Package 'asccdac'

February 3, 2024

Type Package

Title Accelerated Sufficient Condition Conjunction Algorith based on Ant Colony Optimization

Version 1.0

Date 2024-2-1

Author Wei Xu

Maintainer Wei Xu <eric.wxu@hotmail.com>

Description Pkg{asccdac} is a package to perform the Accelerated Sufficient Condition Conjunction Algorith based on Ant Colony Optimization.

License GPL (>= 2)

Encoding UTF-8

LinkingTo Rcpp

Imports Rcpp,plyr,purrr,stringr,admisc,dplyr

RoxygenNote 7.1.0

NeedsCompilation yes

ExperimentalWindowsRuntime ucrt

Archs x64

R topics documented:

`asccdac-package` *Accelerated Sufficient Condition Conjunction Algorith based on Ant Colony Optimization*

Description

Pkgasccdac is a package to perform the Accelerated Sufficient Condition Conjunction Algorith based on Ant Colony Optimization.

Author(s)

Author:
Wei Xu

Maintainer:
Wei Xu <eric.wxu@hotmail.com

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

`aantscc` *Accelerated Ant colony optimization Algorithm Based on Sufficient Condition Conjunction*

Description

This function completes the aantscc analysis.

Usage

```
aantscc(Ant_Num,group,BaseIter,rou,rate,rule,covthreshold,sccsamples)
```

Arguments

`Ant_Num`	Ant population.
`group`	The number of group,the group number corresponds to the order.
`BaseIter`	The base value of the number of iterations.
`rou`	Volatility coefficient.
`rate`	What percentage of the ants don't follow the rules.
`rule`	Set rules for the number of iterations, 0: the number of ants in all order is the same; 1: According to the increase of order, the number of ants increases
`covthreshold`	The threshold of coverage.
`sccsamples`	Dataset.

Value

The aantscc solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
library(stringr)
sccsamples<-data.frame(
V1=c(1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V2=c(1,0,0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0),
V3=c(1,1,1,1,1,0,1,1,1,0,0,0,1,1,0,0,1,1,0,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0),
V4=c(0,1,1,1,1,0,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V6=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
)
threshold=0.01
antnum=500
group=5
rou=0.05
rate=0.1
rule=1
antsccsolution<-aantscc(Ant_Num=antnum,group=group,covthreshold=threshold,sccsamples=sccsamples)
```

`abgscc` *Accelerated Exhaustive Search Algorithm between two groups Based on Sufficient Condition Conjunction*

Description

This function completes the abgscc analysis.

Usage

```
abgscc(consistencythreshold,sccsamples,mvsccsamples,group1,group2)
```

Arguments

`consistencythreshold`	Threadhold of consistency.
`sccsamples`	Dataset.
`mvsccsamples`	Dataset with multivalue coding.
`group1`	A vector of variables in Group1.
`group2`	A vector of variables in Group2.

Value

The abgscc solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
library(purrr)
library(plyr)
library(stringr)
consistencythreshold=0.1;
sccsamples<-data.frame(
V1=c(0,1,0,0,0,0,0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0),
V2=c(0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0),
V3=c(0,0,0,0,1,0,0,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0),
V4=c(1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0)
)
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
)
group1<-c(3,1)
group2<-c(2)
abgsccsolution<-abgscc(consistencythreshold,sccsamples,mvsccsamples,group1,group2)
```

aesascc *Accelerated Exhaustive Search Algorithm Based on Sufficient Condition Conjunction*

Description

This function completes the aesascc analysis.

Usage

```
aesascc(consistencythreshold,maximumofconditions,sccsamples,mvsccsamples)
```

Arguments

`consistencythreshold`	Threadhold of consistency.
`maximumofconditions`	Maximum order of combinations.
`sccsamples`	Dataset.
`mvsccsamples`	Multi-value Dataset.

