Package 'pv'

February 26, 2022

Type Package

Title Prism Vote: A stratified statistical framework for individual disease risk prediction

Version 0.1.6

Author Maggie Haitian Wang, Yexian Zhang

Maintainer Yexian Zhang <yxzhang@cuhkri.org.cn>

Description The Prism Vote framework is in essence a Bayesian hierarchical probability model, in which an individual's disease risk is integrated from subpopulation-specific disease estimates and the prior probabilities of a subject's subpopulation identity. Using PLINK, Principal Component Analysis (PCA) is adopted to identify subpopulations, Logistic Regression (LR) analysis is adopted for case-control Genome-Wide Association Studies (GWAS). Feature selection is performed using GWAS P-values with user-specified cutoff or using user-specified candidate SNPs. Four prediction models are built on selected SNPs. Two are LR and Polygenic Risk Score (PRS)-based LR. The other two are LR and PRS-based LR under the Prism Vote framework. Covariates can be included in GWAS analysis and predictive modeling.

License GPL-3

Encoding UTF-8

LazyData true

Imports plinkQC,
pROC,
infotheo,
purrr,
data.table

Depends R (>= 3.6.0)

SystemRequirements PLINK (1.9)

RoxygenNote 7.1.1

R topics documented:

`feature.selection` *Feature selection*

Description

Logistic regression-based feature selection approach.

Usage

```
feature.selection(
  input.dir,
  output.dir,
  genotype,
  phenotype,
  covar.number = NULL,
  plink.path = NULL,
  topK = 10,
  P.value = NULL,
  candidate.SNPs = NULL,
  verbose = TRUE
)
```

Arguments

`input.dir`	[character] The full absolute path to the directory containing the training and test dataset. If `input.dir` is missing, the current working directory obtained by `getwd()` is used.
`output.dir`	[character] The full absolute path where the result will be written to. If `output.dir` is missing, the current working directory obtained by `getwd()` is used.
`genotype`	[character] The prefix of PLINK binary files (bed/bim/fam).
`phenotype`	[character] A space- or tab-delimited file to specify an alternate phenotype for the logistic regression analysis using the "`--pheno`" flag in plink. This file must have a header row. The first and second columns of the phenotype file must be "FID" and "IID", the case/control phenotype in column 3 (1 = unaffected (control), 2 = affected (case)), and covariates in remaining columns. See the PLINK 1.9 documentation for details (https://www.cog-genomics.org/plink/1.9/).
`plink.path`	[character] The full absolute path to the PLINK executable file. The executable to run is path/to/plink.exe if you are on a Windows operating system, for Unix-like operating system this is path/to/plink. If `plink.path` is NULL, the PLINK PATH should be added as a system environment variable.
`topK`	[numeric] To specify the top K significant SNPs to build a prediction model. For a fair comparison, the number of the top-ranked SNPs from entire sample (for LR and PRS model) equals to the number of the unique union set of the selected SNPs from each stratum in PV. The default value is 10. This value is ignored when P.value or candidate.SNPs is not NULL.

P.value	[double] To specify the genome-wide significance P-value threshold to select the significant SNPs to build a prediction model. The default value is NULL. This value is ignored when candidate.SNPs is not NULL. When left NULL (the default), the topK or candidate.SNPs will be used. The P-value of each SNP is calculated from logistic regression analysis using PLINK 1.9 (via plink.lr).
candidate.SNPs	[vector] A character vector of SNP name, used to specify the candidate SNPs to build a prediction model, ignores P.value and topK. The default value is NULL. Should match the names of SNPs in the provided PLINK binary files.
verbose	[logical] If TRUE, the PLINK log, error, and warning information are printed to standard out. The default value is TRUE.

Value

feature.selection return a list containing the results of logistic regression analysis derived from PLINK (via plink.lr), the indices and names of selected features.

lr.result	The output of plink.lr
index	A vector of indices of the selected features.
name	A vector of names of the selected features.

See Also

plink.lr

Examples

```
input.dir <- system.file("extdata", package="pv")
output.dir <- system.file("extdata", package="pv")
path2plink <- '/path/to/plink'
## Not run:
feature.selection.result <- feature.selection(input.dir = input.dir,
output.dir = output.dir,
genotype = "train",
phenotype = "train.phenotypes.txt",
covar.number = c(2, 3),
plink.path = path2plink,
topK = 10,
verbose = TRUE)

## End(Not run)
```

LR.model *Logistic regression model*

Description

A Function to build a logistic regression prediction model on the training dataset, and make predictions on the test dataset.

