Package: OmicKriging (via r-universe)

September 13, 2024

Type Package

Title Poly-Omic Prediction of Complex TRaits

Version 1.4.0

Date 2016-03-03

Author Hae Kyung Im, Heather E. Wheeler, Keston Aquino Michaels, Vassily Trubetskoy

Maintainer Hae Kyung Im <`haky@uchicago.edu`>

Description It provides functions to generate a correlation matrix from a genetic dataset and to use this matrix to predict the phenotype of an individual by using the phenotypes of the remaining individuals through kriging. Kriging is a geostatistical method for optimal prediction or best unbiased linear prediction. It consists of predicting the value of a variable at an unobserved location as a weighted sum of the variable at observed locations. Intuitively, it works as a reverse linear regression: instead of computing correlation (univariate regression coefficients are simply scaled correlation) between a dependent variable Y and independent variables X, it uses known correlation between X and Y to predict Y.

License GPL (>=3)

Depends R(>= 2.15.1), doParallel

Imports ROCR, irlba, parallel, foreach

Collate 'correlation_matrices.R' 'input_pheno_GT.R' 'omic_KRIGR.R'

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

RemoteUrl https://github.com/hakyimlab/omickriging

RemoteRef HEAD

RemoteSha 48edf855f1bd9078d98134acd0c7b8b946858d86

Contents

krigr_cross_validation 2
load_sample_data 3
make_GXM 3
make_PCs_irlba 4
make_PCs_svd 5
okriging 6
read_GRMBin 7
write_GRMBin 8

Index 9

`krigr_cross_validation`

Multithreaded cross validation routine for Omic Kriging.

Description

This is a flexible cross validation routine which wraps the Omic Kriging calculation. The user can specify the size of the test set, all the way to "Leave One Out" cross validation. Additionally, all relevant parameters in the `okriging` function are exposed. This function uses the doParallel package to distribute computation over multiple cores. If the phenotype is case/control, a ROCR AUC and GLM analysis is run and the results printed to screen.

Usage

```
krigr_cross_validation(cor.list, pheno.df, pheno.id = 1, h2.vec,
  covar.mat = NULL, nfold = 10, ncore = "all", verbose = FALSE, ...)
```

Arguments

`cor.list`	A list of correlation matrices used in Kriging. rownames and colnames of cor should be IID list and include idtest and idtrain.
`pheno.df`	A data frame with rownames set as sample IDs and a column containing phenotype data.
`pheno.id`	The name of the column in pheno which contains phenotype data to test.
`h2.vec`	has weights for each RM relatednes matrix
`covar.mat`	Data frame of covariates with rownames() set to sample IDs.
`nfold`	Select the number of cross validation rounds to run. The value "LOOCV" will run one round of cross validation for each sample in your dataset. The value "ncore" will set the test set size such that a single round runs on each core specified in the ncore option. Any numeric value will be set to the test size. Default runs 10 rounds of cross validation.
`ncore`	The number of cores available to distribute computaition across If a numeric value is supplied, that number of cores is registered. If the value "all" is supplied, all available cores are used.

`verbose`	Report rounds on cross validation on standard out.
`...`	Optional and unnamed arguments.

Value

A dataframe with three columns: sample ID, observed phenotype Ytest, and predicted phenotype Ypred

---

`load_sample_data` *Loads sample phenotype and covariate data into data frame.*

Description

This function loads a file into a data frame. This file should contain one row per sample in your study, and one column for each covariate and phenotype of interest. Additionally, it requires a header with "IID" for the column of sample IDs, and a unique name for each phenotype and covariate.

Usage

```
load_sample_data(phenoFile, main.pheno)
```

Arguments

`phenoFile`	File path to the phenotype/covariate file.
`main.pheno`	Column name of the main phenotype of interest.

Value

A data frame with dimensions (# of samples) x (# of phenotypes/covar)

`make_GXM` *Compute gene expression correlation matrix.*

Description

This function computes a gene expression correlation matrix given a file of transcript expression levels for each sample in the study. It returns a correlation matrix with rownames and colnames as sample IDs.

