

# Package ‘bimetallic’

February 19, 2015

**Type** Package

**Title** Power for SNP analyses using silver standard cases

**Version** 1.0

**Date** 2011-08-03

**Author** Andrew McDavid

**Maintainer** Andrew McDavid <amcdavid@fhcrc.org>

**Description** A power calculator for Genome-wide association studies (GWAs) with combined gold (error-free) and silver (erroneous) phenotyping per McDavid A, Crane PK, Newton KM, Crosslin DR, et al. (2011)

**License** GPL-2

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2012-10-29 08:58:17

**NeedsCompilation** no

## R topics documented:

bimetallic-package . . . . .	2
allelicOR . . . . .	3
bimetallic-internal . . . . .	3
Power of Chi-square tests . . . . .	4

<b>Index</b>	<b>7</b>
--------------	----------

---

bimetallic-package      *Power for SNP analyses using silver standard cases*

---

## Description

A power calculator for Genome-wide association studies (GWAs) with combined gold (error-free) and silver (erroneous) phenotyping per McDavid A, Crane PK, Newton KM, Crosslin DR, et al. (2011)

## Details

Package:	bimetallic
Type:	Package
Version:	1.0
Date:	2011-08-03
License:	GPL-2
LazyLoad:	yes

[chisq.sim.factory](#), [chisq.lytic.func](#) and [dlambda](#), [is.legal](#) all use common parameters. [chisq.sim.factory](#) returns a function that repeatedly calculates chi-square statistics ( $X^2$ ) and point estimates of allelic odds ratios ([allelicOR](#)) for a study specified by the simulation parameters. [chisq.lytic.func](#) returns lambda, the non-centrality parameter of the asymptotic sampling distribution of ( $X^2$ ), which happens to be non-central chi-square with two degrees of freedom. [dlambda](#) returns the derivative of lambda with respect to `gam_ca` or `gam_co`. If this is positive then power is increasing with the addition of a silver-standard subject. [is.legal](#) determines if the parameters provided are 'legal' in the sense of inducing valid probabilities in affected and unaffected populations in the simulation

## Author(s)

Andrew McDavid

Maintainer: Andrew McDavid <amcdavid@fhrc.org>

## References

See McDavid A, Crane PK, Newton KM, Crosslin DR, et al. (2011) for a full description of the model and parameters.

## Examples

```
##Make a chisq simulator under a study scenario
sim = chisq.sim.factory(R = 4, gam_ca = 3, gam_co = 0,
  ppv = .8, npv = 1, homRR = 2.2, N_co = 1000,
  maf = .1, prev = .01, model = "mult")
```

```
##Run one realization of the simulation
sim()
```

---

allelicOR	<i>Calculate an odds ratio from a 2x3 contingency table of genotype X phenotype</i>
-----------	---

---

### Description

Find the odds ratio, per allele, in biallelic marker in a diploid individual. The locus must be under Hardy-Weinburg at the margin, ie, alleles are independent, for this calculation to hold.

### Usage

```
allelicOR(xtable)
```

### Arguments

xtable            A 2x3 contingency table of frequencies or counts of individuals.

### Details

The OR is calculated as  $(2 \cdot \#AA.Case + \#Aa.Case) \cdot (2 \cdot \#AA.Control + \#Aa.Control) / ((2 \cdot \#aa.Case + \#Aa.Case) \cdot (2 \cdot \#aa.Control + \#Aa.Control))$

### Value

The odds ratio, floating point.

### Author(s)

Andrew McDavid

### See Also

[chisq.sim.factory](#), ~~~

### Examples

```
cont.table = matrix(c(5, 4, 1, 4, 3, 4), nrow=2, ncol=3, byrow=TRUE)
allelicOR(cont.table)
```

---

bimetallic-internal	<i>bimetallic internal functions</i>
---------------------	--------------------------------------

---

### Description

Internal functions in package

---

 Power of Chi-square tests

*Power of Chi-square tests*


---

**Description**

Produce a function that returns a simulator of chi-square statistics and odds ratios, determine if a set of parameters is legal, return the non-centrality parameter, lambda, of the chi-square distribution under a set of parameters, take a derivative of lambda with respect to number of silver-standard samples under a set of parameters.

**Usage**

```
chisq.sim.factory(R, gam_ca, gam_co, ppv, npv,
  homRR, N_co, maf, prev, model)
is.legal(R, gam_ca, gam_co, ppv, npv,
  homRR, N_co, maf, prev, model)
chisq.lytic.func(R, gam_ca, gam_co, ppv, npv,
  homRR, N_co, maf, prev, model)
dlambda(R, gam_ca, gam_co, ppv, npv,
  homRR, N_co, maf, prev, model, diff="gam_ca")
```

**Arguments**

R	Ratio of gold standard controls to gold standard cases
gam_ca	Ratio of silver standard cases to gold standard cases
gam_co	Ratio of silver standard controls to gold standard controls
ppv	The positive predictive value of the criteria used to identify silver standard cases, ie, $P(\text{Affected} \mid \text{Case})$
npv	The negative predictive value of the criteria used to identify silver standard controls, ie, $P(\text{Unaffected} \mid \text{Control})$
homRR	The relative risk of a homozygous genotype, ie, $P(\text{Affected} \mid \text{geno=AA}) / P(\text{Affected} \mid \text{geno=aa})$
N_co	The number of gold-standard controls
maf	The minor allele frequency of the risk locus
prev	Disease prevalence, ie, $P(\text{Affected})$
model	Disease risk model, either one of 'dominant', 'recessive', 'multiplicative' or number giving the heterozygous relative risk.
diff	Take the derivative of lambda with respect to 'gam_ca' (default) or 'gam_co'

## Details

This function simulates the behavior of chi-square tests for independence in a genome-wide association study consisting of two cohorts. The first cohort, the *gold standard* cohort, is assumed to have been classified into affected and unaffected without error. The second cohort, the *silver standard* cohort, is assumed to have errors in disease classification subject to the npv and ppv arguments. A distinction is drawn between *affected*, *unaffected* and *case* and *control*. Affected and unaffected are the true disease status, which is observed in the gold standard cohort. In the silver standard cohort the case/control criteria are observed instead, while the affected/unaffected status is latent.