Usage

```
LR.model(train.data, test.data)
```

Arguments

train.data	[data.frame] The training dataset. It must contain a column named Y providing the case/control phenotype (0 = unaffected (control), 1 = affected (case)).
test.data	[data.frame] The test dataset. It must contain a column named Y providing the case/control phenotype (0 = unaffected (control), 1 = affected (case)).

Value

LR.model return a vector containing the predicted probability.

Examples

```
input.dir <- system.file("extdata", package="PrismVote")
output.dir <- system.file("extdata", package="PrismVote")
path2plink <- '/path/to/plink'
## Not run:
covar.number <-  c(2, 3)

feature.selection.result <- feature.selection(input.dir = input.dir,
output.dir = output.dir,
genotype = "train",
phenotype = "train.phenotypes.txt",
covar.number = covar.number,
plink.path = path2plink,
topK = 10,
verbose = TRUE)

train.genotype.path <- file.path(input.dir, "train.raw")
train.genotype.path <- gsub('\\\\', '/', train.genotype.path)

train.phenotype.path <- file.path(input.dir, "train.phenotypes.txt")
train.phenotype.path <- gsub('\\\\', '/', train.phenotype.path)

train.data <- data.table::fread(train.genotype.path, data.table = FALSE)[, -c(1,2,3,4,5,6)]
colnames(train.data) <- unlist(purrr::map(colnames(train.data),function(x) {substr(x,1, nchar(x)-2)}))
train.pheno <- data.table::fread(train.phenotype.path, data.table = FALSE)
train.data$Y <- train.pheno[, 3]-1
train.data <- cbind(train.data[, feature.selection.result$index, drop = FALSE],
train.pheno[, covar.number + 2, drop = FALSE])

test.genotype.path <- file.path(input.dir, "test.raw")
test.genotype.path <- gsub('\\\\', '/', test.genotype.path)

test.phenotype.path <- file.path(input.dir, "test.phenotypes.txt")
test.phenotype.path <- gsub('\\\\', '/', test.phenotype.path)

test.data <- data.table::fread(test.genotype.path, data.table = FALSE)[, -c(1,2,3,4,5,6)]
colnames(test.data) <- unlist(purrr::map(colnames(test.data),function(x) {substr(x,1, nchar(x)-2)}))
test.pheno <- data.table::fread(test.phenotype.path, data.table = FALSE)
test.data$Y <- test.pheno[, 3]-1
test.data <- cbind(test.data[, feature.selection.result$index, drop = FALSE],
test.pheno[, covar.number + 2, drop = FALSE])

LR.pred <- LR.model(train.data, test.data)

## End(Not run)
```

LR.PV.model *Logistic regression-based Prism Vote model*

Description

A Function to build a logistic regression prediction model on the training dataset, and make predictions on the test dataset with the framework of Prism Vote.

Usage

```
LR.PV.model(train.data.list, test.data.list, test.prob.to.stratum)
```

Arguments

train.data.list	[list] A list containing the training data [data.frame] of each stratum. The stratum data with rows corresponding to individuals and columns to features, and must contain a column named Y providing the case/control phenotype (0 = unaffected (control), 1 = affected (case)).
test.data.list	[list] A list containing the test data [data.frame]. It must contain a column named Y providing the case/control phenotype (0 = unaffected (control), 1 = affected (case)).
test.prob.to.stratum	[list] Output of stratification. A list providing the probability that the test samples belong to each stratum.

Value

LR.PV.model return a list containing the predicted probability from each stratum and the aggregated prediction.

pred.stratum	A list of the predicted probability from each stratum.
pred.agg	A vector of the aggregated prediction.