Value

The aesascc solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
consistencythreshold=0.5
maximumofconditions=3
sccsamples<-data.frame(
V1=c(0,1,0,0,0,0,0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0),
V2=c(0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0),
V3=c(0,0,0,0,1,0,0,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0),
V4=c(1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0)
)
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
)
aesasccsolution<-aesascc(consistencythreshold,maximumofconditions,sccsamples,mvsccsamples)
```

`aisscc` *Accelerated Iterative Screening Method Based on Sufficient Condition Conjunction*

Description

This function completes the single step calculation of aisscc analysis.

Usage

```
aisscc(consistencythreshold,maxnumofconditions,sccsamples,mvsccsamples)
```

Arguments

`consistencythreshold`	Threadhold of consistency.
`maxnumofconditions`	Maximum order of combinations.
`sccsamples`	Dataset.
`mvsccsamples`	Dataset with multivalue coding.

Value

The aisscc solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
library(purrr)
library(plyr)
library(stringr)
consistencythreshold=0.5;
maxnumofconditions=2;
sccsamples<-data.frame(
V1=c(0,1,0,0,0,0,0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0),
V2=c(0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0),
V3=c(0,0,0,0,1,0,0,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0),
V4=c(1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0)
)
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
)
aisscconestep<-aisscc(consistencythreshold,maxnumofconditions,sccsamples,mvsccsamples)
```

asccdac *Accelerated Sufficient Condition Conjunction Algorith based on Ant Colony Optimization*

Description

This function completes the asccdac analysis.

Usage

```
asccdac(conthreshold,covthreshold,antnum,group,rou,rate,rule,sccsamples,mvsccsamples)
```

Arguments

conthreshold	The threshold of consistency.
covthreshold	The threshold of coverage.
antnum	Ant population.
group	The number of group,the group number corresponds to the order.
rou	Volatility coefficient.
rate	What percentage of the ants don't follow the rules.
rule	Set rules for the number of iterations, 0: the number of ants in all order is the same; 1: According to the increase of order, the number of ants increases
sccsamples	Dataset.
mvsccsamples	Dataset.

Value

The asccdac solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
library(stringr)
sccsamples<-data.frame(
V1=c(0,1,0,0,0,0,0,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0),
V2=c(0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0),
V3=c(0,0,0,0,1,0,0,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0),
V4=c(1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0),
V6=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0),
V7=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0),
V8=c(1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0)
)
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2),
V6=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2),
V7=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2),
V8=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
)
conthreshold=0.5
covthreshold=0.02
antnum=500
group=5
rou=0.05
rate=0.1
rule=1
asccdacsolutioncon<-asccdac(conthreshold=conthreshold,covthreshold=covthreshold,antnum=antnum,group=group,
```

`asccdaccon` *Accelerated Sufficient Condition Conjunction Based on Ant colony optimization Algorithm according to Consistency*

Description

This function completes the asccdaccon analysis.

Usage

```
asccdaccon(Ant_Num,group,BaseIter,rou,rate,rule,conthreshold,sccsamples)
```

Arguments

`Ant_Num` Ant population.

`group` The number of group,the group number corresponds to the order.

`BaseIter` The base value of the number of iterations.

`rou` Volatility coefficient.

`rate` What percentage of the ants don't follow the rules.

`rule` Set rules for the number of iterations, 0: the number of ants in all order is the same; 1: According to the increase of order, the number of ants increases

`conthreshold` The threshold of consistency.

`sccsamples` Dataset.

Value

The asccdaccon solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
library(stringr)
sccsamples<-data.frame(
V1=c(1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V2=c(1,0,0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0),
V3=c(1,1,1,1,1,0,1,1,1,0,0,0,1,1,0,0,1,1,0,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0),
V4=c(0,1,1,1,1,0,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V6=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
)
threshold=0.5
antnum=500
group=5
rou=0.05
rate=0.1
rule=1
asccdacconsolution<-asccdaccon(Ant_Num=antnum,group=group,conthreshold=threshold,sccsamples=sccsamples)
```

asccdaccov *Accelerated Sufficient Condition Conjunction Algorithm Based on Ant colony optimization according to Coverage*

Description

This function completes the asccdaccov analysis.

Usage

```
asccdaccov(Ant_Num,group,BaseIter,rou,rate,rule,covthreshold,sccsamples)
```

Arguments

`Ant_Num`	Ant population.
`group`	The number of group,the group number corresponds to the order.
`BaseIter`	The base value of the number of iterations.
`rou`	Volatility coefficient.
`rate`	What percentage of the ants don't follow the rules.
`rule`	Set rules for the number of iterations, 0: the number of ants in all order is the same; 1: According to the increase of order, the number of ants increases
`covthreshold`	The threshold of coverage.
`sccsamples`	Dataset.

