_REGION_start a numeric vector

N2_REGION_end a numeric vector

P5D2_start anumeric vector

P5D2_end a numeric vector

D2_REGION_start a numeric vector

D2_REGION_end a numeric vector

P3D2_start a numeric vector

P3D2_end a numeric vector

N3_REGION_start a numeric vector

N3_REGION_end a numeric vector

P5D3_start a numeric vector

P5D3_end a numeric vector

D3_REGION_start a numeric vector

D3_REGION_end a numeric vector

P3D3_start a numeric vector

P3D3_end a numeric vector

N4_REGION_start a numeric vector

N4_REGION_end a numeric vector

P5J_start a numeric vector

P5J_end a numeric vector

5J_REGION_start a numeric vector

5J_REGION_end a numeric vector

D_J_REGION_start a numeric vector

D_J_REGION_end a numeric vector

J_REGION_start a numeric vector

J_REGION_end a numeric vector

FR4_IMGT_start a numeric vector

FR4_IMGT_end a numeric vector

References

Alamyar, E. et al., IMGT/HighV-QUEST: A High-Throughput System and Web Portal for the Analysis of Rearranged Nucleotide Sequences of Antigen Receptors, JOBIM 2010, Paper 63 (2010).

Brochet, X. et al., Nucl. Acids Res. 36, W503-508 (2008). PMID: 18503082

IMGT/HighV-QUEST 3_Nt-sequences file description: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#Ent

Examples

```
data(ntseqtab)
str(ntseqtab)
```

`plotClonesCopyNumber` *Copy number of clones*

Description

This Function plots the copy number distribution of clones.

Usage

```
plotClonesCopyNumber(copyNumber = NULL, withOutliers = TRUE, color = "gray",
     title = NULL, PDF=NULL, ...)
```

Arguments

`copyNumber`	Vector of copy numbers of clones
`withOutliers`	Shall outliers (anything bigger than 75
`color`	color used for plots (default: gray)
`title`	title of the plot (optional)
`PDF`	PDF project name (see Details)
`...`	

Details

The `PDF` character string should be only the project name (without ".pdf").

A figure called "PDF"_Clone_copy_number.pdf will be saved to the working directory.

Author(s)

Julia Bischof

See Also

`clones.CDR3Length`, `plotClonesCDR3Length`

Examples

```
data(clones.ind)
## Not run: plotClonesCopyNumber(copyNumber = clones.ind$total_number_of_sequences,
     color = "red", title = "Copy number distribution")
## End(Not run)
```

readIMGT *Read IMGT/HighV-QUEST output files*

Description

This function reads IMGT/HighV-QUEST output files into a data frame.

Usage

```
readIMGT(data, filterNoResults = TRUE)
```

Arguments

data	path and file name of IMGT/HighV-QUEST output file
filterNoResults	Shall sequences without any information be excluded? (default: TRUE)

Details

"-" or spaces are be converted to "_" (single quotes are ignored).

Value

Output is a data frame containing IMGT/HighV-QUEST output data.

Author(s)

Julia Bischof

References

IMGT/V-QUEST Documentation: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#output3

Examples

```
## Not run:
tab<-readIMGT("PathTo/file.txt",filterNoResults=TRUE)
str(tab)

## End(Not run)
```

`sequences.distance` *Dissimilarity/distance indices for sequence data*

Description

This function calculates different dissmilarity/distance indices of sequences.

Usage

```
sequences.distance(sequences = NULL, groups = NULL,
     method = c("levenshtein", "cosine", "q-gram", "jaccard", "ja-wi",
                "dam-le", "hamming", "osa", "lcs"), divLength = FALSE)
```

Arguments

`sequences`	Vector containing sequences
`groups`	Vector containing names of different samples (if present)
`method`	Dissmilariy method (see details)
`divLength`	Divide sequences into subsets of the same sequence length? (default: FALSE)

Details

This function calculates dissmiliarity/distance indices based on sequences. Levenshtein, cosine, q-gram, Jaccard, Jaro-Winker (`ja-wi`), Damerau-Levenshtein (`dam-le`), Hamming, Optimal string alignment (`osa`) and longest common substring (`lcs`) distance can be chosen. For details see `stringdist-metrics`.

