Package 'minSNPs'

September 8, 2021

Title Resolution-Optimised SNPs Searcher

Version 0.0.1

Description This is a R implementation of ``Minimum SNPs'' software as described in ``Price E.P., Inman-Bamber, J., Thiruvenkataswamy, V., Huygens, F and Giffard, P.M." (2007) <doi:10.1186/1471-2105-8-278> ``Computer-aided identification of polymorphism sets diagnostic for groups of bacterial and viral genetic variants."

Depends R (>= 3.4.0)

License MIT + file LICENSE

Imports BiocParallel

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, testthat, pkgdown, seqinr, Biostrings, rmarkdown, withr

VignetteBuilder knitr

URL https://github.com/ludwigHoon/minSNPs

NeedsCompilation no

Author Ludwig Kian Soon Hoon [aut, cre] (<https://orcid.org/0000-0002-2310-3403>), Peter Shaw [aut, ctb] (<https://orcid.org/0000-0002-3187-8938>), Phil Giffard [aut, ctb] (<https://orcid.org/0000-0002-3030-9127>)

Maintainer Ludwig Kian Soon Hoon <ldwgkshoon@gmail.com>

Repository CRAN

Date/Publication 2021-09-08 09:10:02 UTC

R topics documented:

`calculate_percent` `calculate_percent`

Description

`calculate_percent` is used to calculate dissimilarity index, proportion of isolates not in goi that have been discriminated against. 1 being all and 0 being none.

Usage

```
calculate_percent(pattern, goi)
```

Arguments

`pattern`	list of sequences
`goi`	group of interest

Value

Will return the dissimilarity index of the list of patterns.

`calculate_simpson` `calculate_simpson`

Description

`calculate_simpson` is used to calculate Simpson's index. Which is in the range of 0-1, where the greater the value, the more diverse the population.

Usage

```
calculate_simpson(pattern)
```

Arguments

`pattern`	list of sequences

Value

Will return the Simpson's index of the list of patterns.

check_percent check_percent

Description

check_percent is used to check if parameters needed by calculate_percent are all present.

Usage

```
check_percent(list_of_parameters)
```

Arguments

list_of_parameters	is a list of parameter passed to functions that will perform the calculation

find_optimised_snps find_optimised_snps

Description

find_optimised_snps is used to find optimised SNPs set.

Usage

```
find_optimised_snps(
  seqc,
  metric = "simpson",
  goi = c(),
  accept_multiallelic = TRUE,
  number_of_result = 1,
  max_depth = 1,
  included_positions = c(),
  excluded_positions = c(),
  iterate_included = FALSE,
  bp = SerialParam(),
  ...
)
```

Arguments

seqc	list of sequences, either passed directly from process_allele or read_fasta or equivalence
metric	either ‘simpson‘ or ‘percent‘
goi	group of interest, if creteria is percent, must be specified, ignored otherwise

`accept_multiallelic`
: whether include positions with > 1 state in goi

`number_of_result`
: number of results to return, 0 will be coerced to 1

`max_depth`
: maximum depth to go before terminating, 0 means it will only calculate the metric for included position

`included_positions`
: included positions

`excluded_positions`
: excluded positions

`iterate_included`
: whether to calculate index at each level of the included SNPs

`bp`
: BiocParallel backend. Rule of thumbs: use MulticoreParam()

`...`
: other parameters as needed

Value

Will return the resolution-optimised SNPs set, based on the metric.

`get_metric_fun` `get_metric_fun`

Description

`get_metric_fun` is used to get the metrics function and required parameters. Additional metric may set by assigning to 'MinSNPs_metrics' variable.

Usage

```
get_metric_fun(metric_name)
```

Arguments

`metric_name`
: name of the metric, by default percent/simpson

Value

a list, including the function to calculate the metric based on a position ('calc'), and function to check for additional parameters the function need ('args')

output_result output_result

Description

output_result is used to present the result and save the result as csv.

Usage

```
output_result(result, view = "", ...)
```

Arguments

result	is the result from find_optimised_snps
view	how to present the output, "csv" will be saved as a file. Otherwise, printed to console.
...	if view is "csv", file name can be passed, e.g., file_name = "result.csv", otherwise, file is saved as <timestamp>.csv.

Value

NULL, result either printed or saved as csv.

process_allele process_allele

Description

process_allele is used to returned the processed allelic profiles, by removing the allele profile with duplicate name and length different from most. 1st allele profile with the duplicated name is returned, the longer length is taken as normal should there be 2 modes.

Usage

```
process_allele(
  seqc,
  bp = BiocParallel::SerialParam(),
  dash_ignore = TRUE,
  accepted_char = c("A", "C", "T", "G"),
  ignore_case = TRUE
)
```

Arguments

seqc	a list containing list of nucleotides. To keep it simple, use provided read_fasta to import the fasta file.
bp	is the biocparallel backend, default to serialParam, most likely sufficient in most scenario
dash_ignore	whether to treat '-' as another type
accepted_char	character to accept, default to c("A", "C", "T", "G")
ignore_case	whether to be case insensitive, default to TRUE

Value

Will return the processed allelic profiles.

`read_fasta` `read_fasta`

Description

`read_fasta` is used to read fasta file, implementation similar to seqinr, but much simpler and allow for spaces in sample name.

Usage

```
read_fasta(file, force_to_upper = TRUE)
```

Arguments

file	file path
force_to_upper	whether to transform sequences to upper case, default to TRUE

Value

Will return list of named character vectors.

write_fasta write_fasta

Description

write_fasta is used to write the named character vectors to fasta file.

Usage

```
write_fasta(seqc, filename)
```

Arguments

seqc	a list containing list of nucleotides. To keep it simple, use provided read_fasta to import the fasta file.
filename	filename of the output file

Value

will write the alignments to file

Index