

Package ‘isscc’

February 1, 2024

Type Package

Title Iterative Screening Method Based on Sufficient Condition
Conjunction

Version 1.0

Date 2023-12-12

Author Wei Xu

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

Description Pkg{isscc} is a package to perform the Iterative Screening method based on Sufficient Condition Conjunction.

License GPL (>= 2)

Imports plyr, purrr, stringr

NeedsCompilation no

R topics documented:

isscc-package	2
antscc	2
bgscc	3
bocorrection	4
chi2test	5
consistency	6
coverage	7
esascc	7
generatecombine	8
generatecombineall	9
generatecombinebetweengroups	10
isscc	11
judgecompatible	12
samplescount	12
Index	14

isscc-package

Iterative Screening Method Based on Sufficient Condition Conjunction

Description

Pkgisscc is a package to perform the Iterative Screening method based on Sufficient Condition Conjunction.

Author(s)

Author:

Wei Xu

Maintainer:

Wei Xu <eric.wxu@hotmail.com>

References

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

antscc

Ant colony optimization Algorithm Based on Sufficient Condition Conjunction

Description

This function completes the antscc analysis.

Usage

```
antscc(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.

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)
snps<-data.frame(
  V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1),
  V2=c(1,3,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
  V3=c(1,1,1,1,1,3,1,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,3,1,3,1),
  V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1),
  V5=c(1,1,2,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,1,3,1,1,1,2,1,3)
)
class<-c(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)
screencom<-c(3,4)
pvalue<-chi2test(screencom,snps,class)
numofsnps<-ncol(snps)
pvaluewithbc<-bocorrection(screencom,pvalue,numofsnps)
```

chi2test	<i>Chisquaretest for the Pattern in the Source Data</i>
----------	---

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(factor,snps,class)
```

Arguments

- factor the snp combination.
- snps the snps matrix,0 denotes the missing data,1 denotes homozygous wild-type alleles, 2 denotes homozygous wild-type alleles,3 denotes homozygous mutant alleles.
- class the disease vector,1 denotes case and 2 denotes control.

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)
snps<-data.frame(
  V1=c(1,2,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1),
  V2=c(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
  V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,3,1,3,1),
  V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1),
  V5=c(1,1,2,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,3,1,1,2,1,3)
)
class<-c(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)
factor<-c(2,3,4)
pvalue<-chi2test(factor,snps,class)
```

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

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


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	<i>Generate all the combinations less than or equal to the max order</i>
--------------------	--

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)
```


Arguments

snps	the snps matrix,0 denotes the missing data,1 denotes homozygous wild-type alleles, 2 denotes homozygous wild-type alleles,3 denotes homozygous mutant alleles.
factor	the snp combination.
class	the disease vector,1 denotes case and 2 denotes control.

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

```
library(purrr)
library(plyr)
library(stringr)
snps<-data.frame(
V1=c(1,2,1,1,1,1,1,1,1,1,3,1,1,1,3,3,1,3,1,1,1,1,1,1,1,2,1,1,1,3,1,1,1,1,1,1,1,1),
V2=c(1,3,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,1,3,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,3,1,3,1),
V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1),
V5=c(1,1,2,1,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,3,1,1,2,1,3)
)
class<-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)
factor<-c(2,3)
contingencytable<-samplescount(snps,factor,class)
```

Index

antscc, [2](#)

bgscc, [3](#)

bocorrection, [4](#)

chi2test, [5](#)

consistency, [6](#)

coverage, [7](#)

esascc, [7](#)

generatecombine, [8](#)

generatecombineall, [9](#)

generatecombinebetweenegroups, [10](#)

isscc, [11](#)

isscc-package, [2](#)

judgecompatible, [12](#)

samplescount, [12](#)