

Package ‘AlgebraicHaploPackage’

January 20, 2025

Type Package

Title Haplotype Two Snips Out of a Paired Group of Patients

Version 1.2

Date 2015-10-26

Author Jan Wolfertz

Maintainer Jan Wolfertz <Jan.wolfertz@uni-duesseldorf.de>

Depends R (>= 3.1.3)

Suggests compiler

Description Two unordered pairs of data of two different snips positions is haplotyped by resolving a small number of closed equations.

LazyLoad yes

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-31 08:46:53

Contents

AlgebraicHaploPackage-package	2
callhaplotype	3
cubic	4
findoptimal	6
haplotypeit	7
optimalfrequency	8

Index	10
--------------	-----------

AlgebraicHaploPackage-package

AlgebraicHaploPackage— The package haplotype a contingency of two pairs of snips of a sample.

Description

Assume a group of patients of an unordered pair of data per snips. The contingency table of 2 snips on different positions are calculated. A haplotype 2x2 most likely contingency table is guessed.

Details

Package: AlgebraicHaploPackage
 Type: Package
 Version: 1.2
 Date: 2015-10-26
 License: GPL2.0

Author(s)

Jan Wolfertz.

Maintainer: jan.wolfertz@uni-duesseldorf.de

References

- [1] David Clayton. "An r package for analysis of whole-genome association studies." *Human Heredity*, 64(1):45 - 51, 2007. doi: doi:10.1001/archgenpsychiatry.2010.25. <http://archpsyc.jamanetwork.com/article.aspx?articleid=111111>
- [2] Nathan Jacobson. *Basic Algebra I: Second Edition* (Dover Books on Mathematics). Dover Publication, 2009. [3] Montgomery Slatkin Laurent Excofie. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular biology and evolution*, 12(5):921 - 927, 1995. URL <http://mbe.oxfordjournals.org/content/12/5/921.full.pdf>. [4] Tianhua Niu. Algorithms for inferring haplotypes. *Genetic Epidemiology*, 27:334347, 2004. doi: DOI: 10.1002/gepi.20024. URL <http://biostat.gru.edu/Journal> [5] Werner A. Stahel. *Statistische Datenanalyse Eine*. Vieweg Verlag, 2002.

Examples

```
print("The second example: \n")
dd=matrix(c(1212, 2, 0, 679, 0,0,75,0,0), byrow=TRUE, nrow=3)
colnames(dd)=c("CC", "CT", "TT")
rownames(dd)=c("CC", "CT", "TT")
callhaplotype(dd)
### Check the result of the cubic equation of the second example
print("#####")
print("Check the result of the cubic equation of the second example: \n")
```

```

temp2haplo =as.numeric(t(dd));
t2h=temp2haplo
haplotypeit(t2h[1],t2h[2],t2h[3],t2h[4],t2h[5],t2h[6],t2h[7],t2h[8],t2h[9]);
rm(temp2haplo)
rm(t2h)
### Third example
print("#####")
print("Third example : \n")
dd3=matrix(c(1030,678,123,1,1,0,0,0,0),ncol=3,byrow=TRUE)
colnames(dd3)=c("AA","AG","GG")
rownames(dd3)=c("CC","CT","TT")
callhaplotype(dd3)
### Check for alternative solutions
print("#####")
print("Check for alternative solutions: \n")
temp2haplo =as.numeric(t(dd3));
t2h=temp2haplo;
haplotypeit(t2h[1],t2h[2],t2h[3],t2h[4],t2h[5],t2h[6],t2h[7],t2h[8],t2h[9]);
rm(temp2haplo)
rm(t2h)
print("#####")
print("#####")
print("This tests the result of the first example of the article \n")
dd2=matrix(c(4,0,0,0,30,0,0,0,23),ncol=3,byrow=TRUE)
callhaplotype(dd2)
callhaplotype(dd2)/(2*57)
print("#####")

```

callhaplotype

calculates the cotigency table of the haplotypes

Description

It starts with a contingency table of pairs of haplotypes and ends up with the haplotypes one. The heterocygote cases are the middle of the column and rows.

