R **documentation**

of ‘`branch.simpson.Rd`’ etc.

October 5, 2017

`branch.simpson`   `branch.simpson` *is used to calculate 1 or more simpson index and list the position with highest index.*

Description

`branch.simpson` is used to calculate 1 or more simpson index and list the position with highest index.

Usage

```
branch.simpson(seq, level = 1, included = NULL, excluded = NULL,
  numRes = 1)
```

Arguments

`seq`	is fastaDNA object to analysed.
`level`	is the number of positions.
`included`	is the included position for analysed, these will force the computation to compute the simpson’s index at the position no matter what
`excluded`	is the positions that are excluded from computation
`numRes`	is the number of result to be returned.

Value

Will returns simpson’s index and the position.

`Chlamydia_1`   *Normal Fasta sample file Chlamydia sequence type with 1 deletion at 1*

Description

Normal Fasta sample file Chlamydia sequence type with 1 deletion at 1

`Chlamydia_2` *Normal Fasta sample file Chlamydia sequence type with multiple deletions at 1, 3, 6*

Description

Normal Fasta sample file Chlamydia sequence type with multiple deletions at 1, 3, 6

`Chlamydia_mapped` *Normal Fasta sample file Chlamydia sequence types with no deletion*

Description

Normal Fasta sample file Chlamydia sequence types with no deletion

`flagAllele` `flagAllele` *is used to find out a list of allelic profiles that has been flagged and will not be included in computation of minimum SNPs.*

Description

`flagAllele` is used to find out a list of allelic profiles that has been flagged and will not be included in computation of minimum SNPs.

Usage

```
flagAllele(seq)
```

Arguments

`seq`	a list of SeqFastadna. To keep it simple, use read.fasta from seqinr to import the fasta file.

Value

Will return a list of ignored allelic profiles.

`present.percent` — `present.percent` *is used to find present and filter the similarity calculated using* `similar.percent`.

Description

`present.percent` is used to find present and filter the similarity calculated using `similar.percent`.

Usage

```
present.percent(result, percent = 100, number = 100)
```

Arguments

`result`	the result from `similar.percent`.
`percent`	minimum percentage to be included
`number`	number of results to be displayed

Value

Will return the a list of SNPs (as specified) that can be used and the associated percentage at the particular location.

`present.simpson` — `present.simpson` *is used to present the result from the calculation of simpson's index.*

Description

`present.simpson` is used to present the result from the calculation of simpson's index.

Usage

```
present.simpson(seq, result)
```

Arguments

`seq`	is fastaDNA object to analysed.
`result`	is the result from `branch.simpson`.

Value

Will returns the presentation of the result.

processAllele — processAllele *is used to returned the processed allelic profiles.*

Description

`processAllele` is used to returned the processed allelic profiles.

Usage

```
processAllele(seq)
```

Arguments

`seq`	a list of SeqFastadna. To keep it simple, use read.fasta from seqinr to import the fasta file.

Value

Will return the processed allelic profiles.

result — *Result file for validation Result from old software*

Description

Result file for validation Result from old software

similar.percent — similar.percent *is used to find the calculate the percentage of similarity at alleles.*

Description

`similar.percent` is used to find the calculate the percentage of similarity at alleles.

Usage

```
similar.percent(seq, ref)
```

Arguments

`seq`	a list of SeqFastadna. To keep it simple, use read.fasta from seqinr to import the fasta file.
`ref`	the specific allele to be identified.

Value

Will return the a list of SNPs that can be used

similar.simpson — `similar.simpson` *is used to calculate the simpson index and list the position with highest index.*

Description

`similar.simpson` is used to calculate the simpson index and list the position with highest index.

Usage

```
similar.simpson(seq, level = 1, included = NULL, excluded = NULL)
```

Arguments

`seq`	is fastaDNA object to analysed.
`level`	is the number of positions.
`included`	is the included position for analysed, these will force the computation to compute the simpson's index at the position no matter what
`excluded`	is the positions that are excluded from computation

Value

Will returns simpson's index and the position.

simpson.calculate — `simpson.calculate` *is used to calculate the simpson's index given a pattern.*

Description

`simpson.calculate` is used to calculate the simpson's index given a pattern.

Usage

```
simpson.calculate(pattern, N)
```

Arguments

`pattern`	is a pattern, can be a vector or a list.
`N`	is the total number of entities that are in the pattern.

Value

Will returns the simpson's index for the pattern.

simpson.pattern *simpson.pattern is used to generate pattern for calculation at a later stage.*

Description

`simpson.pattern` is used to generate pattern for calculation at a later stage.

Usage

```
simpson.pattern(seq, position, appended = NULL)
```

Arguments

`seq`	is fastaDNA object to analysed.
`position`	is the position of the sequences used to generate pattern.
`appended`	is the pattern that the current operation will appended onto

Value

Will returns the generated pattern.

usualLength *usualLength is used to find out the length of the sequence (W/O deletion).*

Description

`usualLength` is used to find out the length of the sequence (W/O deletion).

Usage

```
usualLength(seq)
```

Arguments

`seq`	a list of SeqFastadna. To keep it simple, use read.fasta from seqinr to import the fasta file.