Examples

```
input.dir <- system.file("data", package="pv")
output.dir <- system.file("data", package="pv")
path2plink <- '/path/to/plink'
## Not run:
stratum.count <- 2
covar.number <-  c(2, 3)

stratification.result <- stratification(input.dir = input.dir,
output.dir = input.dir,
train.genotype = "train",
test.genotype = "test",
stratum.count = stratum.count,
PCA.separate = FALSE,
PCs.count = 10,
plink.path = path2plink,
verbose = TRUE)
```

```
feature.selection.result <- list()
for (i in 1:stratum.count) {
  feature.selection.result[[i]] <- feature.selection(input.dir = input.dir,
  output.dir = output.dir,
  genotype = paste0("train.stratum.",i),
  phenotype = paste0("train.stratum.",i,".phenotype.txt"),
  covar.number = covar.number,
  plink.path = path2plink,
  topK = 10,
  verbose = TRUE)
}

train.genotype.path <- file.path(input.dir, "train.raw")
train.genotype.path <- gsub('\\\\', '/', train.genotype.path)
train.data <- data.table::fread(train.genotype.path, data.table = FALSE)[, -c(1,2,3,4,5,6)]
colnames(train.data) <- unlist(purrr::map(colnames(train.data),function(x) {substr(x,1, nchar(x)-2)}))
train.pheno <- data.table::fread(train.phenotype.path, data.table = FALSE)

test.genotype.path <- file.path(input.dir, "test.raw")
test.genotype.path <- gsub('\\\\', '/', test.genotype.path)
test.data <- data.table::fread(test.genotype.path, data.table = FALSE)[, -c(1,2,3,4,5,6)]
colnames(test.data) <- unlist(purrr::map(colnames(test.data),function(x) {substr(x,1, nchar(x)-2)}))
test.pheno <- data.table::fread(test.phenotype.path, data.table = FALSE)

train.data.list <- list()
test.data.list <- list()
for (i in 1:stratum.count) {
  train.data.stratum <- train.data[stratification.result$train.stratum.index[[i]],
  feature.selection.result[[i]]$index, drop = FALSE]
  train.data.stratum$Y <- train.pheno[stratification.result$train.stratum.index[[i]], 3]-1

  test.data.stratum <- test.data[, feature.selection.result[[i]]$index, drop = FALSE]
  test.data.stratum$Y <- test.pheno[, 3]-1
  if(!is.null(covar.number){
    train.data.stratum <- cbind(train.data.stratum,
   train.pheno[stratification.result$train.stratum.index[[i]], covar.number + 2, drop = FALSE])
   test.data.stratum <- cbind(test.data.stratum, test.pheno[, covar.number + 2, drop = FALSE])
  }

  train.data.list[[i]] <- train.data.stratum
  test.data.list[[i]] <- test.data.stratum
}

LR.PV.pred <- LR.PV.model(train.data.list, test.data.list, stratification.result$test.prob.to.stratum)

## End(Not run)
```

PCA *Principal component analysis*

Description

A Function to run principal component analysis on input dataset using PLINK 1.9.

Usage

```
PCA(
  input.dir,
  output.dir,
  train.genotype,
  test.genotype,
  PCA.separate = FALSE,
  PCs.count = 10,
  plink.path = NULL,
  verbose = TRUE
)
```

Arguments

input.dir	[character] The full absolute path to the directory containing the training and test dataset. If input.dir is missing, the current working directory obtained by getwd() is used.
output.dir	[character] The full absolute path where the result will be written to. If output.dir is missing, the current working directory obtained by getwd() is used.
train.genotype	[character] The prefix of PLINK binary files (bed/bim/fam) of the training dataset.
test.genotype	[character] The prefix of PLINK binary files (bed/bim/fam) of the test dataset.
PCA.separate	[logical] If TURE, the principal components are calculated from the training dataset and then project the test dataset onto those principal components. If FALSE, the principal components are calculated from the combined data of the training and test dataset. The default value is FALSE.
PCs.count	[numeric] To specify the number of top principal components that should be extracted. The default value is 10.
plink.path	[character] The full absolute path to the PLINK executable file. The executable to run is path/to/plink.exe if you are on Windows operating system, for Unix-like operating system this is path/to/plink. If plink.path is NULL, the PLINK PATH should be added as a system environment variable.
verbose	[logical] If TRUE, the PLINK log, error, and warning information are printed to standard out. The default value is TRUE.

Value

PCA returns a list containing eigenvalues and eigenvectors of the training and test dataset:

eigenvalue	A vector containing the top eigenvalues according to PCs.count specified.
train.eigenvector	A data frame containing the eigenvectors of the training dataset.
test.eigenvector	A data frame containing the eigenvectors of the test dataset.