Value

Output is a distance matrix containing dissimilarity indices/distances between sequences.

Author(s)

Julia Bischof

References

van der Loo M (2014). The stringdist package for approximate string matching. The R Journal, 6, pp. 111-122. https://CRAN.R-project.org/package=stringdist

See Also

`dist.PCoA`, `plotDistPCoA`, `geneUsage.distance`

Examples

```
## Not run:
data(clones.ind)
data(clones.allind)
dist1<-sequences.distance(sequences = clones.ind$unique_CDR3_sequences_AA,
     method = "levenshtein", divLength=TRUE)
dist2<-sequences.distance(sequences = clones.allind$unique_CDR3_sequences_AA,
     groups = clones.allind$individuals, method = "cosine", divLength=FALSE)

## End(Not run)
```

`sequences.functionality`

Summary of functionality of sequences

Description

This function gives information about the proportion of productive and unproductive sequences.

Usage

```
sequences.functionality(data = NULL, relativeValues=TRUE, ...)
```

Arguments

`data`	Vector containing functionality information, e.g. taken from IMGT/HighV-QUEST output, e.g. 1_Summary(...).txt, 3_Nt-sequences(...).txt, 5_AA-sequences(...).txt
`relativeValues`	Shall relative or absolute values be returned? (default: TRUE)
`...`	

Details

Productive sequences: include no stop codon and are in-frame

Unproductive sequences: include a stop codon and/or are out-of-frame

Value

Output is a data frame including proportions of productive and unproductive sequences, as well as proportion of sequences with unknown functionality.

Author(s)

Julia Bischof

References

IMGT/LIGM-DB labels: http://www.imgt.org/ligmdb/label

IMGT/HighV-QUEST definition of functionality: http://www.imgt.org/IMGTScientificChart/SequenceDescription/IMGTfunctionality.html#func

See Also

sequences.junctionFrame

Examples

```
data(summarytab)
funct<-sequences.functionality(data = summarytab$Functionality)
barplot(as.numeric(funct),xlab="",xlim=c(0,2),legend.text = colnames(funct),
     col=c("orange","darkblue","gray"),xaxt="n",main="Functionality",ylab="proportion")
```

---

sequences.geneComb *Gene/gene combinations*

Description

This function calculates the abundance of two paired gene families. Analysis can be done for subgroup, gene and allele level (see Details).

sequences.geneComb returns a data frame containing all possible combinations of gene families 1 (rows) and 2 (columns) in relative or absolute values.

plotGeneComb visualizes relative abundances of gene/gene combinations in a heatmap. Absolute values will be converted to relative abundances.

Usage

```
sequences.geneComb(family1 = NULL, family2 = NULL,
    level = c("subgroup","gene","allele"), abundance = c("relative", "absolute"),
     nrCores=1)

plotGeneComb(geneComb.tab=NULL,color=c("gray97","darkblue"),withNA=TRUE,
     title=NULL, PDF=NULL,...)
```

Arguments

family1	Vector including gene family 1, should be matched to order of family2 (see Details)
family2	Vector including gene family 2, should be matched to order of family1 (see Details)
level	Gene family level: subgroup, gene or allele (see Details)
abundance	How values shall be returned: relative or absolute abundance.

nrCores	Number of cores used for parallel processing (default: 1)
geneComb.tab	Output data frame from sequences.genecomb()
color	colors of heatmap (default: gray and blue)
withNA	Shall combination without information for one of both families be included? (default: TRUE)
title	title of plot (NULL: "family1 & family2 gene combinations (zoom: xx)"; turn of with "")
PDF	PDF project name (see Details)
...	

