markerSearchPower-internal
Internal functions

Description

Internal markerSearchPower functions.

Usage

```
CDF.F3l1(x, c, d)
CDF.F4l3(x)
CDF.F34(x)
CDF.T3(x)
covT1l3T2l3Func(b1, b2, b3, p1, p2, sigm)
covT1T1l3Func(b1, b2, b3, p1, p2, sigm)
covT1T2Func(b1, b2, b3, p1, p2, sigm)
covT1T2l3Func(b1, b2, b3, p1, p2, sigm)
covT12T1Func(b1, b2, b3, p1, p2, sigm)
meanF3l1Func(b1, b2, b3, p1, p2, p3, sigm)
meanT1Func(b1, b2, b3, p1, p2, sigm, n)
meanT1l3Func(b1, b2, b3, p1, p2, sigm, n)
meanT12Func(b1, b2, b3, p1, p2, sigm, n)
PDF.F34(x)
power.exhaustive.atleast1(meanT12, meanT1, meanT2, covMT12s, c3l1,
                          d3l1, c3l2, d3l2, R, p, n, samplePointN)
power.exhaustive.both(meanT12, meanT1, meanT2, covMT12s, c3l1, d3l1,
                      c3l2, d3l2, R, p, n, samplePointN)
power.forward.atleast1(meanT1, meanT2, meanT1l3, meanT2l3, covMTvector,
                       R, p, samplePointN, decompMethod)
power.forward.both(meanT12, meanT1, meanT2, covMT12s, c3l1, d3l1,
                   c3l2, d3l2, R, p, n, samplePointN)
power.marginal(meanT1, meanT2, covMT1T2, R, p, isBoth, samplePointN)
tripleSet(r, p, F34N)
tripleSet2(r, p1, p2, F34N)
varF3l1Func(b1, b2, b3, p1, p2, p3, sigm)
varT1Func(b1, b2, b3, p1, p2, sigm)
varT1l3Func(b1, b2, b3, p1, p2, sigm)
varT12Func(b1, b2, b3, p1, p2, sigm)
```

Details

These functions are not for use at user level.

Value

`CDF.F3l1, CDF.F4l3, CDF.F34, CDF.T3` give discribution functions. `PDF.F34` gives density functions. `meanF3l1Func, meanT1Func, meanT1l3Func, meanT12Func` give means

for relevant asymptotic distributions. `varF3l1Func, varT1Func, varT1l3Func, varT12Func` give variances for relevant asymptotic distributions. `covT1l3T2l3Func, covT1T1l3Func, covT1T2Func, covT1T2l3Func, covT12T1Func` give covariances for relevant asymptotic distributions. `power.xx` give calculated power values. `tripleSet, tripleSet2` return all possible distributions in three groups for finite samples.

Author(s)

Zheyang Wu

References

Zheyang Wu and Hongyu Zhao (2009) *Statistical Power of Model Selection Strategies for Genome-Wide Association Studies*. Submitted

---

`markerSearchPower-package`

Package of power calculation for marker detection strategies

Description

This package calculates statistical power of detecting associated markers based on one of the model selection strategies: marginal selection, exhaustive search, or forward selection. With assumed genetic effects (including interaction effect), allele frequencies, noise level, sample size, number of genotyped markers, and control level (i.e. number of markers or models intended to select), this package provides fast and accurate consultation on power of different model selection strategies. It helps researchers to decide a more efficient way for marker detection in genome-wide association studies.

Details

Package:	markerSearchPower
Type:	Package
Version:	1.0
Date:	2009-02-11
License:	GPL-2

This package requires three other R packages: mvtnorm, adapt, corpcor.

Installation from local files: For windows: Download the .zip file into your local hard drive. Open Rgui (R graphical interface), go to menu "Packages or install packages from local (zip) files ...", then select the provided zip file. Go to menu "Packages or load package..." to load the package into R for use. For Linux or Unix: Download the .tar.gz file into your local hard drive. Install the R package from the source file using the following commands: first go to the directory where the downloaded file is stored and then use "R CMD INSTALL <filename>.tar.gz". For more options please see R help on "INSTALL".

Very easy to use. Only one external function. Choose proper parameters and methods for

function `markerSearchPower`, which returns a power value.

Author(s)

Zheyang Wu and Hongyu Zhao

Maintainer: Zheyang Wu <zheyang.wu@yale.edu>

References

Zheyang Wu and Hongyu Zhao (2009) *Statistical Power of Model Selection Strategies for Genome-Wide Association Studies*. Submitted.

