Package 'SSHAARP'

December 11, 2024

Version 2.0.5

Date 2024-12-10

Title Searching Shared HLA Amino Acid Residue Prevalence

Maintainer Livia Tran <livia.tran@ucsf.edu>

Depends R (>= 3.6)

Description Processes amino acid alignments produced by the 'IPD-IMGT/HLA (Immuno Polymorphism-ImMunoGeneTics/Human Leukocyte Antigen) Database' to identify user-defined amino acid residue motifs shared across HLA alleles, HLA alleles, or HLA haplotypes, and calculates frequencies based on HLA allele frequency data. 'SSHAARP' (Searching Shared HLA Amino Acid Residue Prevalence) uses 'Generic Mapping Tools (GMT)' software and the 'GMT' R package to generate global frequency heat maps that illustrate the distribution of each user-defined map around the globe. 'SSHAARP' analyzes the allele frequency data described by Solberg et al. (2008) <doi:10.1016/j.humimm.2008.05.001>, a global set of 497 population samples from 185 published datasets, representing 66,800 individuals total. Users may also specify their own datasets, but file conventions must follow the prebundled Solberg dataset, or the mock haplotype dataset.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports data.table, stringr, gtools, BIGDAWG, gmt, DescTools, dplyr, utils, filesstrings, purrr, stringi, HLAtools

Suggests knitr, rmarkdown

VignetteBuilder knitr

SystemRequirements GMT (5 or 6), Ghostscript (>=9.6)

RoxygenNote 7.3.2

NeedsCompilation no

Author Livia Tran [aut, cre],
Steven Mack [aut],
Josh Bredeweg [ctb],
Dale Steinhardt [ctb]

Repository CRAN

Date/Publication 2024-12-11 07:50:06 UTC

Contents

`checkAlleleSyntax` *Check allele syntax*

Description

Checks if allele syntax is valid.

Usage

```
checkAlleleSyntax(allele, filename)
```

Arguments

`allele`	An allele name written in the IPD-IMGT/HLA Database format.
`filename`	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.

Value

TRUE if allele syntax is correct. Otherwise,a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

`checkHaplotypeSyntax` *Check haplotype syntax*

Description

Checks if alleles in a haplotype have correct syntax and an appropriate number of fields.

Usage

```
checkHaplotypeSyntax(haplotype, filename)
```

Arguments

`haplotype`	A haplotype where allele names are written in the IPD-IMGT/HLA Database format, and have 1-4 fields. Alleles in haplotypes may be delimited by "-" or "~".
`filename`	The full file path of the user specified dataset if the user wishes to use their own file, or pre-bundled mock haplotype dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the mock haplotype dataset bundled with the package.

Value

TRUE if all alleles in entered haplotype have correct syntax and appropriate number of fields. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

`checkLocusANHIG` *Check locus validity*

Description

Checks if the locus in the entered variant is a protein-coding gene annotated by the IPD-IMGT/HLA Database

Usage

```
checkLocusANHIG(variant)
```

Arguments

variant	An amino acid motif or allele with an HLA locus name followed by an asterisk. This function ONLY evaluates if the locus in the entered variant is a protein-coding gene.

Value

TRUE if locus in entered variant are in the IPD-IMGT/HLA Database. Otherwise, a vector with FALSE and an error message is returned.

Note

For internal SSHAARP use only.

Examples

```
#Example of valid locus in a motif
checkLocusANHIG("DRB1*26F~28E")
#[1] TRUE

#Example of an invalid locus in an allele
checkLocusANHIG("BOO*01:01")
#[1] "FALSE"                    "BOO is not a valid locus."
```

---

checkLocusDataset — *Check locus validity and if the locus is present in the user specified dataset*

Description

Checks if the locus in the entered variant is a protein-coding gene annotated by the IPD-IMGT/HLA Database, and if it is in the user specified dataset.

Usage

```
checkLocusDataset(variant, filename)
```

Arguments

variant	An allele or an amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids. Haplotypes must contain alleles that follow the aforementioned format, and may be delimited by "~" or "-".
filename	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset or mock haplotype dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the datasets bundled with the package. Allele and motif datasets should follow the Solberg dataset format, and haplotype datasets should follow the SSHAARP haplotype mock data format.

Value

TRUE if locus is a protein-coding gene and is in the specified dataset. Otherwise, a vector with FALSE and an error message is returned.

Note

For internal SSHAARP use only.

`checkMotifSyntax` *Check motif syntax*

Description

Checks if motif syntax is valid.