Details

Input are vectors of IMGT/HighV-QUEST output file 1_Summary(...).txt. Columns like "V-GENE and allele", "D-GENE and allele" or "J-GENE and allele" can be used as input, but need to have the same order! Levels can be subgroup (e.g. IGHV1), genes (e.g. IGHV1-1) or alleles (e.g. IGHV1-1*2).

The PDF character string should be only the project name (without ".pdf").

A figure called "PDF"_"family1"-"family2"-combinations.pdf will be saved to the working directory.

Value

Output is a data frame containing gene/gene combinations, with family 1 as rows and family 2 as columns. Either in absolute or relative abundances.

Note

For large datasets computational time can be extensive.

Author(s)

Julia Bischof

References

IMGT Repertoire (IG and TR): http://www.imgt.org/IMGTrepertoire/LocusGenes/

See Also

sequences.geneComb, plotGeneComb, geneUsage

Examples

```
data(summarytab)
VDcomb.tab<-sequences.geneComb(family1 = summarytab$V_GENE_and_allele,
     family2 = summarytab$D_GENE_and_allele, level = "subgroup", abundance = "relative")

plotGeneComb(geneComb.tab=VDcomb.tab, withNA=FALSE)
```

`sequences.getAnyFunctionality` *Filter for productive or unproductive sequences*

Description

Filter IMGT/HighV-QUEST output files for productive or unproductive sequences.

`sequences.getProductives` filters only productive sequences

`sequences.getUnproductives` filters only unproductive sequences

`sequences.getAnyFunctionality` filters all sequences with any functionality information (productive or unproductive)

Usage

```
sequences.getAnyFunctionality(data = NULL)

sequences.getProductives(data = NULL)

sequences.getUnproductives(data = NULL)
```

Arguments

`data`	IMGT/HighV-QUEST output file with functionality information, e.g. 1_Summary().txt, 3_Nt-sequences.txt, ...

Details

Productive sequences: include no stop codon and are in-frame

Unproductive sequences: include a stop codon and/or are out-of-frame

Value

Output is a data frame with the same columns like the input, but filtered for only productive or unproductive sequences or sequences having any functionality information.

Author(s)

Julia Bischof

References

IMGT/LIGM-DB labels: http://www.imgt.org/ligmdb/label

IMGT/HighV-QUEST definition of functionality: http://www.imgt.org/IMGTScientificChart/SequenceDescription/IMGTfunctionality.html#func

See Also

sequences.getAnyFunctionality, sequences.getProductives, sequences.getUnproductives, sequences.functionality, sequences.getAnyJunctionFrame, sequences.getInFrames, sequences.getOutOfFrames

Examples

```
data(summarytab)
ProductiveSequences<-sequences.getProductives(summarytab)
UnroductiveSequences<-sequences.getUnproductives(summarytab)
AnyFunctSequences<-sequences.getAnyFunctionality(summarytab)
```

---

sequences.getAnyJunctionFrame

Filter for in-frame or out-of-frame sequences

Description

Filter IMGT/HighV-QUEST output files for in-frame or out-of-frame sequences.

`sequences.getInFrames` filters only in-frame sequences

`sequences.getOutOfFrames` filters only out-of-frame sequences

`sequences.getAnyJunctionFrame` filters all sequences with any junction frame information (in-frame or out-of-frame)

Usage

```
sequences.getAnyJunctionFrame(data = NULL)

sequences.getInFrames(data = NULL)

sequences.getOutOfFrames(data = NULL)
```

Arguments

`data`	IMGT/HighV-QUEST output file with junction frame information, e.g. 1_Summary().txt, ...

Value

Output is a data frame with the same columns like the input, but filtered for only in-frame or out-of-frame sequences or sequences having any junction frame information.

