Package 'PredictABEL'

February 19, 2015

Title Assessment of Risk Prediction Models

Version 1.2-2

Date 2014-12-20

Author Suman Kundu, Yurii S. Aulchenko, A. Cecile J.W. Janssens

Maintainer Suman Kundu <suman_math@yahoo.com>

Depends R (>= 2.12.0), Hmisc, ROCR, epitools, PBSmodelling

Suggests GenABEL

Description PredictABEL includes functions to assess the performance of risk models. The package contains functions for the various measures that are used in empirical studies, including univariate and multivariate odds ratios (OR) of the predictors, the c-statistic (or area under the receiver operating characteristic (ROC) curve (AUC)), Hosmer-Lemeshow goodness of fit test, reclassification table, net reclassification improvement (NRI) and integrated discrimination improvement (IDI). Also included are functions to create plots, such as risk distributions, ROC curves, calibration plot, discrimination box plot and predictiveness curves. In addition to functions to assess the performance of risk models, the package includes functions to obtain weighted and unweighted risk scores as well as predicted risks using logistic regression analysis. These logistic regression functions are specifically written for models that include genetic variables, but they can also be applied to models that are based on non-genetic risk factors only. Finally, the package includes function to construct a simulated dataset with genotypes, genetic risks, and disease status for a hypothetical population, which is used for the evaluation of genetic risk models.

License GPL (>= 2)

URL http://www.genabel.org/packages/PredictABEL

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-21 17:55:30

R topics documented:

`PredictABEL-package` *An R package for the analysis of (genetic) risk prediction studies.*

Description

An R package for the analysis of (genetic) risk prediction studies.

Details

Fueled by the substantial gene discoveries from genome-wide association studies, there is increasing interest in investigating the predictive ability of genetic risk models. To assess the performance of genetic risk models, PredictABEL includes functions for the various measures and plots that have been used in empirical studies, including univariate and multivariate odds ratios (ORs) of the predictors, the c-statistic (or AUC), Hosmer-Lemeshow goodness of fit test, reclassification table, net reclassification improvement (NRI) and integrated discrimination improvement (IDI). The plots included are the ROC plot, calibration plot, discrimination box plot, predictiveness curve, and several risk distributions.

These functions can be applied to predicted risks that are obtained using logistic regression analysis, to weighted or unweighted risk scores, for which the functions are included in this package. The functions can also be used to assess risks or risk scores that are constructed using other methods, e.g., Cox Proportional Hazards regression analysis, which are not included in the current version. Risks obtained from other methods can be imported into R for assessment of the predictive performance.

The functions to construct the risk models using logistic regression analyses are specifically written for models that include genetic variables, eventually in addition to non-genetic factors, but they can

also be applied to construct models that are based on non-genetic risk factors only.

Before using the functions `fitLogRegModel` for constructing a risk model or `riskScore` for computing risk scores, the following checks on the dataset are advisable to be done:

(1) Missing values: The logistic regression analyses and computation of the risk score are done only for subjects that have no missing data. In case of missing values, individuals with missing data can be removed from the dataset or imputation strategies can be used to fill in missing data. Subjects with missing data can be removed with the R function `na.omit` (available in `stats` package). Example: `DataFileNew <- na.omit(DataFile)` will make a new dataset (`DataFileNew`) with no missing values;

(2) Multicollinearity: When there is strong correlation between the predictor variables, regression coefficients may be estimated imprecisely and risks scores may be biased because the assumption of independent effects is violated. In genetic risk prediction studies, problems with multicollinearity should be expected when single nucleotide polymorphisms (SNPs) located in the same gene are in strong linkage disequilibrium (LD). For SNPs in LD it is common to select the variant with the lowest p-value in the model;

(3) Outliers: When the data contain significant outliers, either clinical variables with extreme values of the outcomes or extreme values resulting from errors in the data entry, these may impact the construction of the risk models and computation of the risks scores. Data should be carefully checked and outliers need to be removed or replaced, if justified;

(4) Recoding of data: In the computation of unweighted risk scores, it is assumed that the genetic variants are coded `0,1,2` representing the number of alleles carried. When variants are coded `0,1` representing a dominant or recessive effect of the alleles, the variables need to be recoded before unweighted risk scores can be computed.