Examples

```
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=2.4, noiseSD=sqrt(3), alleleFreq1=0.7,
            alleleFreq2=0.7, strategy = "marginal", powerDef = "both", DetectionN=10,
            obsN=1000, markerN=300000);
```

`markerSearchPower` *Function of power calculation*

Description

Analytically calculate statistical power for three model selection methods in genome-wide association studies: marginal selection, exhaustive search, and forward selection. Power definitions are (A) the probability of identifying exactly the true model (in marginal selection, it is the probability of detecting both true markers); and (b) the probability of detecting at least one of the true markers.

Usage

```
markerSearchPower(mainEff1, mainEff2, epistasisEff, noiseSD, alleleFreq1, alleleFreq2,
                  alleleFreq3 = 0.5, DetectionN, obsN, markerN,
                  strategy = c("marginal", "exhaustive", "forward"),
                  powerDef = c("both", "either"), samplePointN = 20000,
                  decompMethod = c("svd", "chol", "eigen"))
```

Arguments

`mainEff1`	The main effect of associated marker 1. It is represented by b1 in the true genetic model. See Details below.
`mainEff2`	The main effect of associated marker 2. It is represented by b2 in the true genetic model. See Details below.
`epistasisEff`	The epistatic effect (interaction) between markers 1 and 2. It is represented by b3 in the true genetic model.
`noiseSD`	The standard deviation of noise term.

`alleleFreq1`	The frequency of the effective allele at marker 1.
`alleleFreq2`	The frequency of the effective allele at marker 2.
`alleleFreq3`	Allele frequency of other non-associated markers. This value has little effect on power calculation.
`DetectionN`	The number of top models selected. For power of detecting both associated markers in marginal selection, `DetectionN >= 2`.
`obsN`	Sample size.
`markerN`	Number of markers genptyped.
`strategy`	The stratgy of marker searching: "marginal", "exhaustive", or "forward". Default is "marginal".
`powerDef`	Definition of power. Either "all" indicates the probability of identifying exactly the true model (in marginal selection, it is the probability of detecting both true markers); or "either" indicates the probability of detecting at least one of the true markers.
`samplePointN`	Number of random points for Monte Carlo integration. Default is 20,000.
`decompMethod`	Matrix decomposition used to determine the matrix root of sigma, possible methods are singular value decomposition ("svd", default), eigenvalue decomposition ("eigen"), and Cholesky decomposition ("chol"). This argument is only useful for forward selection identifying either true SNP.

Details

The power calculation is based on a hypothetical true underlying genetic model of a two biallelic marker system for a quantitative trait: y = b0 + b1*X1 + b2*X2 + b3*X1*X2 + noise. Noise has normal distribution with mean zero and standard deviation `noiseSD`. X1 and X2 are the genotypes of markers 1 and 2, respectively. Genotype values are based on the assumption of additive allelic effect: it is equal to the number of copies of the effective allele. For the main covariates, the signs of coefficients b1 and b2 illustrate the direction of effective alleles of markers 1 and 2. The coefficient b3 for the interaction term represents the direction and magnitude of epistasis. The power calculation for binary trait is under development.

Value

The power of a given marker search strategy under a given definition either (A) or (B).

Author(s)

Zheyang Wu and Hongyu Zhao (Maintainer: Zheyang Wu <zheyang.wu@yale.edu>)

References

Zheyang Wu and Hongyu Zhao (2009) *Statistical Power of Model Selection Strategies for Genome-Wide Association Studies*. Submitted.

Examples

```
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=2.4, noiseSD=sqrt(3),
     alleleFreq1=0.7, alleleFreq2=0.7, DetectionN=2, obsN=100, markerN=300);
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=2.4, noiseSD=sqrt(3),
     alleleFreq1=0.7, alleleFreq2=0.7, DetectionN=1, obsN=100, markerN=300,
     powerDef = "either");
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=1.4, noiseSD=sqrt(3),
     alleleFreq1=0.7, alleleFreq2=0.7, DetectionN=1, obsN=100, markerN=300,
     strategy = "exhaustive", powerDef = "either");
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=1.4, noiseSD=sqrt(3),
     alleleFreq1=0.7, alleleFreq2=0.7, DetectionN=1, obsN=100, markerN=300,
     strategy = "exhaustive", powerDef = "both");
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=1.4, noiseSD=sqrt(3),
     alleleFreq1=0.7, alleleFreq2=0.7, DetectionN=5, obsN=100, markerN=300,
     strategy = "forward", powerDef = "both");
markerSearchPower (mainEff1=0.1, mainEff2=0.1, epistasisEff=1.4, noiseSD=sqrt(3),
     alleleFreq1=0.7, alleleFreq2=0.7, DetectionN=5, obsN=100, markerN=300,
     strategy = "forward", powerDef = "either");
```