Author(s)

Julia Bischof

References

IMGT/LIGM-DB labels: http://www.imgt.org/ligmdb/label

IMGT/HighV-QUEST definition of functionality (and junction frame): http://www.imgt.org/IMGTScientificChart/SequenceDescription/IMGTfunctionality.html#func

See Also

sequences.getAnyJunctionFrame, sequences.getInFrames, sequences.getOutOfFrames, sequences.junctionFrame, sequences.getAnyFunctionality, sequences.getProductives, sequences.getUnproductives

Examples

```
data(summarytab)
InFrameSequences<-sequences.getInFrames(summarytab)
OutOfFrameSequences<-sequences.getOutOfFrames(summarytab)
AnyJunctionFrSequences<-sequences.getAnyJunctionFrame(summarytab)
```

---

sequences.junctionFrame

Summary of junction frames of sequences

Description

This function gives information about the proportion of in-frame and out-of-frame sequences.

Usage

```
sequences.junctionFrame(data = NULL, relativeValues=TRUE, ...)
```

Arguments

data	Vector containing junction frame information, e.g. taken from IMGT/HighV-QUEST output 1_Summary(...).txt
relativeValues	Shall relative or absolute values be returned? (default: TRUE)
...	

Details

Junction: coding region of the V-J or V-D-J junction from 2nd CYS (position 104) to J-PHE or J-TRP (position 118) in an IG or TR V-DOMAIN (http://www.imgt.org/ligmdb/label)

Value

Output is a data frame including proportions of in-frame and out-of-frame sequences, as well as proportion of sequences without any information about junction frames.

Author(s)

Julia Bischof

References

IMGT/LIGM-DB labels: http://www.imgt.org/ligmdb/label

IMGT/HighV-QUEST definition of functionality (and junction frame): http://www.imgt.org/IMGTScientificChart/SequenceDescription/IMGTfunctionality.html#func

See Also

sequences.functionality

Examples

```
data(summarytab)
junctionfr<-sequences.junctionFrame(data = summarytab$JUNCTION_frame)
barplot(as.numeric(junctionfr),xlab="",xlim=c(0,2),legend.text = colnames(junctionfr),
    col=c("orange","darkblue","gray"),xaxt="n",main="Junction frame usage",ylab="proportion")
```

---

sequences.mutation *Basic statistics on mutations of sequences*

Description

This function summarizes the number of mutations of sequences. It gives also information about the number of silent and replacement mutations, R/S ratio, as well as mutation numbers, depending on functionality or junction frame distributions.

Usage

```
sequences.mutation(mutationtab = NULL, summarytab = NULL,
    sequence = c("V", "FR1", "FR2", "FR3", "CDR1", "CDR2"), functionality = FALSE,
     junctionFr = FALSE, rsRatio=FALSE,...)
```

Arguments

summarytab	IMGT/HighV-QUEST output 1_Summarytab(...).txt
mutationtab	IMGT/HighV-QUEST output 7_V-REGION-mutation-and-AA-change-table(...).txt
sequence	One of V, FR1, FR2, FR3, CDR1, CDR2
functionality	TRUE: mutation vs. functionality will be returned (default: FALSE)
junctionFr	TRUE: mutation vs. junction frame usage will be returned, summarytab required (default: FALSE)
rsRatio	TRUE: R/S ratio will be returned (default: FALSE)
...	

Details

IMGT/HighV-QUEST output file 7_V-REGION-mutation-and-AA-change-table(...).txt (`mutationtab`) is required as input. 1_Summarytab(...).txt (`summarytab`) is optional; if specified, junction frame information and in special for V sequences "V-REGION identity [nt]" can be returned. Mutations of V region, as well as FR1, 2, 3 and CDR1, 2 can be analyzed. `rsRatio=T` returns the ratio of replacement and silent mutations per sequence. Sequences without silent or replacement mutation, will have a ratio of 0.