Usage

```
make_GXM(expFile = NULL, gxmFilePrefix = NULL, idfile = NULL)
```

Arguments

`expFile`	Path to gene expression file.
`gxmFilePrefix`	File path prefixes for outputting GCTA style binary correlation matrices.
`idfile`	Path to file containing family IDs and sample IDs with header FID and IID.

Value

Returns a correlation matrix of (N-samples x N-samples), with rownames and colnames as sample IDs.

Examples

```
## load gene expression values from vignette
 expressionFile <- system.file(package = "OmicKriging",
                     "doc/vignette_data/ig_gene_subset.txt.gz")
 ## compute correlation matrix
 geneCorrelationMatrix <- make_GXM(expressionFile)
```

`make_PCs_irlba` *Run Principal Component Analysis (PCA) using the irlba package.*

Description

A simple wrapper around the irlba() function which computes a partial SVD efficiently. This function's run time depends on the number of eigenvectors requested but scales well. Use this function to generate covariates for use with the `okriging` or `krigr_cross_validation` functions.

Usage

```
make_PCs_irlba(X, n.top = 2)
```

Arguments

`X`	A correlation matrix.
`n.top`	Number of top principal compenents to return

Value

A matrix of Principal Components of dimension (# of samples) x (n.top). As expected, eigenvectors are ordered by eigenvalue. Rownames are given as sample IDs.

References

library(irlba)

Examples

```
## compute PC's using the  gene expression correlation matrix from vignette
 ## load gene expression values from vignette
 expressionFile <- system.file(package = "OmicKriging",
                     "doc/vignette_data/ig_gene_subset.txt.gz")
 ## compute correlation matrix
 geneCorrelationMatrix <- make_GXM(expressionFile)
 ## find top ten PC's of this matrix using SVD
 topPcs <- make_PCs_irlba(geneCorrelationMatrix, n.top=10)
```

`make_PCs_svd` *Run Principal Component Analysis (PCA) using base R svd() function.*

Description

A simple wrapper around the base R svd() function which returns the top N eigenvectors of a matrix. Use this function to generate covariates for use with the `okriging` or `krigr_cross_validation` functions. This wrapper preserves the rownames of the original matrix.

Usage

```
make_PCs_svd(X, n.top = 2)
```

Arguments

`X`	A correlation matrix.
`n.top`	Number of top principal compenents to return

Value

A matrix of Principal Components of dimension (# of samples) x (n.top). As expected, eigenvectors are ordered by eigenvalue. Rownames are given as sample IDs.

Examples

```
## compute PC's using the  gene expression correlation matrix from vignette
 ## load gene expression values from vignette
 expressionFile <- system.file(package = "OmicKriging",
                     "doc/vignette_data/ig_gene_subset.txt.gz")
 ## compute correlation matrix
 geneCorrelationMatrix <- make_GXM(expressionFile)
 ## find top ten PC's of this matrix using SVD
 topPcs <- make_PCs_svd(geneCorrelationMatrix, n.top=10)
```

`okriging` *Run omic kriging on a set of correlation matrices and a given phenotype.*

Description

Universal kriging formula: lambda' = (c + X m)' iSig m' = (x - X' iSig c)' (X' iSig X)^-1 m' = (t(x) - c' iSig X) (X' iSig X)^-1 lambda' = (c' + m' X) iSig x: #covariates x ntest X: ntrain x #cov c: ntrain x ntest

Usage

```
okriging(idtest, idtrain = NULL, corlist, H2vec, pheno, phenoname,
  Xcova = NULL)
```

Arguments

`idtest`	A vector of sample IDs which constitute the test set.
`idtrain`	A vector of sample IDs which constitute the training set.
`corlist`	A list of correlation matrices used in Kriging. rownames and colnames of cor should be IID list and include idtest and idtrain.
`H2vec`	has weights for each RM relatednes matrix
`pheno`	A data frame with rownames set as sample IDs and a column containing phenotype data.
`phenoname`	The name of the column in pheno which contains phenotype data to test.
`Xcova`	Data frame of covariates with rownames() set to sample IDs.

