

# Package ‘aesascc’

February 17, 2024

**Type** Package  
**Title** Accelerated Exhaustive Search Algorithm Based on Sufficient  
Condition Conjunction  
**Version** 1.0  
**Date** 2024-1-20  
**Author** Wei Xu  
**Maintainer** Wei Xu <eric.wxu@hotmail.com>  
**Description** Pkg{aesascc} is a package to perform the Accelerated Exhaustive Search Algo-  
rithm based on Sufficient Condition Conjunction(hsascc) Analysis of data.  
**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:

aesascc-package . . . . .	2
aesascc . . . . .	2
chi2test . . . . .	3
consistency . . . . .	4
coverage . . . . .	5
generatecombine . . . . .	5
judgecompatible . . . . .	6
samplescount . . . . .	7
<b>Index</b>	<b>8</b>

---

aesascc-package	<i>Accelerated Exhaustive Search Algorithm Based on Sufficient Condition Conjunction Analysis of the Data</i>
-----------------	---

---

### Description

Pkgesascc is a package to perform the Accelerated Exhaustive Search Algorithm based on Sufficient Condition Conjunction(esascc) Analysis of data.

### 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.

---

aesascc	<i>Accelerated Exhaustive Search Algorithm Based on Sufficient Condition Conjunction</i>
---------	--

---

### Description

This function completes the aesascc analysis.

### Usage

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

### Arguments

consistencythreshold	Threshold of consistency.
coveragethreshold	Threshold of coverage.
maximumofconditions	Maximum order of combinations.
sccsamples	Dataset.
mvsccsamples	Multi-value Dataset.

### Value

The aesascc solution.



## Examples

```
library(purrr)
library(plyr)
library(stringr)
mvscsamples<-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(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
  V3=c(1,1,1,1,3,1,1,1,2,1,3,1,1,1,3,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,3,1,3,1),
  V4=c(2,1,3,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1),
  V5=c(1,1,2,1,1,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)
)
pattern<-c(2,3)
pvalue<-chi2test(pattern,mvscsamples)
```

---

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(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
  V2=c(0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0),
  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),
  V4=c(1,1,1,1,1,0,0,0,1,0,0,0,0,0,1,1,1,1,1,1,0,1,0,1,0,1,0,1,0,1,1,0,1,0),
  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)
)
pattern<-c(2,3)
consistencyresult<-consistency(pattern=pattern,samples=samples)
```



conditionalvarvec	the vector of the conditional variables.
maxorder	the maxorder of combinations.

A list of all combinations.

```
conditionalvarvec<-unlist(1:20)
maxorder<-2
allcombinations<-generatecombine(conditionalvarvec,maxorder)
```

judgecompatible	<i>Judge Whether a Sample is Compatible with the Pattern</i>
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

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

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

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

Logical value.

[illegible]

samplescount

*Generate a contingency table for the Source Data***Description**

This function generate a contingency table for the source data.

**Usage**

```
samplescount(pattern,mvscsamples)
```

**Arguments**

**pattern** the snp combination.

**mvscsamples** 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**

```
mvscsamples<-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(1,3,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,3,1,1,1,2,1,3,1,1,3,1),
  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),
  V5=c(1,1,2,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,1,1,3,1,1,2,1,3),
  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)
)
pattern<-c(2,3)
contingencytable<-samplescount(pattern,mvscsamples)
```

# Index

`aesascc`, [2](#)  
`aesascc-package`, [2](#)  
  
`chi2test`, [3](#)  
`consistency`, [4](#)  
`coverage`, [5](#)  
  
`generatecombine`, [5](#)  
  
`judgecompatible`, [6](#)  
  
`samplescount`, [7](#)