Value

Output is a list containing

`Number_of_mutations` data frame with number of total mutations, replacement and silent mutations (optional: V sequences V-REGION identity [nt], R/S ratio)

`Functionality` Proportions of mutations and no mutations in productive and unproductive sequences (optional)

`Junction_frame` Proportions of mutations and no mutations in in-frame and out-of-frame sequences (optional)

`RS_ratio` Ratio of replacement and silent mutations (optional)

Author(s)

Julia Bischof

References

IMGT/HighV-QUEST V-REGION mutation and AA change table: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#mut-table

IMGT Index "Mutation": http://www.imgt.org/IMGTindex/mutation.php

Examples

```
data(mutationtab)
data(summarytab)
V.mutation<-sequences.mutation(mutationtab = mutationtab, summarytab = summarytab,
     sequence = "V", junctionFr = TRUE, rsRatio=TRUE)
CDR1.mutation<-sequences.mutation(mutationtab = mutationtab, sequence = "CDR1",
     functionality=TRUE)
par(mar=c(18,5,5,3))
barplot(as.numeric(CDR1.mutation$Functionality[,1]),
     names=rownames(CDR1.mutation$Functionality),
     ylab="proportion",main="Mutation vs. Functionality",las=3)
```

`sequences.mutation.AA` *Detect amino acid mutations in V-, FR- or CDR-regions*

Description

This functions detects amino acid mutations in V-, FR- or CDR-regions and returns a data frame containing proportions of mutations from amino acid x to amino acid y.

Usage

```
sequences.mutation.AA(mutationtab = NULL,
     sequence = c("V", "FR1", "FR2", "FR3", "CDR1", "CDR2"))
plotSequencesMutationAA(mutationAAtab = NULL,
    showChange = c("no","hydropathy", "volume", "chemical"), title = NULL, PDF = NULL)
```

Arguments

`mutationtab`	IMGT output table 7_V-REGION-mutation-and-AA-change-table(...).txt
`sequence`	Sequence region to be analyzed (V, FR1, FR2, FR3, CDR1, CDR2)
`mutationAAtab`	Output table of sequences.mutation.AA()
`showChange`	Shall hydropathy, volume, chemical or no changes be returned?
`title`	Title of the plot
`PDF`	PDF project name (see Details)

Details

`sequences.mutation.AA()` returns a 20 x 20 data frame containing proportions of amino acid mutations in a given region. Original amino acids are in rows, mutated ones in columns.

`plotSequencesMutationAA()` returns a kind of heatmap containing proportion of amino acid changes and hydropathy, chemical or volume changes. Original amino acids are in rows, mutated ones in columns. The `PDF` character string should be only the project name (without ".pdf"). A figure called "PDF"_AA-mutation.pdf will be saved to the working directory.

Author(s)

Julia Bischof

References

IMGT/HighV-QUEST V-REGION mutation and AA change table: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#mut-table

IMGT Index "Mutation": http://www.imgt.org/IMGTindex/mutation.php

See Also

sequences.mutation.AA, plotSequencesMutationAA,

sequences.mutation, sequences.mutation.base,

plotSequencesMutationBase

Examples

```
data(mutationtab)
V.mut.AA<-sequences.mutation.AA(mutationtab = mutationtab, sequence = "V")
## Not run:
plotSequencesMutationAA(mutationAAtab = V.mut.AA, showChange = "hydropathy")

## End(Not run)
```

---

sequences.mutation.base

Statistics about silent mutations

Description

This function calculates a) proportions of silent mutations from nucleotide A to B and b) proportions of A, C, G and T/U from positions -3 to +3 around a mutation.

Usage

```
sequences.mutation.base(mutationtab = NULL, summarytab = NULL,
     analyseEnvironment = FALSE, analyseMutation = TRUE,
     sequence = c("V", "FR1", "FR2", "FR3", "CDR1", "CDR2"), nrCores = 1)


plotSequencesMutationBase(mutationBaseTab = NULL, plotEnvironment = FALSE,
    plotMutation = TRUE, colHeatmap = c("white","darkblue"), title = NULL, PDF = NULL)
```