Usage

```
callhaplotype(dd)
```

Arguments

dd This is a contingency table. Rows and columns are in the order are AA, AB,BB.

Details

A 2x2 contingency table of haplotypes is calculated. The most likely solution had been choosen.

Value

The haplotype contingency table is returned. All entries are numeric.

Note

The differences are the coice of the solution of the cubic equations. About 4 percent differences and about 7 assuming 1 per thousand. For data export or import you can use a different package.

Author(s)

Jan Wolfertz

References

David Clayton. "An r package for analysis of whole-genome association studies." *Human Heredity*, 64(1):45 - 51, 2007. doi: doi:10.1001/archgenpsychiatry.2010.25. URL <http://archpsyc.jamanetwork.com/article.aspx?articleid=117111>
 Jan Wolfertz(in press.):""

Examples

```
print("#####")
dd2=matrix(c(4,0,0,0,30,0,0,0,23),ncol=3,byrow=TRUE)
callhaplotype(dd2)
callhaplotype(dd2)/(2*57)

### The second example
print("#####")
print("The second example: \n")
dd=matrix(c(1212, 2, 0, 679, 0,0,75,0,0), byrow=TRUE, nrow=3)
colnames(dd)=c("CC", "CT", "TT")
rownames(dd)=c("CC", "CT", "TT")
callhaplotype(dd)
print("#####")
```

cubic

Function that can resolve the cubic equation numerical stable and any lower dimensional case except unsolvable cases.

Description

$A*x^3+B*x^2+C*x+D=0$. All coefficients had to be numeric or integers. This function calculates from 4 coefficient all possible and senfully solutions. $D=0$ returns no values at all. This would be a impossibel case. It returns upto 3 potential complex solutions. Less solutions are copied to get the tripple solution format.

Usage

cubic(A, B, C, D)

Arguments

A	The coefficient of x^3 .
B	The coefficient of x^2 .
C	The coefficient of X .
D	The constant.

Details

This function is called by haplotypeit. The results are returned as vector of the three possible solutions: output[1],output[2],output[3]. Further data for checks of the roots. p,q and the discriminat. 10 and 11 are only usable for symmetry checks.

Value

Returns cubic(A,B,C,D)[c(1:3)] roots of the at most cubic equation.

Note

Using cardenian formular, a well known method.

Author(s)

Jan Wolfertz

References

Cardans formular as in e.g. The Mathematical Gazette (1993); 77 (Nov, No 480), 354-359 (jstor)
<http://www.nickalls.org/dick/papers/math/cubic1993.pdf> or any other book for algebraic solutions.
See also : http://de.wikipedia.org/wiki/Cardanische_Formeln and http://en.wikipedia.org/wiki/Cubic_equation

See Also

haplotypeit,callhaplotype

Examples

```
cubic(1,0,0,-1)[c(1:3)]  
cubic(1,1,0,0)[c(1:3)]
```

 findoptimal

Chose the most likely solution of the three potential onces.

Description

Starting with a 3x3 matrix, three potential haplotypes 2x2 matrices will be calculated and evaluated. The most likely one is chosen. A discrete solution values are not enforced. This reduces increases the right prediction on real data.

Usage

```
findoptimal(A, B, C, D, mmorg, exact = 1e-05)
```

Arguments

A	First entry of the 2x2 matrix.
B	Second numeric entry of the 2x2 matrix.
C	Third numeric entry of the 2x2 matrix.
D	Last numeric entry of the 2x2 matrix.
mmorg	3x3 matrix of the pair of original snip pairs.
exact	exact is a parameter when data are assumed to be equal. Actually 1e-5 is taken. Should not be larger than the inverse of four times the number of people.

Details

It chose the 2x2 model of haplotypes with the smallest prediction error.