Examples

```
input.dir <- system.file("extdata", package="pv")
output.dir <- system.file("extdata", package="pv")
path2plink <- '/path/to/plink'
## Not run:
pca.result <- PCA(input.dir = input.dir,
```

```
output.dir = output.dir,
train.genotype = "train",
test.genotype = "test",
PCA.separate = FALSE,
PCs.count = 10,
plink.path = path2plink,
verbose = TRUE)

## End(Not run)
```

`plink.lr` *Logistic regression analysis*

Description

A Function to perform logistic regression analysis given a case/control phenotype using PLINK 1.9.

Usage

```
plink.lr(
  input.dir,
  output.dir,
  genotype,
  phenotype = NULL,
  covar.number = NULL,
  plink.path = NULL,
  verbose = TRUE
)
```

Arguments

`input.dir`	[character] The full absolute path to the directory containing the training and test dataset. If `input.dir` is missing, the current working directory obtained by `getwd()` is used.
`output.dir`	[character] The full absolute path where the result will be written to. If `output.dir` is missing, the current working directory obtained by `getwd()` is used.
`genotype`	[character] The prefix of PLINK binary files (bed/bim/fam).
`phenotype`	[character] To specify an alternate phenotype for the logistic regression analysis using the "`--pheno`" flag in plink. This file must have a header row. The first and second columns of the phenotype file must be "FID" and "IID", the case/control phenotype in column 3 (1 = unaffected (control), 2 = affected (case)), and covariates in remaining columns. If `phenotype` is NULL, the original fam file must contain a phenotype in column 6. See the PLINK 1.9 documentation for details (https://www.cog-genomics.org/plink/1.9/).
`covar.number`	[vector] To specify a subset of column numbers of covariates to load from `phenotype` file using the "`--covar-number`" flag in plink. If NULL (the default), the logistic regression model without covariate adjustment.
`plink.path`	[character] The full absolute path to the PLINK executable file. The executable to run is path/to/plink.exe if you are on a Windows operating system, for Unix-like operating system this is path/to/plink. If `plink.path` is NULL, the PLINK PATH should be added as a system environment variable.

verbose [logical] If TRUE, the PLINK log, error, and warning information are printed to standard out. The default value is TRUE.

Value

plink.lr returns a data frame with the results of logistic regression analysis derived from PLINK.

Examples

```
input.dir <- system.file("extdata", package="pv")
output.dir <- system.file("extdata", package="pv")
path2plink <- '/path/to/plink'
## Not run:
lr.result <- plink.lr(input.dir = input.dir,
output.dir = output.dir,
genotype = "train",
phenotype = "train.phenotypes.txt",
covar.number = c(2, 3),
plink.path = path2plink,
verbose = TRUE)

## End(Not run)
```

PRS *Polygenic risk score*

Description

A Function to calculate polygenic risk score (PRS). PRS is calculated by summing risk alleles, which are weighted by the effect size of the risk alleles (i.e. beta coefficient for continuous traits or log(OR) for binary traits).

Usage

```
PRS(genotype, beta)
```

Arguments

genotype [data.frame/matrix] The genotype data is a data frame or matrix with rows corresponding to individuals and columns to SNPs. SNP genotypes are encoded as the number of minor allele (0, 1, 2).

beta [vector] The effect size of each SNP. If beta is missing, assuming that all SNPs have the same effect size.

Value

PRS return a vector of polygenic risk scores.

Examples

```
input.dir <- system.file("extdata", package="pv")
output.dir <- system.file("extdata", package="pv")
path2plink <- '/path/to/plink'
## Not run:
lr.result <- plink.lr(input.dir = input.dir,
output.dir = output.dir,
genotype = "train",
phenotype = "train.phenotypes.txt",
covar.number = c(2, 3),
plink.path = path2plink,
verbose = TRUE)

beta <- lr.result[which(lr.result$TEST == "ADD"), ]$BETA
beta[is.na(beta)] <- 0
genotype.path <- file.path(input.dir, "train.raw")
genotype.path <- gsub('\\\\', '/', genotype.path)
genotype <- data.table::fread(genotype.path, data.table = FALSE)[, -c(1,2,3,4,5,6)]
prs <- PRS(genotype, beta)

## End(Not run)
```

PV *Prism Vote*

Description

Perform Prism Vote method to build a prediction model on the training dataset, and make predictions on the test dataset.