Value

A dataframe with three columns: sample ID, observed phenotype Ytest, and predicted phenotype Ypred

References

Cressie 1993 Statistics for Spatial Data p.154

`read_GRMBin` *Read the GRM binary file.*

Description

Function provided by GCTA maintainers (modified slightly) for accessing their recently introduced binary GRM format. The GRM is stored as a vector of numerics which correspond to the lower triangular elements including the diagonal. We simply read these, pull the diagonal elements, and inflate them into a full symmetric matrix. We add sample IDs to colnames and rownames for compatibility with other Kriging functions.

Usage

```
read_GRMBin(prefix, size = 4)
```

Arguments

`prefix`	The file path prefix to GRM binary files (e.g., test.grm.bin, test.grm.N.bin, test.grm.id.)
`size`	The length (in bytes) of each value in the raw GRM vector. Default is 4, and matches GRM writen by GCTA 1.11.

Details

Note that the GRM is described by three files, and this function assumes that all have a common prefix that is passed in.

Value

GRM of dim (N.samples x N.samples) with rownames and colnames as sample ID.

References

http://www.complextraitgenomics.com/software/gcta/estimate_grm.html

Examples

```
## read binary Genetic Relatedness Matrix (GRM) generated by GCTA
  grmFile <- system.file(package = "OmicKriging",
                         "doc/vignette_data/ig_genotypes.grm.bin")
  grmFileBase <- substr(grmFile,1, nchar(grmFile) - 4)
  GRM <- read_GRMBin(grmFileBase)
```

`write_GRMBin` *Write GRM binary files.*

Description

Function to write a binary GRM format recently introduced by GCTA. It takes a correlation matrix as used by other Kriging functions, and writes three files: binary file for storing the diagonal + lower triangular elements, a text file for sample IDs, and a binary file storing the number of SNPs used in the correlation matrix calculation.

Usage

```
write_GRMBin(X, n.snps = 0, prefix, size = 4)
```

Arguments

`X`	Correlation matrix with rownames and colnames as sample IDs.
`n.snps`	Number of SNPs used in correlation matrix calculation. Default is 0.0.
`prefix`	Base file path and names for the three output files.
`size`	Number of bytes to write for each value. Default is 4

Value

None. Though side effects are writing three files as described above.

References

http://www.complextraitgenomics.com/software/gcta/estimate_grm.html

Examples

```
## create a random genotype matrix
  nSamples <- 10
  mMarkers <- 100
  X <- matrix(rbinom(n=100, size=2, prob=0.5), nrow=nSamples)
  ## compute the Genetric Relatedness Matrix
  grm <- cor(X)
  ## write a Genetic Relatedness Matrix (GRM)
  ## NOTE: to following is not run here -- not writing any files in examples
  #write_GRMBin(grm, n.snps=mMarkers, prefix="grm.out")
```

Index

∗ **GRM**
    make_PCs_irlba, 4
    make_PCs_svd, 5
∗ **PCA,**
    make_PCs_irlba, 4
    make_PCs_svd, 5
∗ **covariate,**
    make_PCs_irlba, 4
    make_PCs_svd, 5
∗ **cross**
    krigr_cross_validation, 2
∗ **input**
    load_sample_data, 3
∗ **prediction,**
    krigr_cross_validation, 2
∗ **prediction**
    okriging, 6
∗ **validation**
    krigr_cross_validation, 2

krigr_cross_validation, 2, *4*, *5*

load_sample_data, 3

make_GXM, 3
make_PCs_irlba, 4
make_PCs_svd, 5

okriging, *2*, *4*, *5*, 6

read_GRMBin, 7

write_GRMBin, 8