Usage

```
checkMotifSyntax(motif, filename)
```

Arguments

`motif`	An amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids.
`filename`	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.

Value

TRUE if the motif syntax is valid. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

Examples

```
#Example with correct motif syntax where user specified dataset is the Solberg dataset
checkMotifSyntax("DRB1*26F~28E~30Y", filename=SSHAARP::solberg_dataset)

#Example with incorrect motif syntax where user specified dataset is the Solberg dataset
checkMotifSyntax("DRB1****26F~28E", filename=SSHAARP::solberg_dataset)
```

checkPosition *Checks if amino acid positions in motif exist*

Description

Checks if amino acid positions in the entered motif exist in IMGTprotalignments.

Usage

```
checkPosition(motif, filename, alignments)
```

Arguments

motif	An amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids. This function ONLY checks if the entered amino acid positions exist in IMGTprotalignments.
filename	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.
alignments	A list object of sub-lists of data frames of protein alignments for the HLA and HLA-region genes supported in the ANHIG/IMGTHLA GitHub Repository. Alignments will always be the most recent version IPD-IMGT/HLA Database version.

Value

TRUE if all of the amino acid positions in a motif exist. Otherwise, a vector with FALSE and an error message is returned.

Note

For internal SSHAARP use only.

dataSubset *Dataset manipulation for motifs and alleles*

Description

Returns a modified version of the user selected dataset that includes a column of locus*allele names, is sorted by by population name, and is reduced to the specified locus. Cardinal coordinates are converted to their Cartesian equivalents (i.e. 50S is converted to -50).

Usage

```
dataSubset(variant, filename)
```

Arguments

variant	An allele or an amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids.
filename	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.

Value

A data frame containing a reformatted version of the user selected dataset, with rows ordered by population name, Cartesian coordinates in the latit and longit columns, and limited to populations with data for the specified locus. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

The Solberg dataset is the tab-delimited '1-locus-alleles.dat' text file in the results.zip archive at http://pypop.org/popdata/.

The Solberg dataset is also prepackaged into SSHAARP as 'solberg_dataset'.

---

dataSubsetHaplo *Dataset manipulation for haplotypes*

Description

Returns the user input dataset that contains the selected haplotype.

Usage

```
dataSubsetHaplo(haplotype, filename, AFND, alignments)
```

Arguments

haplotype	A haplotype where allele names are written in the IPD-IMGT/HLA Database format, and have 1-4 fields. Alleles in haplotypes may be delimited by "-" or "~".
filename	The full file path of the user specified dataset if the user wishes to use their own file, or pre-bundled mock haplotype dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the mock haplotype dataset bundled with the package.

AFND	A logical parameter that determines whether the user specified dataset is data from AFND. This parameter is only relevant if haplotype maps are being made.
alignments	A list object of sub-lists of data frames of protein alignments for the HLA and HLA-region genes supported in the ANHIG/IMGTHLA GitHub Repository. Alignments will always be the most recent version IPD-IMGT/HLA Database version.

Value

A two element list with 1) a subset data frame containing only haplotypes with alleles present in the user input haplotype, and 2) a data frame of the full dataset. Alleles with two fields will be evaluated with their three and four field allele equivalents, and alleles with three fields will be evaluated with their four field allele equivalent. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

---

findMotif *Returns an alignment data frame of alleles that share a specific amino acid motif*

Description

Returns an alignment data frame of alleles that share a specific amino acid motif.

Usage

```
findMotif(motif, filename, alignments)
```

Arguments

motif	An amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids.
filename	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.
alignments	A list object of sub-lists of data frames of protein alignments for the HLA and HLA-region genes supported in the ANHIG/IMGTHLA GitHub Repository. Alignments will always be the most recent version IPD-IMGT/HLA Database version.

Value

An amino acid alignment dataframe of alleles that share the specified motif. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

`getVariantInfo` *Locus and allele or motif extraction for motifs, allele, or haplotype mapping*

Description

Extracts locus and allele or motif information from variant.

Usage

```
getVariantInfo(variant)
```

Arguments

`variant`	An amino acid motif or allele. Amino acid motifs must be in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Alleles must have 1-4 fields.

Value

A list object with loci and allele or motif information.

Note

For internal SSHAARP use only.

`isNamePresent` *Checks if name portion of entered variant is present*

Description

Checks if the name portion of the entered variant is present. Names consist of information following the locus and asterisk of the entered variant.

Usage

```
isNamePresent(variant, variantType)
```

Arguments

variant	An amino acid motif or allele. Amino acid motifs must be in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Alleles must have 1-4 fields.
variantType	Identifies whether the variant is an allele or motif.