Usage

```
PV(
  input.dir,
  output.dir,
  train.genotype,
  train.phenotype,
  test.genotype,
  test.phenotype,
  covar.number.PV = NULL,
  covar.number.LR = NULL,
  PCA.separate = FALSE,
  PCs.count = 10,
  stratum.count = 2,
  plink.path = NULL,
  P.value = NULL,
  topK = 10,
  candidate.SNPs = NULL,
  CS = FALSE,
  verbose = TRUE
)
```

Arguments

input.dir	[character] The full absolute path to the directory containing the training and test dataset. If input.dir is missing, the current working directory obtained by getwd() is used.
output.dir	[character] The full absolute path where the result will be written to. If output.dir is missing, the current working directory obtained by getwd() is used.
train.genotype	[character] The prefix of PLINK binary files (bed/bim/fam) of the training dataset.
train.phenotype	[character] A space- or tab-delimited file to specify an alternate phenotype of the training dataset for the logistic regression analysis using the "--pheno" flag in plink. This file must have a header row. The first and second columns of the phenotype file must be "FID" and "IID", the case/control phenotype in column 3 (1 = unaffected (control), 2 = affected (case)), and covariates in remaining columns. See the PLINK 1.9 documentation for details (https://www.cog-genomics.org/plink/1.9/).
test.genotype	[character] The prefix of PLINK binary files (bed/bim/fam) of the test dataset.
test.phenotype	[character] A space- or tab-delimited file to specify an alternate phenotype of the training dataset for the logistic regression analysis using the "--pheno" flag in plink. This file must have a header row. The first and second columns of the phenotype file must be "FID" and "IID", the case/control phenotype in column 3 (1 = unaffected (control), 2 = affected (case)), and covariates in remaining columns. See the PLINK 1.9 documentation for details (https://www.cog-genomics.org/plink/1.9/).
covar.number.PV	[vector] Used in PV model to specify a subset of column numbers of covariates to load from phenotype file using the "--covar-number" flag in plink (via plink.lr). If NULL (the default), the logistic regression model without covariate adjustment.
covar.number.LR	[vector] Used in LR and PRS model to specify a subset of column numbers of covariates to load from phenotype file using the "--covar-number" flag in plink (via plink.lr). If NULL (the default), the logistic regression model without covariate adjustment.
PCA.separate	[logical] If TURE, the principal components are calculated from the training dataset and then project the test dataset onto those principal components. If FALSE, the principal components are calculated from the combined data of the training and test dataset. The default value is FALSE.
PCs.count	[numeric] To specify the number of top principal components that should be extracted. The default value is 10.
stratum.count	[numeric] To specify the number of strata, as default the sample size of each stratum is N/stratum.count, N is the sample size. The default value is 2.
plink.path	[character] The full absolute path to the PLINK executable file. The executable to run is path/to/plink.exe if you are on a Windows operating system, for Unix-like operating system this is path/to/plink. If plink.path is NULL, the PLINK PATH should be added as a system environment variable.
P.value	[double] To specify the genome-wide significance P-value threshold to select the significant SNPs to build a prediction model. The default value is NULL. This value is ignored when candidate.SNPs is not NULL. When left NULL (the default), the topK or candidate.SNPs will be used. The P-value of each SNP is calculated from logistic regression analysis using PLINK 1.9 (via plink.lr).

topK [numeric] To specify the top K significant SNPs to build a prediction model. For a fair comparison, the number of the top-ranked SNPs from entire sample (for LR and PRS model) equals to the number of the unique union set of the selected SNPs from each stratum in PV. The default value is 10. This value is ignored when P.value or candidate.SNPs is not NULL.

candidate.SNPs [vector] A character vector of SNP name, used to specify the candidate SNPs to build a prediction model, ignores P.value and topK. The default value is NULL. Should match the names of SNPs in the provided PLINK binary files.

CS [logical] If TRUE, the softmax of cosine similarity will be used to calculate the probability that the samples belong to each stratum. If FALSE, the squared distance of a subject to a cluster center empirically follows a chi-squared distribution will be used. The default value is FALSE.

verbose [logical] If TRUE, the PLINK log, error, and warning information are printed to standard out. The default value is TRUE.

Value

PV returns a list with the following components:

stratification.result
The output of stratification

feature.selection.result
The output of feature.selection

LR.result The results of logistic regression model. A list with i) predict, the output of LR.model. ii) performance, a list containing the AUC and accuracy (Acc) value.

LR.PV.result The results of the logistic regression model under the Prism Vote framework. A list with i) predict, the output of LR.PV.model. ii) performance, a list containing the AUC and accuracy (Acc) value.

