Package ‘hegp’

May 26, 2020

Type Package

Title Homomorphic Encryption of Genotypes and Phenotypes

Version 0.1.0

Imports parallel, mixed.model.gwas

Author Richard Mott

Maintainer The package maintainer <yourself@somewhere.net>

Description Uses random orthogonal matrices to homomorphically encrypt phenotypes and genotypes for quantitative genetic analysis.

License GPL

Encoding UTF-8

LazyData true

R topics documented:

`basic.gwas` *basic.gwas*

Description

Perform a standard genome wide association analysis. Used to check that plaintext and ciphertext data produce the same gwas results.

Usage

```
basic.gwas( D, mc.cores=10 )
```

Arguments

`D`	A Dataset
`mc.cores`	Number of cores for parellelisation

Details

Each vector of SNP dosages is tested for assocation with the phenotype by simple linear regression.

Value

A dataframe containing the logP of each tested SNP joined to the columns of D$map

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

`basic.mm.gwas` *Mixed Model GWAS*

Description

Perform a mixed-model GWAS to check that plaintext and ciphertext produce the same results.

Usage

```
basic.mm.gwas(D, mc.cores=10)
```

Arguments

`D`	A Dataset
`mc.cores`	Number of cores over which to parallelize computation

Details

A standard mixed model is fitted to the data, using a SNP-based genetic reelationship matrix. The phenotype and genotype are then transformed and each transformed SNP is tested for association with the transformed phenotype. Uses the function mixed.model.gwas from the package mixed.model.gwas

Value

A dataframe containing the logP of each tested SNP joined to the columns of D$map

Author(s)

Richard Mott

References

Mott eet al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

---

build.D *build.D*

Description

Create a Dataset object from its constituent components

Usage

```
build.D( y, dosages, cov=NULL, map=NULL, kinship=FALSE )
```

Arguments

y	Numeric phenotype vector
dosages	Matrix of genotype dosages
cov	Optional matrix of covariates
map	Optional data frame of information about genotypes. If supplied, the i'th row of map refers to the i'th column of the genotype dosages.
kinship	Optional switch to generate a geneetic relationship matrix from the genotype dosages

Value

A list with the components y=y.s, geno=geno, cov=cov, map=map, maf=af

y	vector of phenotypes, scaled to have zero mean and variance equal to one
geno	matrix of genotype dosages, each column (SNP) scaled to have zero mean and variance equal to one

`cov`	matrix of covariates. If the input covariate matrix is NULL this is a vector of ones
`map`	optional dataframe of information about SNPs, e.g. chromosome and base-pair coordinate
`af`	allele frequencies of the SNPs, computed from the genotype dosages
`kinship`	optional genetic relationship matrix

Note

No missing values are allowed. The dimensions of the phenotypes and genotypes are made compatible by matching the rownames of the genotypes with the names of the phenotypes. If a genetic relationship matrix is calculated it uses the function make.kinship from the library mixed.model.

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

`encrypt.D` *encrypt.D*

Description

Encrypt or decrypt a dataset.

Usage

```
encrypt.D( D, encrypter, invert=FALSE, kinship=FALSE )
```

Arguments

`D`	A Dataset to be encrypted
`encrypter`	An encrypter as generated by a call to make.encrypter
`invert`	Decrypt the data by using the inverse (matrix transpose)
`kinship`	An optional kinship matrix of dimension $N * N$, which if supplied will be encrypted as well

Value

An encrypted or decrypted Dataset derived from the input data by applying thee encryptor to it

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

`make.encrypter` *Create encryption keys for a Dataset*

Description

Sample a series of orthogonal matrices suitable for encrypting a given Dataset object.

Usage

```
make.encrypter( D, blocksize=0 )
```

Arguments

`D`	A Dataset object
`blocksize`	Optional size of encryption blocks. Each block of individuals is encrypted separately. If blocksize is zero then a single encrypter is generated.

Details

Create random orthogonal encryption keys for the dataset D. Each encryption key is a random orthogonal matrix generated from the Steifel manifold. If the dataset contains N individuals then if $blocksize > 0$, $N/blocksize + 1$ keys are generated. Most keys are of dimension $blocksize * blocksize$ with the final key with smaller dimension to make the sum of the dimensions of the keys equal to N.

Value

A list with elements

`blocks`	The number of blocks
`block`	a list of encryption keys, each a matrix

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

---

qnorm.D *qnorm.D*

Description

Replace a phenotype and each vector of genotype dosages by their Normal quantiles

Usage

```
qnorm.D(D,digits=NA)
```

Arguments

D	A Dataset object
digits	Optionally truncate the digits of the quantiles, if digits>0

Details

The phenotype `D$y` and each column of the genotype matrix `D$genos` are replaced by a permutation of the standard Normal quantiles, to improve the security of the encryption. If digits>0 then in addition only the first few decimal digits of each quantil are kept. If digits=NA then no truncation is performed.

Value

A Dataset object with transformed phenotype and genotypes. Other elements of the input Dataset are copied verbatim.

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

`rustiefel` *rustiefel*

Description

Simulate a random orthogonal matrix of dimensions $m * R$ using the Steiefel manifold

Usage

```
rustiefel(m, R=m)
```

Arguments

m	the number of rows of the simulated matrix
R	thee number of columns

Details

Function adapted from R package rsteifel.

Note

Function adapted from R package rsteifel.

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

safe.scale *Scale the phenotypes and genotypes in a Dataset*

Description

Scale the phenotypes and genotypes in a Dataset, safely taking into account the possibility a genotype may have zero variance.

Usage

```
safe.scale(mat)
```

Arguments

mat	A numeric matrix with no missing values

Details

Scales the columns of mat by substracting the column mean and dividing by the column standard deviation. If the standard deviation is zero the column is set to zero.

Value

A matrix with the same dimensions as mat in which each column has been scaled.

Author(s)

Richard Mott

References

Mott et al Genetics 2020

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

Index