Value

TRUE if name is present. Otherwise, a vector with FALSE and an error message is returned.

Note

For internal SSHAARP use only.

---

mock_haplotype_dataset *Mock Haplotype Dataset*

Description

A dataframe containing mock haplotype data modeled after the Allele Frequency Network Database (AFND) haplotype data structure.

Usage

```
mock_haplotype_dataset
```

Format

An object of class `data.frame` with 24989 rows and 7 columns.

PALM *Population Allele Locating Mapmaker*

Description

Produces a frequency heatmap for a specified allele, amino-acid motif, or haplotype based on the allele frequency data in the Solberg dataset.

Usage

```
PALM(
  variant,
  variantType,
  filename,
  mask = FALSE,
  color = TRUE,
  filterMigrant = TRUE,
  mapScale = TRUE,
  direct = getwd(),
  AFND = TRUE,
  generateLowFreq = TRUE,
  resolution = 500
)
```

Arguments

variant	An allele or an amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids. Haplotypes must contain alleles that follow the aforementioned format, and may be delimited by "~" or "-".
variantType	Specifies whether the variant is an allele, motif, or haplotype.
filename	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset or mock haplotype dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the datasets bundled with the package. Allele and motif datasets should follow the Solberg dataset format, and haplotype datasets should follow the SSHAARP haplotype mock data format.
mask	A logical parameter that determines if areas with little to no population coverage should be masked. The default value is set to FALSE.
color	A logical parameter that identifies if the heat maps should be made in color (TRUE) or gray scale (FALSE). The default value is TRUE.
filterMigrant	A logical parameter that determines if admixed populations (OTH) and migrant populations (i.e. any complexities with the 'mig') should be excluded from the dataset. The default value is TRUE.
mapScale	A logical parameter that determines if the max frequency value of the map scale should be 1 (FALSE), or if it should represent the maximum frequency of the chosen motif, allele, or haplotype (TRUE). The default value is TRUE.
direct	The directory into which the map produced is written. The default directory is the user's working directory.
AFND	A logical parameter that determines whether the user specified dataset is data from AFND. This parameter is only relevant if haplotype maps are being made. Default is TRUE.
generateLowFreq	A logical parameter that determines whether maps should be generated for a variant if the maximum frequency for the variant is low frequency. Low fre-

quency populations are defined as those with a frequency of 0.000, indicating three zeros after the decimal point.

`resolution` An integer for raster resolution in dpi for the final map output. It is not recommended to go below 400. Default is set to 500.

Value

The specified motif and the directory into which the heat map was written are returned in an invisible character vector. Otherwise, a warning message is returned.

Note

IMGT protein alignments will be generated for the locus of the specified variant the first time PALM is executed for a given locus. The alignments will be saved to the temp directory and referenced by PALM. PALM checks if the locus specific alignment is present in the temp directory; if it is not, a protein alignment object will be built for the locus. Restarting the R session will remove existing alignments.

The produced frequency heatmap is generated by using the Generic Mapping Tools (GMT) R Package, which is an interface between R and the GMT map making software.

The Solberg dataset is the tab-delimited '1-locus-alleles.dat' text file in the results.zip archive at http://pypop.org/popdata/.

The Solberg dataset is also prepackaged into SSHAARP as 'solberg_dataset'.

A mock haplotype dataset modeled after the AFND network's haplotype dataset structure is available for usage under "mock_haplotype_dataset".

While the map legend identifies the highest frequency value, values in this range may not be represented on the map due to frequency averaging over neighboring populations.

References

Solberg et.al. (2008) <doi: 10.1016/j.humimm.2008.05.001>

Examples

```
#Example to produce a motif color map where migrant populations are filtered out, mask is off
## Not run: PALM("DRB1*26F~28E~30Y",
variantType="motif",
mask = FALSE,
filterMigrant=TRUE,
filename = SSHAARP::solberg_dataset)
## End(Not run)

#Example to produce an allele greyscale map where migrant populations are not filtered, mask is on
## Not run: PALM("DRB1*01:01",
variantType="allele",
mask = TRUE, color=FALSE,
filterMigrant=FALSE,
filename = SSHAARP::solberg_dataset)
## End(Not run)
```

```
#Example to produce a color allele map with mapScale T and the allele has more than 2 fields
## Not run: PALM("DRB1*01:01:01",
variantType="allele",
filterMigrant=FALSE,
mapScale=TRUE,
filename = SSHAARP::solberg_dataset)
## End(Not run)

#Example to produce a color haplotype map with default parameters with the mock haplotype dataset
## Not run: PALM("DRB1*01:01~A*01:01",
variantType = "haplotype",
filename = SSHAARP::mock_haplotype_dataset)
## End(Not run)
```

`readFilename` *Designates dataset by either reading in file the user has provided, or using the Solberg dataset or mock haplotype dataset*

Description

Returns the user specified dataset, either by reading in the file the user has provided, or by using the Solberg dataset or mock haplotype dataset. If the user provides a dataset and the filename is not found, an error will be returned. If the user provided dataset does not have the same number of columns or the same column names as the reference datasets, an error message will be returned.