Arguments

mutationtab	IMGT output table 7_V-REGION-mutation-and-AA-change-table(...).txt
summarytab	IMGT output table 1_Summary(...).txt
analyseEnvironment	Shall proportions of A, C, G and T/U from positions -3 to +3 around a mutation be analysed? (default: FALSE)
analyseMutation	Shall proportions of mutations from nucleotide A to B be analysed? (default: TRUE)
sequence	Sequence region to be analyzed (V, FR1, FR2, FR3, CDR1, CDR2)

nrCores	Number of cores to be used for parallel processing
mutationBaseTab	Output of sequences.mutation.base()
plotEnvironment	Shall proportions of A, C, G and T/U from positions -3 to +3 around a mutation be plotted? (default: FALSE)
plotMutation	Shall proportions of mutations from nucleotide A to B be plotted? (default: TRUE)
colHeatmap	Colours for the heatmap (default: c("white","darkblue"))
title	Title of the plot
PDF	PDF project name (see Details)

Details

This function calculates a) proportions of silent mutations from nucleotide A to B and b) proportions of A, C, G and T/U from positions -3 to +3 around a mutation. For case a) a matrix containing proportions of mutations is returned. In case b) a data frame containing positions in columns and base changes in rows is returned. Position 0 is the position where mutation occurs. Position -3 to -1 and +1 to +3 show proportions of bases next to the mutated position.

Function `plotSequencesMutationBase()` returns a heatmap or barplots for every base. The `PDF` character string should be only the project name (without ".pdf"). A figure called "PDF"_Base-mutation_mutated-position.pdf" and/or "PDF"_Base-mutation_environment.pdf" will be saved to the working directory.

Author(s)

Julia Bischof

References

IMGT/HighV-QUEST V-REGION mutation and AA change table: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#mut-table

IMGT Index "Mutation": http://www.imgt.org/IMGTindex/mutation.php

See Also

sequences.mutation.base, plotSequencesMutationBase,

sequences.mutation.AA, plotSequencesMutationAA,

sequences.mutation,

Examples

```
data(mutationtab)
data(summarytab)
V.base.mut<-sequences.mutation.base(mutationtab = mutationtab, summarytab = summarytab,
     sequence = "V", nrCores = 1)
```

```
## Not run:
plotSequencesMutationBase(mutationBaseTab = V.base.mut, plotMutation = T)

## End(Not run)
```

---

summarytab *Summary information of B cells of one individual*

Description

Data frame represents an output from IMGT/HighV-QUEST, file 1_Summary(...).txt

Usage

```
data("summarytab")
```

Format

A data frame with 3000 observations on the following 29 variables.

`Sequence_number` a numeric vector

`Sequence_ID` a character vector

`Functionality` a character vector

`V_GENE_and_allele` a character vector

`V_REGION_score` a numeric vector

`V_REGION_identity_%` a numeric vector

`V_REGION_identity_nt` a character vector

`V_REGION_identity_%_(with_ins/del_events)` a numeric vector

`V_REGION_identity_nt_(with_ins/del_events)` a character vector

`J_GENE_and_allele` a character vector

`J_REGION_score` a numeric vector

`J_REGION_identity_%` a numeric vector

`J_REGION_identity_nt` a character vector

`D_GENE_and_allele` a character vector

`D_REGION_reading_frame` a numeric vector

`CDR1_IMGT_length` a numeric vector

`CDR2_IMGT_length` a character vector

`CDR3_IMGT_length` a character vector

`CDR_IMGT_lengths` a character vector

`FR_IMGT_lengths` a character vector

`AA_JUNCTION` a character vector

JUNCTION_frame a character vector

Orientation a character vector

Functionality_comment a character vector

V_REGION_potential_ins/del a character vector

J_GENE_and_allele_comment a character vector

V_REGION_insertions a character vector

V_REGION_deletions a character vector

Sequence a character vector

References

Alamyar, E. et al., IMGT/HighV-QUEST: A High-Throughput System and Web Portal for the Analysis of Rearranged Nucleotide Sequences of Antigen Receptors, JOBIM 2010, Paper 63 (2010).