Value

The asccdaccov solution.

References

Baumgartner, Michael.(2009). Inferring Causal Complexity. Sociological Methods &amp Research. 38. 10.1177/0049124109339369.

Examples

```
library(stringr)
sccsamples<-data.frame(
V1=c(1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V2=c(1,0,0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0),
V3=c(1,1,1,1,1,0,1,1,1,0,0,0,1,1,0,0,1,1,0,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0),
V4=c(0,1,1,1,1,0,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V5=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V6=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
)
threshold=0.5
antnum=500
group=5
rou=0.05
rate=0.1
rule=1
asccdaccovsolution<-asccdaccov(Ant_Num=antnum,group=group,covthreshold=threshold,sccsamples=sccsamples)
```

`bocorrection` *Bonferroni correction for the Chi-squared test.*

Description

This function corrects Pearson's Chi-squared test by bonferroni correction.

Usage

```
bocorrection(screencom,pvaluevec,numofsnps)
```

Arguments

`screencom`	the snp combination.
`pvaluevec`	the p-value of Pearson's Chi-squared test.
`numofsnps`	the number of SNPs.

Value

A numeric value of pvalue in the pearsons chisquared by bonferroni correction..

References

Benjamini, Y. and D. Yekutieli (2001). The control of the false discovery rate in multiple testing under dependency. The Annals of Statistics. 29: 1165-1188.

Examples

```
library(purrr)
library(plyr)
library(stringr)
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,3,1,1,1,2,1,3),
V6=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
)
screencom<-c(3,4)
pvalue<-chi2test(screencom,mvsccsamples)
numofsnps<-ncol(mvsccsamples)-1
pvaluewithbc<-bocorrection(screencom,pvalue,numofsnps)
```

`chi2test` *Chisquaretest for the Pattern in the Source Data*

Description

This function tests whether the pattern is related to the depend variable in the source dataset by Pearson's Chi-squared test.

Usage

```
chi2test(pattern, samples)
```

Arguments

`pattern`	the pathogenic pattern,for example,"[2,3]" denotes the mutation of the sceond and the third snp.
`samples`	the data of samples.

Value

A numeric value of pvalue in the Pearsons Chi-squared test.

References

Haviland MG. Yates's correction for continuity and the analysis of 2 x 2 contingency tables. Stat Med. 1990 Apr;9(4):363-7; discussion 369-83. doi: 10.1002/sim.4780090403. PMID: 2362976.

Examples

```
library(purrr)
library(plyr)
library(stringr)
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
)
pattern<-c(2,3)
pvalue<-chi2test(pattern,mvsccsamples)
```

`consistency` *Consistency calculation for the Pattern in the Source Data*

Description

This function calculates the consistency in the source dataset.

Usage

```
consistency(pattern, samples)
```

Arguments

`pattern`	the pathogenic pattern,for example,"[2,3]" denotes the mutation of the sceond and the third snp.
`samples`	the data of samples.

Value

A numeric value between 0 and 1.

References

Ragin,C.C.(2008).Redesigning social inquiry: Fuzzy sets and beyond: University of Chicago Press.

Examples

```
samples<-data.frame(
V1=c(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
V2=c(0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0),
V3=c(0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V4=c(1,1,1,1,1,0,0,0,1,0,0,0,0,0,1,1,1,1,1,1,1,0,1,0,1,0,1,0,0,1,0,1,0,1,1,0,1,0),
V5=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
)
pattern<-c(2,3)
consistencyresult<-consistency(pattern=pattern,samples=samples)
```

coverage *Coverage calculation for the Pattern in the Source Data*

Description

This function calculates the coverage in the source dataset.

Usage

```
coverage(pattern, samples)
```

Arguments

`pattern`	the pathogenic pattern,for example,"[2,3]" denotes the mutation of the sceond and the third snp.
`samples`	the data of samples.

Value

A numeric value between 0 and 1.

References

Ragin,C.C.(2008).Redesigning social inquiry: Fuzzy sets and beyond: University of Chicago Press.