To import data into R several alternative strategies can be used. Use the `Hmisc` package for importing SPSS and SAS data into R. Use "`ExampleData <- read.table("DataName.txt", header=T, sep="\t")`" for text files where variable names are included as column headers and data are separated by tabs. Use "`ExampleData <- read.table("Name.csv", sep=",", header=T)`" for comma-separated files with variable names as column headers. Use "`setwd(dir)`" to set the working directory to "dir". The datafile needs to be present in the working directory.

To export datafiles from R tables to a tab-delimited textfile with the first row as the name of the variables, use "`write.table(R_Table, file="Name.txt", row.names=FALSE, sep="\t")`" and when a comma-separated textfile is requested and variable names are provided in the first row, use "`write.table(R_Table, file="Name.csv", row.names=FALSE, sep=",")`". When the directory is not specified, the file will be saved in the working directory. For exporting R data into SPSS, SAS and Stata data, use functions in the the `foreign` package.

Several functions in this package depend on other R packages:

(1) `Hmisc`, is used to compute NRI and IDI;

(2) `ROCR`, is used to produce ROC plots;

(3) `epitools`, is used to compute univariate odds ratios;

(4) `PBSmodelling`, is used to produce predictiveness curve.

Acknowledgements

The authors would like to acknowledge Lennart Karssen, Maksim Struchalin and Linda Broer from the Department of Epidemiology, Erasmus Medical Center, Rotterdam for their valuable comments and suggestions to make this package.

Note

The current version of the package includes the basic measures and plots that are used in the assessment of (genetic) risk prediction models and the function to construct a simulated dataset that contains individual genotype data, estimated genetic risk and disease status, used for the evaluation of genetic risk models (see Janssens et al, Genet Med 2006). Planned extensions of the package include functions to construct risk models using Cox Proportional Hazards analysis for prospective data and assess the performance of risk models for time-to-event data.

Author(s)

Suman Kundu

Yurii S. Aulchenko

A. Cecile J.W. Janssens

References

S Kundu, YS Aulchenko, CM van Duijn, ACJW Janssens. PredictABEL: an R package for the assessment of risk prediction models. Eur J Epidemiol. 2011;26:261-4.

ACJW Janssens, JPA Ioannidis, CM van Duijn, J Little, MJ Khoury. Strengthening the Reporting of Genetic Risk Prediction Studies: The GRIPS Statement Proposal. Eur J Epidemiol. 2011;26:255-9.

ACJW Janssens, JPA Ioannidis, S Bedrosian, P Boffetta, SM Dolan, N Dowling, I Fortier, AN. Freedman, JM Grimshaw, J Gulcher, M Gwinn, MA Hlatky, H Janes, P Kraft, S Melillo, CJ O'Donnell, MJ Pencina, D Ransohoff, SD Schully, D Seminara, DM Winn, CF Wright, CM van Duijn, J Little, MJ Khoury. Strengthening the reporting of genetic risk prediction studies (GRIPS)-Elaboration and explanation. Eur J Epidemiol. 2011;26:313-37.

Aulchenko YS, Ripke S, Isaacs A, van Duijn CM. GenABEL: an R package for genome-wide association analysis. Bioinformatics 2007;23(10):1294-6.

ExampleData *A hypothetical dataset that is used to demonstrate all functions.*

Description

ExampleData is a hypothetical dataset constructed to demonstrate all functions in the package. ExampleData is a data frame containing a binary outcome variable (e.g., disease present/absent) and genetic and non-genetic predictor variables for 10,000 persons. In this dataset, column 1 is the ID number, column 2 is the outcome variable, columns 3-10 are non-genetic variables and columns 11-16 are genetic variables.

Usage

```
data(ExampleData)
```

Examples

```
data(ExampleData)
# show first 5 records (rows) of the dataset
head(ExampleData,5)
```

ExampleModels *An example code to construct a risk model using logistic regression analysis.*

Description

ExampleModels constructs two risk models using logistic regression analysis. Most of the functions in this package require a logistic regression model as an input and estimate predicted risks from this fitted model. To illustrate these functions without repeating the construction of a logistic regression model, this example code has been created. The function returns two different risk models, riskModel1 which is based on non-genetic predictors and riskModel2 which includes genetic and non-genetic predictors.

