Package: relater (via r-universe)

August 23, 2024

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports data.table, Rcpp, R.utils, reshape2

RoxygenNote 6.1.1

LinkingTo Rcpp

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://leospeidel.r-universe.dev

RemoteUrl https://github.com/leospeidel/relater

RemoteRef HEAD

RemoteSha a9568a086894678c0b4255c12cd4d538f525960b

Contents

`filter.allele_ages` *Filter allele_ages by quality of tree*

Description

This function filters out selection pvalues on bad trees using summary statistics on tree quality stored in qual.

Usage

```
filter.allele_ages(allele_ages, qual)
```

Arguments

`allele_ages`	data.table. Obtained from get.allele_ages()
`qual`	data.frame. qual file

Value

Returns a data table.

Examples

```
# get allele ages
mut      <- read.mut(system.file("extdata/example.mut.gz", package = "relater"), CHR = 1)
sele     <- read.sele(system.file("extdata/example.sele.gz", package = "relater"), CHR = 1)
freq     <- read.freq(system.file("extdata/example.freq.gz", package = "relater"), CHR = 1)
allele_ages <- get.allele_ages(mut, freq, sele)

# read quality file
qual     <- read.qual(system.file("extdata/example.qual.gz", package = "relater"), CHR = 1)
filter.allele_ages(allele_ages, qual)
```

`get.allele_ages` *Compile allele ages file*

Description

This function aggregates mut, freq, and sele files inferred by Relate into an allele_ages file.

Usage

```
get.allele_ages(mut, freq, sele)
```

Arguments

`mut`	data.table. mut file
`freq`	data.frame. freq file
`sele`	data.frame. sele file

Value

Returns a data table.

Examples

```
mut      <- read.mut(system.file("extdata/example.mut.gz", package = "relater"), CHR = 1)
sele     <- read.sele(system.file("extdata/example.sele.gz", package = "relater"), CHR = 1)
freq     <- read.freq(system.file("extdata/example.freq.gz", package = "relater"), CHR = 1)
get.allele_ages(mut, freq, sele)
```

`get_hap_matrix` *Get haplotype matrix (L x N)*

Description

This function extracts the haplotype matrix from .haps data frames

Usage

```
get_hap_matrix(haps)
```

Arguments

`haps`	data.table.

Value

Returns a matrix.

Examples

```
haps <- read.haps(system.file("extdata/example.haps.gz", package = "relater"))
get_hap_matrix(haps)
```

PolyTest *Test for evidence of polygenic selection*

Description

Tests for evidence of polygenic selection by matching SNPs by DAF and using a one-sided Wilcoxon rank sum test.

Usage

```
PolyTest(DAFs, pvalues, quantiles, allele_ages_quantiles)
```

Arguments

DAFs	1d array. Derived allele frequencies of trait associations
pvalues	1d array. Selection pvalues of trait associations
quantiles	data frame. Obtained from PolyTest_Init.
allele_ages_quantiles	List of data tables. Obtained from PolyTest_Init.

Value

Returns a pvalue.

Examples

```
# Example analysis:
# read files
mut      <- read.mut(system.file("extdata/example.mut.gz", package = "relater"), CHR = 1)
sele     <- read.sele(system.file("extdata/example.sele.gz", package = "relater"), CHR = 1)
freq     <- read.freq(system.file("extdata/example.freq.gz", package = "relater"), CHR = 1)
qual     <- read.qual(system.file("extdata/example.qual.gz", package = "relater"), CHR = 1)

# Obtain allele_ages data table
allele_ages <- get.allele_ages(mut, freq, sele)
allele_ages <- filter.allele_ages(allele_ages, qual)

######### Polygenic selection test #########

# Initialise
quant       <- PolyTest_Init(allele_ages)

# Make a fake polygenic trait
df          <- allele_ages[!is.na(allele_ages$pvalue),]
df          <- df[sample(1:nrow(df), 50, replace = FALSE),]

# Run polygenic test
PolyTest(df$DAF, df$pvalue, quant$quantiles, quant$allele_ages_quantiles)
```

PolyTest_Init *Initialise polygenic selection test*

Description

This function initialises the polygenic selection test.

Usage

```
PolyTest_Init(allele_ages)
```

Arguments

allele_ages	data.table. Obtained from get.allele_ages()

Value

Returns a list containing a data frame (quantiles) and a list of data tables (allele_ages_quantiles).