Value

AA	Coefficient of x^3
BB	Coefficient of x^2
CC	Coefficient of x
DD	Coefficient of the intercept

Author(s)

Jan wolfertz

Examples

```
dd2=matrix(c(4,0,0,0,30,0,0,0,23),ncol=3,byrow=TRUE)
A=c(38+0i,2+12.1655i,2-12.1655i)
B=c(0+0i,36-12.1655i,36+12.1655i)
C=c(0+0i,36-12.1655i,36+12.1655i)
D=c(76+0i,40+12.1655i,40-12.1655i)
```

haploypeit *Haplotype a 3 x 3 counting matrix.*

Description

This functions recalculates the potential 2x2 haplotype matrices. It gets a 3x3 matrix and returns a list A,B,C,D of vectors. A[1],B[1],C[1],D[1] is the first solution of the matrix. There are always three solutions.

Usage

```
haploypeit(a, b, c, d, e, f, g, h, i)
```

Arguments

a	Number of counts of matching snip pairs.
b	Number of counts of matching snip pairs..
c	Number of counts of matching snip pairs.
d	Number of counts of matching snip pairs.
e	Number of counts of matching snip pairs.
f	Number of counts of matching snip pairs.
g	Number of counts of matching snip pairs.
h	Number of counts of matching snip pairs.
i	Number of counts of matching snip pairs.

Details

The software automatically resolves the cases $e=0$ by circumventing the cubic equation. If the degree of the equation is lower additional copies of some solution will be made to produce the outputformat. The output format is a list of four vectors of coefficients. Each vector contains three complex numbers.

Value

output\$A is a vector of length 3. output\$B, output\$C, output\$D is a list of length 3. One potential solution is A[1],B[1],C[1],D[1].

Author(s)

Jan wolfertz

References

This method refers to an article: David Clayton. An R package for analysis of whole-genome association studies. *Human Heredity*, 64(1):45 - 51, 2007. doi: doi:10.1001/archgenpsychiatry.2010.25. URL <http://archpsyc.jamanetwork.com/article.aspx?articleid=210679>.

See Also

callhaplotype

Examples

```
haplotypeit(4,0,0,0,30,0,0,0,23)
print("#####")
print("This tests the cubic routine")
haplotypeit(4,0,0,0,30,0,0,0,23)
### Formated of 4 digits
print("Formated of 4 digits")
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$A)),digit=4)
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$B)),digit=4)
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$C)),digit=4)
round(as.numeric(Re(haplotypeit(4,0,0,0,30,0,0,0,23)$D)),digit=4)
###
```

optimalfrequency *Evaluate potential haplotypes.*

Description

Calculate the difference of a known haplotype and the resulting unordered pair of snip pairs.

Usage

```
optimalfrequency(mm, mmorg)
```

Arguments

mm	This is a contingency table of two haplotype snips. 2x2 matrix or ndata.frame
mmorg	This is a contingency table of two diplotype snips pairs.

Details

The average squared distance to the expected result 3x3 table is used as a T statistic. The p value not to be zero is calculated. The higher the p value the more exact is the haplotype.

Value

A list of values is returned.

result\$ <i>LK</i>	Linkage disequilibrium
result\$ <i>Testvalue</i>	The squared sitance multiplied by the number of entries in 3x3 matrix mmorg
result\$ <i>prSimilarByChange</i>	The probability not o be equal to zero by change.

optimalfrequency

9

Author(s)

Jan wolfertz.

References

Stahel: Statistik fuer Naturwissenschaftler und Mediziner, pp. 107-120.

See Also

findoptimal

Index

- * **'model-fit'**
 - optimalfrequency, 8
 - * **cubic**
 - callhaplotype, 3
 - cubic, 4
 - * **functions**
 - cubic, 4
 - * **haplotypeit**
 - callhaplotype, 3
 - * **haplotype**
 - optimalfrequency, 8
 - * **haplotyping**
 - callhaplotype, 3
 - findoptimal, 6
 - haplotypeit, 7
 - * **package**
 - AlgebraicHaploPackage-package, 2
- AlgebraicHaploPackage
(AlgebraicHaploPackage-package),
2
- AlgebraicHaploPackage-package, 2
- callhaplotype, 3
cubic, 4
- findoptimal, 6
- haplotypeit, 7
- optimalfrequency, 8