Package: mvnpermute (via r-universe)

August 27, 2024

Type Package

Title Generate New Multivariate Normal Samples from Permutations

Version 1.0.1

Date 2022-04-11

Description Given a vector of multivariate normal data, a matrix of covariates and the data covariance matrix, generate new multivariate normal samples that have the same covariance matrix based on permutations of the transformed data residuals.

License GPL (>=3.0)

Imports stats

Encoding UTF-8

RoxygenNote 7.1.2

URL https://github.com/markabney/MVNpermute

Repository https://markabney.r-universe.dev

RemoteUrl https://github.com/markabney/mvnpermute

RemoteRef HEAD

RemoteSha 30e444dc812baf1ba64a1923b54cce247f004f4e

Contents

mvnpermute 2

Index **4**

`mvnpermute` *Generate Permutation-based Multivariate Normal Data*

Description

Simulate new multivariate normal data sets with a specified covariance matrix by permutation. Permutations are done on a linear transformation of residuals that is guaranteed to be exchangeable when the original data are multivariate normally distributed.

Usage

```
mvnpermute(y, X, S, nr, seed)
```

Arguments

`y`	Vector of multivariate normal data.
`X`	Matrix or data frame of covariate values (including the intercept term if desired). Number of rows must equal the length of `y`.
`S`	Known or estimated covariance matrix of `y`.
`nr`	Number of data sets to generate.
`seed`	Optional random number seed for sampling the `nr` permutations.

Details

This function takes multivariate normal data with known covariates and covariance matrix and generates "permutations" of this data that maintain the mean and covariance of the original data. The permutations are generated by finding the residuals of the data and mapping the residuals to an orthonormal space, then simulating random permutations within this orthonormal space. The permutations are then mapped back to the original space, and the expected values (given the covariates, and their estimated effect sizes) are added back in. The resulting "permuted" data can now be used exactly like the original data; for example, you could estimate the null distribution of some test statistic while maintaining the (known) covariance of the data.

Value

A matrix with number of rows equal to `length(y)` and `nr` columns.

References

Abney M (2015) "Permutation testing in the presence of polygenic variation." Genetic Epidemiology, in press.

Abney M, Ober C, McPeek MS (2002). "Quantitative trait homozygosity and association mapping and empirical genome-wide significance in large complex pedigrees: Fasting serum insulin levels in the Hutterites." American Journal of Human Genetics 70: 920–934.

Examples

```
set.seed(98765)
N <- 100
r.mat <- matrix(rexp(N^2), ncol=N)
cv.mat <- crossprod(r.mat) # a positive definite covariance matrix
x.mat <- rep(1, N)
r.mean <- 3
r1 <- rnorm(N) # independent random noise
r.data <- x.mat * r.mean + drop( t(chol(cv.mat)) %*% r1 ) # correlated random noise
r.perm <- mvnpermute( r.data, x.mat, cv.mat, nr=1000, seed=1234)
est.cvm <- cov(t(r.perm))
plot(cv.mat, est.cvm); abline(0,1)
```

Index

mvnpermute, 2