Value

Will return the maximum length of all the allelic profiles.

Examples

```
sample.case1 <-function(){
#Read the file
Chlamydia <- read.fasta(file='../data/Chlamydia_mapped.txt')

#STEP 1. Process the file
Chlamydia <- processAllele(Chlamydia)
untempered <-read.fasta(file='../data/Chlamydia_mapped.txt')

#Since Chlamydia is normal
checkIdentical(Chlamydia, untempered)

#PERCENT MODE
result=similar.percent(Chlamydia, 'A_D213')
present=present.percent(result, 98, 100)

#All result should have percent higher or equal to 98
for (a in 1:100){
checkTrue(present[[a]]$percent>=98)
}

#SIMPSON MODE
result=branch.simpson(Chlamydia, level=1, numRes=3)
output=present.simpson(Chlamydia, result)

#Should have 3 results
checkTrue(length(output)==3)

checkEquals(output[[1]]$'Index',  0.7344, tolerance=0.00016)
Description=paste('At position:', '1988', sep='-')
checkEquals(output[[1]]$'Description', Description)

checkEquals(output[[2]]$'Index', 0.7318, tolerance=0.00016)
Description=paste('At position:', '2044', sep='-')
checkEquals(output[[2]]$'Description', Description)

checkEquals(output[[3]]$'Index', 0.7266, tolerance=0.00016)
Description=paste('At position:', '2034', sep='-')
checkEquals(output[[3]]$'Description', Description)
}

sample.case2 <-function(){
#Read the file
Chlamydia <- read.fasta(file='../data/Chlamydia_1.txt')

#STEP 1. Process the file
Chlamydia <- processAllele(Chlamydia)

#Since 1 sequence has deletion and is ignored
checkEquals(length(Chlamydia), 55)

#PERCENT MODE
result=similar.percent(Chlamydia, 'A_D213')
present=present.percent(result, 98, 100)

#Should have no result because A_D213 is ignored
```

```
checkEquals(result, NULL)
checkEquals(present, list())

#PERCENT MODE
result=similar.percent(Chlamydia, 'H_S1432')
present=present.percent(result, 98, 100)

#Check result
checkEquals(length(present), 100)
checkEquals(present[[1]]$'position', 171)
checkEquals(present[[1]]$'percent', 100)

#SIMPSON MODE
result=branch.simpson(Chlamydia, level=1, numRes=3)
output=present.simpson(Chlamydia, result)

#Should have 3 results
checkTrue(length(output)==3)

checkEquals(output[[1]]$'Index',  0.7347, tolerance=0.00016)
Description=paste('At position:', '1988', sep='-')
checkEquals(output[[1]]$'Description', Description)

checkEquals(output[[2]]$'Index', 0.7306, tolerance=0.00016)
Description=paste('At position:', '2044', sep='-')
checkEquals(output[[2]]$'Description', Description)

checkEquals(output[[3]]$'Index', 0.7199, tolerance=0.00016)
Description=paste('At position:', '2034', sep='-')
checkEquals(output[[3]]$'Description', Description)
}

sample.case3 <-function(){
#Read the file
Chlamydia <- read.fasta(file='../data/Chlamydia_2.txt')

#STEP 1. Process the file
Chlamydia <- processAllele(Chlamydia)

#Since 1 sequence has deletion and is ignored
checkEquals(length(Chlamydia), 53)

#PERCENT MODE
result=similar.percent(Chlamydia, 'A_D213')
present=present.percent(result, 98, 100)

#Should have no result because A_D213 is ignored
checkEquals(result, NULL)
checkEquals(present, list())

#PERCENT MODE
result=similar.percent(Chlamydia, 'H_S1432')
present=present.percent(result, 98, 100)

#Check result
checkEquals(length(present), 100)
checkEquals(present[[1]]$'position', 171)
checkEquals(present[[1]]$'percent', 100)

#SIMPSON MODE
result=branch.simpson(Chlamydia, level=1, numRes=3)
output=present.simpson(Chlamydia, result)

#Should have 3 results
checkTrue(length(output)==3)

checkEquals(output[[1]]$'Index',  0.7343, tolerance=0.00016)
Description=paste('At position:', '2044', sep='-')
checkEquals(output[[1]]$'Description', Description)

checkEquals(output[[2]]$'Index', 0.7329, tolerance=0.00016)
Description=paste('At position:', '1988', sep='-')
checkEquals(output[[2]]$'Description', Description)

checkEquals(output[[3]]$'Index', 0.7263, tolerance=0.00016)
Description=paste('At position:', '2034', sep='-')
checkEquals(output[[3]]$'Description', Description)
}

test.deactivation <- function()
{
  DEACTIVATED('Deactivating integration test function')
}
```

Index

branch.simpson, 1

Chlamydia_1, 1
Chlamydia_2, 2
Chlamydia_mapped, 2

flagAllele, 2

present.percent, 3
present.simpson, 3
processAllele, 4

result, 4

similar.percent, 4
similar.simpson, 5
simpson.calculate, 5
simpson.pattern, 6

usualLength, 6