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2
# specify column numbers of non-genetic predictors
cNonGenPred1 <- c(3:10)
cNonGenPred2 <- c(3:10)
# specify column numbers of non-genetic predictors that are categorical
cNonGenPredCat1 <- c(6:8)
cNonGenPredCat2 <- c(6:8)
# specify column numbers of genetic predictors
cGenPred1 <- c(0)
cGenPred2 <- c(11:16)
# specify column numbers of genetic predictors that are categorical
cGenPredsCat1 <- c(0)
cGenPredsCat2 <- c(0)

# fit logistic regression models
riskmodel1 <- fitLogRegModel(data=ExampleData, cOutcome=cOutcome,
cNonGenPreds=cNonGenPred1, cNonGenPredsCat=cNonGenPredCat1,
cGenPreds=cGenPred1, cGenPredsCat=cGenPredsCat1)
riskmodel2 <- fitLogRegModel(data=ExampleData, cOutcome=cOutcome,
cNonGenPreds=cNonGenPred2, cNonGenPredsCat=cNonGenPredCat2,
cGenPreds=cGenPred2, cGenPredsCat=cGenPredsCat2)
```

```
# combine output in a list
ExampleModels <- list(riskModel1=riskmodel1, riskModel2=riskmodel2)
```

`fitLogRegModel` *Function to fit a logistic regression model.*

Description

The function fits a standard GLM function for the logistic regression model.

Usage

```
fitLogRegModel(data, cOutcome, cNonGenPreds, cNonGenPredsCat,
cGenPreds, cGenPredsCat)
```

Arguments

`data`	Data frame or matrix that includes the outcome and predictor variables.
`cOutcome`	Column number of the outcome variable. `cOutcome=2` means that the second column of the dataset is the outcome variable. To fit the logistic regression model, the outcome variable needs to be (re)coded as `1` for the presence and `0` for the absence of the outcome of interest.
`cNonGenPreds`	Column numbers of the non-genetic predictors that are included in the model. An example to denote column numbers is `c(3,6:8,10)`. Choose `c(0)` when no non-genetic predictors are considered.
`cNonGenPredsCat`	Column numbers of the non-genetic predictors that are entered as categorical variables in the model. When non-genetic predictors are not specified as being categorical they are treated as continuous variables in the model. If no non-genetic predictors are categorical, denote `c(0)`.
`cGenPreds`	Column numbers of the genetic predictors or genetic risk score. Denote `c(0)` when the prediction model does not consider genetic predictors or genetic risk score.
`cGenPredsCat`	Column numbers of the genetic predictors that are entered as categorical variables in the model. When SNPs are considered as categorical, the model will estimate effects per genotype. Otherwise, SNPs are considered as continuous variables for which the model will estimate an allelic effect. Choose c(0) when no genetic predictors are considered as categorical or when genetic predictors are entered as a risk score into the model.

Details

The function fits a standard GLM function for the logistic regression model. This function can be used to construct a logistic regression model based on genetic and non-genetic predictors. The function also allows to enter the genetic predictors as a single risk score. For that purpose, the function requires that the dataset additionally includes the risk score. A new dataset can be constructed using "`NewExampleData <- cbind(ExampleData,riskScore)`". The genetic risk scores can be obtained using the function `riskScore` in this package or be imported from other methods.

Value

No value returned.

See Also

`predRisk`, `ORmultivariate`, `riskScore`

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of outcome variable
cOutcome <- 2
# specify column numbers of non-genetic predictors
cNonGenPred <- c(3:10)
# specify column numbers of non-genetic predictors that are categorical
cNonGenPredCat <- c(6:8)
# specify column numbers of genetic predictors
cGenPred <- c(11,13:16)
# specify column numbers of genetic predictors that are categorical
cGenPredCat <- c(0)

# fit logistic regression model
riskmodel <- fitLogRegModel(data=ExampleData, cOutcome=cOutcome,
cNonGenPreds=cNonGenPred, cNonGenPredsCat=cNonGenPredCat,
cGenPreds=cGenPred, cGenPredsCat=cGenPredCat)

# show summary details for the fitted risk model
summary(riskmodel)
```

`ORmultivariate` *Function to obtain multivariate odds ratios from a logistic regression model.*

Description

The function estimates multivariate (adjusted) odds ratios (ORs) with 95% confidence intervals (CIs) for all the genetic and non-genetic variables in the risk model.