PRS.result The results of the logistic regression model based on the polygenic risk score. A list with i).predict, the output of LR.model. ii) performance, a list containing the AUC and accuracy (Acc) value.

PRS.PV.result The results of the logistic regression model based on the polygenic risk score under the Prism Vote framework. A list with i) predict, the output of LR.PV.model. ii) performance, a list containing the AUC and accuracy (Acc) value.

See Also

PCA, stratification, LR.model, PV.model, PRS, feature.selection

Examples

```
input.dir <- system.file("extdata", package="pv")
output.dir <- system.file("extdata", package="pv")
path2plink <- '/path/to/plink'
## Not run:
pv.result <- PV(input.dir = input.dir,
output.dir = output.dir,
train.genotype = "train",
train.phenotype = "train.phenotypes.txt",
test.genotype = "test",
test.phenotype = "test.phenotypes.txt",
covar.number.PV = c(2,3),
```

```
covar.number.LR = c(2,3),
PCA.separate = FALSE,
PCs.count = 10,
stratum.count = 2,
plink.path = path2plink,
P.value = NULL,
topK = 10,
CS = FALSE,
candidate.SNPs = NULL,
verbose = TRUE)

## End(Not run)
```

`stratification` *Data stratification*

Description

A Function to stratify samples to several strata using top principal components.

Usage

```
stratification(
  input.dir,
  output.dir,
  train.genotype,
  test.genotype,
  stratum.count = 2,
  PCA.separate = FALSE,
  PCs.count = 10,
  plink.path = NULL,
  CS = FALSE,
  verbose = TRUE
)
```

Arguments

`input.dir`	[character] The full absolute path to the directory containing the training and test dataset. If `input.dir` is missing, the current working directory obtained by `getwd()` is used.
`output.dir`	[character] The full absolute path where the result will be written to. If `output.dir` is missing, the current working directory obtained by `getwd()` is used.
`train.genotype`	[character] The prefix of PLINK binary files (bed/bim/fam) of the training dataset.
`test.genotype`	[character] The prefix of PLINK binary files (bed/bim/fam) of the test dataset.
`stratum.count`	[numeric] To specify the number of strata, as default the sample size of each stratum is N/stratum.count, N is the sample size.
`PCA.separate`	[logical] If TURE, the principal components are calculated from the training dataset and then project the test dataset onto those principal components. If FALSE, the principal components are calculated from the combined data of the training and test dataset. The default value is FALSE.

PCs.count	[numeric] To specify the number of top principal components that should be extracted. The default value is 10.
plink.path	[character] The full absolute path to the PLINK executable file. The executable to run is path/to/plink.exe if you are on a Windows operating system, for Unix-like operating system this is path/to/plink. If plink.path is NULL, the PLINK PATH should be added as a system environment variable.
CS	[logical] If TRUE, the softmax of cosine similarity will be used to calculate the probability that the samples belong to each stratum. If FALSE, the squared distance of a subject to a cluster center empirically follows a chi-squared distribution will be used. The default value is FALSE.
verbose	[logical] If TRUE, the PLINK log, error, and warning information are printed to standard out. The default value is TRUE.

Value

stratification returns a list containing the following components:

train.stratum	A vector containing the stratum number each training sample belongs to.
train.stratum.index	A list containing the index of training samples belonging to each stratum.
stratum.center	A vector containing the center of each stratum.
train.distance.to.center	A list containing the distance between the training samples and the center of each stratum.
test.distance.to.center	A list containing the distance between the training samples and the center of each stratum.
train.prob.to.g	A list containing the probability of the training samples having variable g under the hypothesis that the sample belongs to each stratum.
test.prob.to.g	A list containing the probability of the test samples having variable g under the hypothesis that the sample belongs to each stratum.
train.prob.to.stratum	A list containing the probability that the training samples belong to each stratum.
test.prob.to.stratum	A list containing the probability that the test samples belong to each stratum.

See Also

PCA

Examples

```
input.dir <- system.file("extdata", package="pv")
output.dir <- system.file("extdata", package="pv")
path2plink <- '/path/to/plink'
## Not run:
stratification.result <- stratification(input.dir = input.dir,
output.dir = input.dir,
train.genotype = "train",
test.genotype = "test",
stratum.count = 2,
```

```
    PCA.separate = FALSE,
    PCs.count = 10,
    plink.path = path2plink,
    CS = FALSE,
    verbose = TRUE)

    ## End(Not run)
```

Index