Brochet, X. et al., Nucl. Acids Res. 36, W503-508 (2008). PMID: 18503082

IMGT/HighV-QUEST 1_Summary file description: http://www.imgt.org/IMGT_vquest/share/textes/imgtvquest.html#Esummary

Examples

```
data(summarytab)
str(summarytab)
```

---

trueDiversity *True diversity of sequences*

Description

This function provides information about the true diversity. Richness or diversity is calculated for sequences of the same length, for each position. Analysis of true diversity of order 0 (richness), 1 (Shannon) and 2 (Simpson) is possible (see Details).

trueDiversity returns a list containing diversity indices.

plotTrueDiversity gives an overview about the richness or diversity of sequences with the same length.

Usage

```
trueDiversity(sequences = NULL, aaDistribution.tab = NULL, order = c(0,1,2))

plotTrueDiversity(trueDiversity.tab=NULL, mean.plot=T, color="black", PDF=NULL)
```

Arguments

sequences	Vector containing sequences (see Details)
aaDistribution.tab	Output of the function AADistribution() (see Details)
order	True diversity order (q). Values: 0, 1, 2 (default: 1)
trueDiversity.tab	Output of function trueDiversity()
mean.plot	Includes only one figure with mean diversities (default = T)
color	color used for plot (default: black)
PDF	PDF project name (see Details)

Details

This functions needs either a vector of sequences or the output of AADistribution() as input. In first case AADistribution() is applied to data set and than diversity is measured. Richness or diversity is calculated for sequences of the same length, for each position. Analysis of true diversity of order 0, 1 and 2 is possible. Order 0: Richness (in this case it represents number of different amino acids per position). Order 1: Exponential function of Shannon entropy using the natural logarithm (weights all amino acids by their frequency). Order 2: Inverse Simpson entropy (weights all amino acids by their frequency, but weights are given more to abundant amino acids). These indices are very similar (Hill, 1973). For example the exponential function of Shannon index is linearly related to inverse Simpson.

plotTrueDiversity returns an image with diversity plots for each length, if mean.plot = F. In the case of mean.plot = T, only one figure is returned, where mean diversity values for each sequence length are plotted. Each plot contains the richness or diversity (y-axis) for each position (x-axis).

The PDF character string should be only the project name (without ".pdf"). A figure called "PDF"_Trueiversity_q"order".pdf will be saved to the working directory.

Value

Output is a list containing

True_diversity_order	order of true diversity (q=0,1,2)
True_diversity	a list of true diversities for each position of each length

Note

For large datasets computational time can be extensive for the calculation of amino acid proportions.

Author(s)

Julia Bischof

References

M. O. Hill: Diversity and Evenness: A Unifying Notation and Its Consequences; Ecology 54:2, p 427-432 (1973)

Lou Jost: Entropy and diversity; OIKOS 113:2 (2006)

Jari Oksanen, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens and Helene Wagner (2015). vegan: Community Ecology Package. R package version 2.3-0. http://CRAN.R-project.org/package=vegan

See Also

aaDistribution, trueDiversity, diversity

Examples

```
data(aaseqtab)
trueDiv<-trueDiversity(sequences = aaseqtab$CDR3_IMGT, order = 1)
## Not run: plotTrueDiversity(trueDiversity.tab=trueDiv,color="red", PDF="Example")
```

---

vgenes *VH gene usage data*

Description

VH gene usage data of 10 samples (rows) and 30 genes (columns).

Usage

```
data("vgenes")
```

Format

The format is: num [1:10, 1:30] 0.0628 0.0529 0.0248 0.0512 0.0184 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:10] "Sample1" "Sample2" "Sample3" "Sample4" ... ..$: chr [1:30] "IGHV4-34" "IGHV1-8" "IGHV1-3" "IGHV3-23" ...

Examples

```
data(vgenes)
str(vgenes)
barplot(t(vgenes), col = rainbow(n = ncol(vgenes),start=0.2, end = 0.9),
     xlim = c(0, nrow(vgenes)+5), ylab="proportion",
     main="VH gene usage", las=3)
legend("right",col = rainbow(n = ncol(vgenes)), colnames(vgenes), pch=15)
```

Index