

Package ‘immunogenetr’

January 18, 2025

Type Package

Title Data Wrangling for HLA Data

Version 0.1.0

Description Makes wrangling HLA data to WHO-approved nomenclature easier.
Uses GL string for encoding genotypes.

License GPL-3 # Should probably think about this.

URL https://github.com/k96nb01/immunogenetr_package

BugReports https://github.com/k96nb01/immunogenetr_package/issues

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports cli, dplyr, glue, purrr, rlang, stringr, tibble, tidyr,
tidyselect, xml2

NeedsCompilation no

Author Nicholas Brown [cre, aut],
Busra Coskun [aut]

Maintainer Nicholas Brown <nicholas.brown@pennmedicine.upenn.edu>

Depends R (>= 3.5.0)

Contents

ambiguity_table_to_GLstring	2
GLstring_gene_copies_combine	3
HLA_column_repair	3
HLA_mismatch_logical	4
HLA_mismatch_number	5
HLA_prefix_add	6
HLA_validate	7

Index	8
--------------	----------

```
ambiguity_table_to_GLstring
      ambiguity_table_to_GLstring
```

Description

A function that converts a data table of HLA allele ambiguities into a GL string format. The function processes the table by combining allele ambiguities, haplotypes, gene copies, and loci into a structured GL string.

Usage

```
ambiguity_table_to_GLstring(data)
```

Arguments

data	A data frame containing columns that represent possible gene locations, loci, genotype ambiguities, genotypes, and haplotypes.
------	--------------------------------------------------------------------------------------------------------------------------------

Value

A GL string representing the combined gene locations, loci, genotype ambiguities, genotypes, and haplotypes.

Examples

```
# Example data frame input
data <- tibble(
  value = c(
    "HLA-A*01:01:01:01", "HLA-A*01:02", "HLA-A*01:03", "HLA-A*01:95",
    "HLA-A*24:02:01:01", "HLA-A*01:01:01:01", "HLA-A*01:03",
    "HLA-A*24:03:01:01", "HLA-B*07:01:01", "B*15:01:01"),
  possible_gene_location = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
  locus = c("HLA-A", "HLA-A", "HLA-A", "HLA-A", "HLA-A", "HLA-A", "HLA-A",
    "HLA-A", "HLA-B", "HLA-B"),
  genotype_ambiguity = c(1, 1, 1, 1, 1, 2, 2, 2, 1, 1),
  genotype = c(1, 1, 1, 1, 2, 1, 1, 2, 1, 2),
  haplotype = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
  allele = c(1, 2, 3, 4, 1, 1, 2, 1, 1, 1)
)
result <- ambiguity_table_to_GLstring(data)
print(result)
```

```
GLstring_gene_copies_combine
      GLstring_gene_copies_combine
```

Description

A function for combining two columns of typing from the same locus into a single column in the appropriate GL string format.

Usage

```
GLstring_gene_copies_combine(.data, columns, sample_column = "sample")
```

Arguments

<code>.data</code>	A data frame
<code>columns</code>	The names of the columns in the data frame that contain typing information to be combined
<code>sample_column</code>	The name of the column that identifies samples in the data frame. Default is "sample".

Value

A data frame with the specified columns combined into a single column for each locus, in GL string format.

Examples

```
HLA_type <- tibble(
  sample = c("sample1", "sample2"),
  HLA_A1 = c("HLA-A*01:01", "HLA-A*02:01"),
  HLA_A2 = c("HLA-A*01:02", "HLA-A*02:02"),
  stringsAsFactors = FALSE
)

HLA_type %>% GLstring_gene_copies_combine(columns = c("HLA_A1", "HLA_A2"))
```

```
HLA_column_repair      HLA_column_repair
```

Description

This function will change column names that have the official HLA nomenclature (e.g. "HLA-A*" or "HLA-A") to a format more easily selected in tidyverse functions (e.g. "HLA_A"). The dash and asterisk are special characters in R, and makes selecting columns by name difficult. This function will also allow for easily changing back to WHO-compliant nomenclature (e.g. "HLA-A*").

Usage

```
HLA_column_repair(data, format = "tidyverse", asterisk = FALSE)
```

Arguments

data	A data frame
format	Either "tidyverse" or "WHO".
asterisk	Logical value to return column with an asterisk.

Value

A data frame object with column names renamed in the specified format.

Examples

```
HLA_type <- tibble(
  "HLA-A*" = c("01:01", "02:01"),
  "HLA-B*" = c("07:02", "08:01"),
  "HLA-C*" = c("03:04", "04:01")
)
HLA_type %>% HLA_column_repair(format = "tidyverse")
```

HLA_mismatch_logical *HLA_mismatch_logical*

Description

Determines if there are any mismatches between recipient and donor HLA alleles for the specified loci. Returns 'TRUE' if mismatches are present, and 'FALSE' otherwise.

Usage

```
HLA_mismatch_logical(
  GL_string_recip,
  GL_string_donor,
  loci,
  direction = c("HvG", "GvH", "bidirectional"),
  homozygous_count = 2
)
```

Arguments

GL_string_recip	A GL strings representing the recipient's HLA genotypes.
GL_string_donor	A GL strings representing the donor's HLA genotypes.
loci	A character vector specifying the loci to be considered for mismatch calculation.
direction	A character string indicating the direction of mismatch. Options are "HvG" (host vs. graft), "GvH" (graft vs. host), or "bidirectional" (max of "HvG" and "GvH").
homozygous_count	An integer specifying how to handle homozygosity. Defaults to 2, where homozygous alleles are treated as duplicated for mismatch calculations. Can be specified to be 1, in which case homozygous alleles are treated as single occurrences without duplication.