The numbers of gold and silver standard cases and controls are set by N\_co and the ratios R, gam\_co and gam\_ca. The genotype frequencies in the gold standard cohort are governed by a genotypic disease risk model and the parameters homRR, maf, prev and model. The model argument may be a character vector of length one, either 'dominant', 'recessive', 'multiplicative' (or an unambiguous abbreviation thereof), which links the heterozygous relative risk  $P(\text{Affected} | Aa) / P(\text{Affected} | aa)$  to homRR, or it may be a floating point value directly specifying the heterozygous relative risk.

These functions offer a way to simulate power as well as calculate it asymptotically. The distribution of chi-square statistics under the disease risk model, ppv, npv and case/control ratios is asymptotically distributed non-central chi-square. chisq.lytic.func returns the non-centrality parameter associated with the model. dlambd returns the first derivative of lambda. Since power is increasing in lambda, if dlambd is positive, then an additional silver standard case (if diff="gam\_ca") increases power.

## Value

chisq.sim.factory returns a argument-less function that may be repeatedly called to sample from the simulated distribution. This function returns a vector of length 2, consisting of the chi-square statistic and the point estimate of the allelic odds ratio. Parameters are not checked for legality.

is.legal returns a boolean value indicating if the disease risk model parameters are legal, ie, induce genotype probabilities on [0,1], conditional on affected/unaffected status.

chisq.lytic.func returns *lambda*, the non-centrality parameter of the asymptotic sampling distribution of the chi-square test under the model.

dlambd returns the first derivative of lambda with respect to gam\_ca or gam\_co

## Author(s)

Andrew McDavid <amcdavidatfhcrcdotorg>

## References

McDavid A, Crane PK, Newton KM, Crosslin DR, et al. (2011) The Potential to Enhance Power of Genetic Association Studies through the Use of Silver Standard Cases Derived from Electronic Medical Records: Single Nucleotide Polymorphism Associations with Dementia in the Electronic Medical Records and Genomics (eMERGE) Network

## See Also

[dlambd](#)

**Examples**

```

##Make a chisq simulator under a study scenario
sim = chisq.sim.factory(R = 4, gam_ca = 3, gam_co = 0,
  ppv = .8, npv = 1, homRR = 2.2, N_co = 1000,
  maf = .1, prev = .01, model = "mult")

##Run one realization of the simulation
sim()

##Run 100 realizations of the simulation
times = 100
chisq_or = as.data.frame(t(replicate(times, sim())))

##Find the number of times chisq_or$stat.X2[i] exceeded .01 significance
sig = .01
critval = qchisq(1-sig, 2)
nsucess = sum(chisq_or$stat > critval)

##Find the power
nsucess/times

##Compare to asymptotic
lambda = chisq.lytic.func(R = 4, gam_ca = 3, gam_co = 0,
  ppv = .8, npv = 1, homRR = 2.2, N_co = 1000,
  maf = .1, prev = .01, model = "mult")

1-pchisq(qchisq(1-sig, 2), 2, ncp=lambda)

## generate a multifactorial design
paramset = list(R = c(.5, 1, 2, 4), gam_ca = c(0, 1), gam_co = 0,
  ppv = c(.4, .6, .8), npv = .8, homRR = c(1.4, 3, 9), N_co = 1000,
  maf = .1, prev = .01, model = "mult")
paramgrid = do.call(expand.grid, c(paramset, stringsAsFactors=FALSE))

##call chisq.lytic and dlambd for each experiment
lambda = vector()
dl = vector()
for( i in 1:nrow(paramgrid)){
  lambda[i] = do.call(chisq.lytic.func, paramgrid[i,])
  dl[i] = do.call(dlambd, paramgrid[i,])
}

##bind it to the parameter data.frame
##calculate the difference in lambda for gam_ca=0 vs gam_ca=1
paramgrid = cbind(paramgrid, lambda, dl)
param0 = subset(paramgrid, gam_ca==0)
param1 = subset(paramgrid, gam_ca==1)
all(paste(param0[,c(1, 3:10)])==paste(param1[,c(1, 3:10)]))
param0 = cbind(param0, finite_diff_dl = param1$lambda - param0$lambda)

##do they agree?
with(param0, cor(dl, finite_diff_dl))

```

# Index

## \*Topic **package**

- bimetallic-package, 2
  
- allelicOR, 2, 3
  
- bimetallic (bimetallic-package), 2
- bimetallic-internal, 3
- bimetallic-package, 2
  
- chisq.lytic.func, 2
- chisq.lytic.func (Power of Chi-square tests), 4
- chisq.setup (bimetallic-internal), 3
- chisq.sim.factory, 2, 3
- chisq.sim.factory (Power of Chi-square tests), 4
  
- dlambda, 2, 5
- dlambda (Power of Chi-square tests), 4
  
- is.legal, 2
- is.legal (Power of Chi-square tests), 4
  
- Power of Chi-square tests, 4
  
- rmultnom (bimetallic-internal), 3
  
- trueOR (bimetallic-internal), 3