Examples

```
samples<-data.frame(
V1=c(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
V2=c(0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0),
V3=c(0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V4=c(1,1,1,1,1,0,0,0,1,0,0,0,0,0,1,1,1,1,1,1,1,0,1,0,1,0,1,0,0,1,0,1,0,1,1,0,1,0),
V5=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
)
pattern<-c(2,3)
coverageresult<-coverage(pattern=pattern,samples=samples)
```

generatecombine *Generate the combinations equal to the max order*

Description

This function generates all the combinations equal to the max order.

Usage

```
generatecombine(conditionalvarvec,maxorder)
```

Arguments

`conditionalvarvec`
: the vector of the conditional variables.

`maxorder` the maximum order specified.

Value

A list of the combinations.

References

Lockwood, Elise,Gibson, Bryan R..Combinatorial tasks and outcome listing: Examining productive listing among undergraduate students[J].Educational Studies in Mathematics.2016,91(2).

Examples

```
library(purrr)
library(plyr)
library(stringr)
conditionalvarvec<-c(1:8)
maxorder=3
combinations<-generatecombine(conditionalvarvec,maxorder)
```

generatecombineall *Generate all the combinations less than or equal to the max order*

Description

This function generates all the combinations less than or equal to the max order.

Usage

```
generatecombineall(conditionalvarvec,maxorder)
```

Arguments

`conditionalvarvec`
: the vector of the conditional variables.

`maxorder` the maximum order specified.

Value

A list of the combinations.

References

Lockwood, Elise,Gibson, Bryan R..Combinatorial tasks and outcome listing: Examining productive listing among undergraduate students[J].Educational Studies in Mathematics.2016,91(2).

Examples

```
library(purrr)
library(plyr)
library(stringr)
conditionalvarvec<-c(1:8)
maxorder=3
allcombinations<-generatecombineall(conditionalvarvec,maxorder)
```

---

generatecombinebetweengroups

Generate the combinations between two groups

Description

This function generates all the combinations between two groups.

Usage

```
generatecombinebetweengroups(group1,group2)
```

Arguments

group1	the vector of the conditional variables in group1.
group2	the vector of the conditional variables in group2.

Value

A list of the combinations.

References

Lockwood, Elise,Gibson, Bryan R..Combinatorial tasks and outcome listing: Examining productive listing among undergraduate students[J].Educational Studies in Mathematics.2016,91(2).

Examples

```
library(purrr)
library(plyr)
library(stringr)
group1<-c(6,1,5,8,10,12)
group2<-c(15,9,3,4)
combinationsbetweengroups<-generatecombinebetweengroups(group1,group2)
```

judgecompatible *Judge Whether a Sample is Compatible with the Pattern*

Description

This function judges whether a sample is compatible with the pattern or not.

Usage

```
judgecompatible(pattern, samples)
```

Arguments

pattern	the pathogenic pattern,for example,"[2,3]" denotes that the mutation of the sceond and third snp.
samples	the data of one sample.

Value

Logical value.

Examples

```
library(purrr)
library(plyr)
library(stringr)
samples<-data.frame(
V1=c(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
V2=c(0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0),
V3=c(0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
V4=c(1,1,1,1,1,0,0,0,1,0,0,0,0,0,1,1,1,1,1,1,1,0,1,0,1,0,1,0,0,1,0,1,0,1,1,0,1,0),
V5=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
)
splitsamples<-llply(transpose(samples), do.call, what = rbind)
sample<-splitsamples[[3]]
pattern<-c(2,3,4)
result<-judgecompatible(pattern=pattern,sample=sample)
```

samplescount *Generate a contingency table for the Source Data*

Description

This function generate a contingency table for the source data.

Usage

```
samplescount(pattern,mvsccsamples)
```

Arguments

`pattern`	the snp combination.
`mvsccsamples`	the mv matrix,0 denotes the missing data,1 denotes homozygous wild-type alleles, 2 denotes homozygous wild-type alleles,3 denotes homozygous mutant alleles.

Value

A contingency table for the source data as matrix.

References

Gravetter, F. J., & Wallnau, L. B. (2010). Essentials of Statistics for the Behavioral Sciences (PSY 200 (300) Quantitative Methods in Psychology). Boston: Cengage Learning.

Examples

```
mvsccsamples<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,3,1,1,1,2,1,3),
V6=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)
)
pattern<-c(2,3)
contingencytable<-samplescount(pattern,mvsccsamples)
```

Index