

Package: variantstring (via r-universe)

January 16, 2025

Type Package

Title Functions for working with variant string format

Version 1.7.0

Description Contains a series of functions for working with genetic information encoded in variant string format. Includes methods for comparing and manipulating strings.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports dplyr, stringr, tidyr

Config/pak/sysreqs libicu-dev

Repository <https://plasmogenepi.r-universe.dev>

RemoteUrl <https://github.com/mrc-ide/variantstring>

RemoteRef HEAD

RemoteSha 3e0f83f6808ffa2d7d05e2e01d11d37e59daf622

Contents

allowed_amino_acids	2
check_position_string	2
check_variant_string	3
compare_position_string	3
compare_variant_string	4
count_het_loci	4
drop_read_counts	5
extract_single_locus_variants	5
get_consistent_variants	6
long_to_position	6

long_to_variant	6
order_position_string	7
order_variant_string	7
position_from_variant_string	8
position_to_long	8
subset_position	9
variant_to_long	9

Index	10
--------------	-----------

allowed_amino_acids *List allowed amino acids*

Description

Returns a data.frame of allowed amino acid single-letter codes. These come from IUPAC (International Union of Pure and Applied Chemistry), [see here](#) for details.

Usage

```
allowed_amino_acids()
```

check_position_string *Check for a valid position string*

Description

Checks that an input string (or a vector of strings) matches the required format for a position string. This is equivalent to a full variant string but with the amino acid information removed, so just giving the gene name(s) and position(s).

Usage

```
check_position_string(x)
```

Arguments

x a character string or vector of character strings.

check_variant_string *Check for a valid variant string*

Description

Checks that an input string (or a vector of strings) matches the required format for a variant string.

Usage

```
check_variant_string(x)
```

Arguments

x a character string or vector of character strings.

compare_position_string
Compares a position strings against variant strings to look for a match

Description

Compares a target position string against a vector of comparison strings. A match is found if every codon position in every gene of the target is also found within the comparison (irrespective of the observed amino acids).

Usage

```
compare_position_string(target_string, comparison_strings)
```

Arguments

target_string a single position string that we want to compare.

comparison_strings
 a vector of variant strings against which the target is compared.

`compare_variant_string`*Compares variant strings to look for a match*

Description

Compares a target variant string against a vector of comparison strings. A match is found if every amino acid at every codon position in every gene of the target is also found within the comparison. Note that ambiguous matches may occur if there are multiple heterozygous loci in the comparison. In this case, the target may or may not be within this sample. A match is recorded but a second output also flags this as an ambiguous match.

Usage

```
compare_variant_string(target_string, comparison_strings)
```

Arguments

`target_string` a single variant string that we want to compare. Cannot contain any heterozygous calls.

`comparison_strings`
a vector of variant strings against which the target is compared.

`count_het_loci`*Count the number of heterozygous loci in each variant string*

Description

Count the number of heterozygous loci in each variant string and return as a vector.

Usage

```
count_het_loci(x)
```

Arguments

`x` a variant string or vector of variant strings.

drop_read_counts	<i>Drop read counts from a variant string.</i>
------------------	------------------------------------------------

Description

Takes a vector of variant strings and strips and information on read counts.

Usage

```
drop_read_counts(x)
```

Arguments

x a variant string or vector of variant strings.

extract_single_locus_variants	<i>Extract all single-locus variants from a variant string</i>
-------------------------------	----------------------------------------------------------------

Description

Takes a vector of variant strings, potentially with information at multiple codon positions or genes, and returns variant strings corresponding to all unique single-locus variants within the input. For example, crt:72_73:C_N/V can be extracted to crt:72:C, crt:73:N, and crt:73:V.

Usage

```
extract_single_locus_variants(x)
```

Arguments

x a vector of variant strings.

get_consistent_variants

Get all genotypes that are consistent with a variant string

Description

For a variant string with at most one heterozygous locus we can unambiguously define the genotypes that are present in this mixture. This function returns all such component genotypes.

Usage

```
get_consistent_variants(x)
```

Arguments

x a vector of variant strings.

long_to_position

Take long form information and convert to position string

Description

Takes a list of data.frames in long form and converts each to position string format.

Usage

```
long_to_position(x)
```

Arguments

x a list of data.frames.

long_to_variant

Take long form information and convert to variant string

Description

Takes a list of data.frames in long form and converts each to variant string format.

Usage

```
long_to_variant(x)
```

Arguments

x a list of data.frames.

order_position_string *Reorders a position string*

Description

Reorders a position string in alphabetical order of genes. This can be useful when checking for duplicated strings as the same information may be presented in a different order.

Usage

```
order_position_string(x)
```

Arguments

x a position string or vector of position strings.

order_variant_string *Reorders a variant string*

Description

Reorders a variant string in alphabetical order of genes, and then alphabetical order of amino acids at each heterozygous locus. This can be useful when checking for duplicated strings as the same information may be presented in a different order.

Usage

```
order_variant_string(x)
```

Arguments

x a variant string or vector of variant strings.

position_from_variant_string

Extract a position string from a variant string

Description

Extract a position string from a variant string by stripping the amino acids.

Usage

```
position_from_variant_string(x)
```

Arguments

x a character string or vector of character strings.

position_to_long

Expand position strings into long form data.frames

Description

Takes a vector of position strings and expands into a list of data.frames containing the same information in long form.

Usage

```
position_to_long(x)
```

Arguments

x a vector of position strings.

subset_position	<i>Subset position of a variant string</i>
-----------------	--------------------------------------------

Description

Given a vector of variant strings and a single position string, subsets all variant strings to only the genes and codons in the position string. Retains read counts at these positions if present.

Usage

```
subset_position(position_string, variant_strings)
```

Arguments

position_string
a single position string.

variant_strings
a variant string or vector of variant strings.

variant_to_long	<i>Expand variant strings into long form data.frames</i>
-----------------	----------------------------------------------------------

Description

Takes a vector of variant strings and expands into a list of data.frames containing the same information in long form.

Usage

```
variant_to_long(x)
```

Arguments

x
a vector of variant strings.

Index

`allowed_amino_acids`, 2

`check_position_string`, 2
`check_variant_string`, 3
`compare_position_string`, 3
`compare_variant_string`, 4
`count_het_loci`, 4

`drop_read_counts`, 5

`extract_single_locus_variants`, 5

`get_consistent_variants`, 6

`long_to_position`, 6
`long_to_variant`, 6

`order_position_string`, 7
`order_variant_string`, 7

`position_from_variant_string`, 8
`position_to_long`, 8

`subset_position`, 9

`variant_to_long`, 9