Usage

```
readFilename(filename, variant)
```

Arguments

`filename`	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset or mock haplotype dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the datasets bundled with the package. Allele and motif datasets should follow the Solberg dataset format, and haplotype datasets should follow the SSHAARP haplotype mock data format.
`variant`	An allele or an amino acid motif in the following format: Locus*##$~##$~##$, where ## identifies a peptide position, and $ identifies an amino acid residue. Motifs can include any number of amino acids. Haplotypes must contain alleles that follow the aforementioned format, and may be delimited by "~" or "-".

Value

A dataframe of the user specified dataset.

Note

For internal SSHAARP use only.

`solberg_dataset` *Solberg Dataset*

Description

A dataframe of the original Solberg dataset, which is a global dataset of 497 population samples from 185 published datasets, representing 66,800 individuals. For more information on the Solberg dataset, please see the vignette.

Usage

```
solberg_dataset
```

Format

A dataframe with 20163 rows and 13 columns.

Source

results.zip file from http://pypop.org/popdata/

References

Solberg et.al "Balancing selection and heterogeneity across the classical human leukocyte antigen loci: A meta-analytic review of 497 population studies". Human Immunology (2008) 69, 443–464

`verifyAlleleANHIG` *Verifies the allele entered is present in the IMGT protein alignments*

Description

Verifies the allele entered is present in IMGT protein alignments

Usage

```
verifyAlleleANHIG(allele, filename, alignments)
```

Arguments

allele	An allele name written in the IPD-IMGT/HLA Database format.
filename	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.
alignments	A list object of sub-lists of data frames of protein alignments for the HLA and HLA-region genes supported in the ANHIG/IMGTHLA GitHub Repository. Alignments will always be the most recent version IPD-IMGT/HLA Database version.

Value

TRUE if allele is present in the IMGTprotalignment object. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

---

verifyAlleleANHIGHaplo

Verifies the alleles in entered haplotype are present in IMGTprotalignments

---

Description

Verifies the alleles in entered haplotype are present in IMGTprotalignments.

Usage

```
verifyAlleleANHIGHaplo(haplotype, filename, alignments)
```

Arguments

haplotype	A haplotype where allele names are written in the IPD-IMGT/HLA Database format, and have 1-4 fields. Alleles in haplotypes may be delimited by "-" or "~".
filename	The full file path of the user specified dataset if the user wishes to use their own file, or pre-bundled mock haplotype dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the mock haplotype dataset bundled with the package.
alignments	A list object of sub-lists of data frames of protein alignments for the HLA and HLA-region genes supported in the ANHIG/IMGTHLA GitHub Repository. Alignments will always be the most recent version IPD-IMGT/HLA Database version.

Value

TRUE if all alleles in a haplotype are present in the IMGTprotalignment object. Otherwise, a vector containing FALSE and an error message is returned.

Note

For internal SSHAARP use only.

`verifyAlleleDataset` *Verifies the allele entered is present in specified dataset*

Description

Verifies the allele entered is present in the specified dataset.

Usage

```
verifyAlleleDataset(allele, filename, alignments)
```

Arguments

`allele`	An allele name written in the IPD-IMGT/HLA Database format.
`filename`	The full file path of the user specified dataset if the user wishes to use their own file, or the pre-bundled Solberg dataset. User provided datasets must be a .dat, .txt, or.csv file, and must conform to the structure and format of the Solberg dataset.
`alignments`	A list object of sub-lists of data frames of protein alignments for the HLA and HLA-region genes supported in the ANHIG/IMGTHLA GitHub Repository. Alignments will always be the most recent version IPD-IMGT/HLA Database version.

Value

TRUE if the allele is present in the specified data set, and the filtered allele dataset. If a user enters an allele with more than two fields and has selected the Solberg dataset as the data source, a message informing the user that the allele has been truncated is appended to the output. If an allele entered is valid, but is not present in the user provided dataset, a warning message is returned.

Note

For internal SSHAARP use only.

Index