Value

A logical value ('TRUE' or 'FALSE'): - 'TRUE' if there are mismatches between recipient and donor HLA alleles. - 'FALSE' if there are no mismatches.

Examples

```
# Example recipient and donor GL strings
GL_string_recip <- "HLA-A*03:01+HLA-A*74:01^HLA-DRB3*03:01^HLA-DRB5*02:21"
GL_string_donor <- "HLA-A*03:02+HLA-A*20:01^HLA-DRB3*03:01"

# Check if there are mismatches for HLA-A (Graft vs. Host)
has_mismatch <- HLA_mismatch_logical(GL_string_recip, GL_string_donor, loci =
  "HLA-A", direction = "GvH")
print(has_mismatch)
# Output: TRUE
```

HLA_mismatch_number	<i>HLA_mismatch_number</i>
---------------------	----------------------------

Description

Calculates the number of mismatched HLA alleles between a recipient and a donor across specified loci. Supports mismatch calculations for host-vs-graft (HvG), graft-vs-host (GvH), or bidirectional mismatches.

Usage

```
HLA_mismatch_number(
  GL_string_recip,
  GL_string_donor,
  loci,
  direction = c("HvG", "GvH", "bidirectional"),
  homozygous_count = 2
)
```

Arguments

GL_string_recip	A GL strings representing the recipient's HLA genotypes.
GL_string_donor	A GL strings representing the donor's HLA genotypes.
loci	A character vector specifying the loci to be considered for mismatch calculation.
direction	A character string indicating the direction of mismatch. Options are "HvG" (host vs. graft), "GvH" (graft vs. host), or "bidirectional" (max of "HvG" and "GvH").
homozygous_count	An integer specifying how to handle homozygosity. Defaults to 2, where homozygous alleles are treated as duplicated for mismatch calculations. Can be specified to be 1, in which case homozygous alleles are treated as single occurrences without duplication.

Value

An integer value or a character string: - If 'loci' includes only one locus, the function returns an integer mismatch count for that locus. - If 'loci' includes multiple loci, the function returns a character string in the format "Locus1=Count1, Locus2=Count2, ...".

Examples

```
# Example recipient and donor GL strings
GL_string_recip <- "HLA-A*01:01+HLA-A*02:01^HLA-B*07:02+HLA-B*08:01"
GL_string_donor <- "HLA-A*01:01+HLA-A*03:01^HLA-B*07:02+HLA-B*44:02"
loci <- c("HLA-A", "HLA-B")

# Calculate mismatch numbers (Host vs. Graft)
mismatch_count_HvG <- HLA_mismatch_number(GL_string_recip, GL_string_donor, loci, direction = "HvG")
print(mismatch_count_HvG)

# Calculate mismatch numbers (Graft vs. Host)
mismatch_count_GvH <- HLA_mismatch_number(GL_string_recip, GL_string_donor, loci, direction = "GvH")
print(mismatch_count_GvH)

# Calculate mismatch numbers (Bidirectional)
mismatch_count_bidirectional <- HLA_mismatch_number(GL_string_recip, GL_string_donor,
loci, direction = "bidirectional")
print(mismatch_count_bidirectional)
```

HLA_prefix_add	<i>HLA_prefix_add</i>
----------------	-----------------------

Description

This function adds a specified prefix to the beginning of each value in the identified columns of the given data frame.

Usage

```
HLA_prefix_add(.data, columns, prefix = "HLA-")
```

Arguments

.data	A data frame
columns	Name of columns in .data to which the prefix should be added
prefix	A character string to be added as a prefix to the column values. Default is "HLA-".

Value

A data frame with the specified prefix added to the values in the selected columns.

Examples

```
df <- data.frame(
  A1 = c("01:01", "02:01"),
  A2 = c("03:01", "11:01"),
  B1 = c("07:02", "08:01"),
  B2 = c("15:01", "44:02"),
  stringsAsFactors = FALSE
)

# Add HLA- prefix to columns A1 and A2
df %>% HLA_prefix_add(columns = c("A1", "A2"))
```

HLA_validate

*HLA_validate***Description**

Returns only HLA alleles in valid nomenclature, either serologic or molecular. Simple numbers, such as "2" or "27" will be returned as-is. Suffixes that are not WHO-recognized suffixes (L, S, C, A, Q, N) or G or P group designations will be removed. For example "novel" at the end of the allele will be removed, while "n" at the end of the allele will be retained. Other values, such as "blank" or "-" will be converted to NA values. This function is helpful for cleaning up the typing of an entire table of HLA values.

Usage

```
HLA_validate(data)
```

Arguments

data A string containing an HLA allele.

Value

A string with a valid HLA allele.

Examples

```
HLA_validate("HLA-A2")
HLA_validate("A*02:01:01:01N")
HLA_validate("A*02:01:01N")
HLA_validate("HLA-DRB1*02:03novel")
HLA_validate("HLA-DQB1*03:01v")
HLA_validate("HLA-DRB1*02:03P")
HLA_validate("HLA-DPB1*04:01:01G")
HLA_validate("2")
HLA_validate(2)
HLA_validate("B27")
HLA_validate("A*010101")
HLA_validate("-")
HLA_validate("blank")
```

Index

`ambiguity_table_to_GLstring`, [2](#)
`GLstring_gene_copies_combine`, [3](#)
`HLA_column_repair`, [3](#)
`HLA_mismatch_logical`, [4](#)
`HLA_mismatch_number`, [5](#)
`HLA_prefix_add`, [6](#)
`HLA_validate`, [7](#)