Examples

```
## Not run:
PolyTest_Init(allele_ages)

## End(Not run)
```

PolyTest_ResampleSNPs *Resample selection p-values*

Description

Given DAF, resample 20 selection p-values of SNPs selected uniformly at random with matching DAF

Usage

```
PolyTest_ResampleSNPs(DAF, quantiles, allele_ages_quantiles)
```

Arguments

`DAF` int. Derived allele frequency

`quantiles` data frame. Obtained from PolyTest_Init.

`allele_ages_quantiles` List of data tables. Obtained from PolyTest_Init.

Value

Returns a numeric 1x20 matrix.

Examples

```
## Not run:
PolyTest_ResampleSNPs(DAF, quantiles, allele_ages_quantiles)

## End(Not run)
```

`read.avg_rate` *Parse average rate file*

Description

This function parses a *_avg.rate file, which stores average mutation rates inferred by Relate

Usage

```
read.avg_rate(filename)
```

Arguments

`filename` string. Filename

Value

Returns a data frame.

Examples

```
read.avg_rate(system.file("extdata/example_avg.rate.gz", package = "relater"))
```

`read.coal` *Parse coal file*

Description

This function parses a .coal file, which stores coalescence rates inferred by Relate

Usage

```
read.coal(filename)
```

Arguments

`filename`	string. Filename

Value

Returns a data frame.

Examples

```
read.coal(system.file("extdata/example.coal.gz", package = "relater"))
```

`read.freq` *Parse freq file*

Description

This function parses a .freq file, which stores frequencies in Relate-inferred genealogies

Usage

```
read.freq(filename, CHR = NA)
```

Arguments

`filename`	string. Filename
`CHR`	int. Chromosome index.

Value

Returns a data frame.

Examples

```
read.freq(system.file("extdata/example.freq.gz", package = "relater"))
```

read.haps *Parse haps file*

Description

This function uses fread to parse a .haps file, which is part of the haps/sample output file format of SHAPEIT2.

Usage

```
read.haps(filename, ...)
```

Arguments

`filename`	string. Filename
`...`	Any other parameters for fread

Value

Returns a data table.

Examples

```
read.haps(system.file("extdata/example.haps.gz", package = "relater"))
```

read.lin *Parse lin file*

Description

This function parses a .lin file, which stores number of lineages remaining in Relate-inferred genealogies

Usage

```
read.lin(filename, CHR = NA)
```

Arguments

`filename`	string. Filename
`CHR`	int. Chromosome index.

Value

Returns a data frame.

Examples

```
read.lin(system.file("extdata/example.lin.gz", package = "relater"))
```

read.mut *Parse mut file*

Description

This function uses fread to parse a .mut file, which is part of the anc/mut output file format of SHAPEIT2.

Usage

```
read.mut(filename, CHR = NA, ...)
```

Arguments

`filename`	string. Filename
`CHR`	int. Chromosome index.
`...`	Any other parameters for fread

Value

Returns a data table.

Examples

```
read.mut(system.file("extdata/example.mut.gz", package = "relater"))
```

read.qual *Parse qual file*

Description

This function parses a .qual file, which stores summary statistics on the quality of Relate-inferred genealogies

Usage

```
read.qual(filename, CHR = NA)
```

Arguments

`filename`	string. Filename
`CHR`	int. Chromosome index.

Value

Returns a data frame.

Examples

```
read.qual(system.file("extdata/example.qual.gz", package = "relater"))
```

---

read.sample *Parse sample file*

Description

This function uses fread to parse a .sample file, which is part of the haps/sample output file format of SHAPEIT2.

Usage

```
read.sample(filename, ...)
```

Arguments

filename	string. Filename
...	Any other parameters for fread

Value

Returns a data table.

Examples

```
read.sample(system.file("extdata/example.sample.gz", package = "relater"))
```

---

read.sele *Parse sele file*

Description

This function parses a .sele file, which stores selection p-values inferred by Relate

Usage

```
read.sele(filename, CHR = NA)
```

Arguments

`filename`	string. Filename
`CHR`	int. Chromosome index.

Value

Returns a data frame.

Examples

```
read.sele(system.file("extdata/example.sele.gz", package = "relater"))
```

relater *relater*

Description

Functions for parsing, writing, manipulating input and output files of Relate

Author(s)

Leo Speidel

timesTwo *Multiply a number by two*

Description

Multiply a number by two

Usage

```
timesTwo(x)
```

Arguments

`x`	A single integer.

Examples

```
timesTwo(2)
```

Index