Usage

```
ORmultivariate(riskModel, filename)
```

Arguments

`riskModel`	Name of logistic regression model that can be fitted using the function `fitLogRegModel`.
`filename`	Name of the output file in which the multivariate ORs will be saved. If no directory is specified, the file is saved in the working directory as a txt file. When `filename` is not specified, the output is not saved.

Details

The function requires that first a logistic regression model is fitted either by using `GLM` function or the function `fitLogRegModel`. In addition to the multivariate ORs, the function returns summary statistics of model performance, namely the Brier score and the Nagelkerke's R^2 value. The Brier score quantifies the accuracy of risk predictions by comparing predicted risks with observed outcomes at individual level (where outcome values are either 0 or 1). The Nagelkerke's R^2 value indicates the percentage of variation of the outcome explained by the predictors in the model.

Value

The function returns:

`Predictors Summary`	OR with 95% CI and corresponding p-values for each predictor in the model
`Brier Score`	Brier score
`Nagelkerke Index`	Nagelkerke's R^2 value

References

Brier GW. Verification of forecasts expressed in terms of probability. Monthly weather review 1950;78:1-3.

Nagelkerke NJ. A note on a general definition of the coefficient of determination. Biometrika 1991;78:691-692.

See Also

`fitLogRegModel`, `ORunivariate`

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of outcome variable
cOutcome <- 2
# specify column numbers of non-genetic predictors
cNonGenPred <- c(3:10)
# specify column numbers of non-genetic predictors that are categorical
cNonGenPredCat <- c(6:8)
```

```
# specify column numbers of genetic predictors
cGenPred <- c(11,13:16)
# specify column numbers of genetic predictors that are categorical
cGenPredCat <- c(0)

# fit logistic regression model
riskmodel <- fitLogRegModel(data=ExampleData, cOutcome=cOutcome,
cNonGenPreds=cNonGenPred, cNonGenPredsCat=cNonGenPredCat,
cGenPreds=cGenPred, cGenPredsCat=cGenPredCat)

# obtain multivariate OR(95% CI) for all predictors of the fitted model
ORmultivariate(riskModel=riskmodel, filename="multiOR.txt")
```

`ORunivariate` *Function to compute univariate ORs for genetic predictors.*

Description

The function computes the univariate ORs with 95% CIs for genetic predictors.

Usage

```
ORunivariate(data, cOutcome, cGenPreds, filenameGeno, filenameAllele)
```

Arguments

`data`	Data frame or matrix that includes the outcome and predictors variables.
`cOutcome`	Column number of the outcome variable. `cOutcome=2` means that the second column of the dataset is the outcome variable.
`cGenPreds`	Column numbers of genetic variables for which the ORs are calculated.
`filenameGeno`	Name of the output file in which the univariate ORs and frequencies per genotype will be saved. The file is saved in the working directory as a txt file. When no `filenameGeno` is specified, the output is not saved.
`filenameAllele`	Name of the output file in which the univariate ORs and frequencies per allele will be saved. The file is saved in the working directory as a txt file. When no `filenameAllele` is specified, the output is not saved.

Details

The function computes the univariate ORs with 95% CIs for the specified genetic variants both per allele and per genotype. The ORs are saved with the data from which they are calculated. Genotype frequencies are provided for persons with and without the outcome of interest. The genotype or allele that is coded as `'0'` is considered as the reference to computes the ORs.

Value

The function returns two different tables. One table contains genotype frequencies and univariate ORs with 95% CIs and the other contains allele frequencies and univariate ORs with 95% CIs.

See Also

ORmultivariate

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2
# specify column numbers of genetic predictors
cGenPreds <- c(11:13,16)

# compute univariate ORs
ORunivariate(data=ExampleData, cOutcome=cOutcome, cGenPreds=cGenPreds,
filenameGeno="GenoOR.txt", filenameAllele="AlleleOR.txt")
```

---

plotCalibration *Function for calibration plot and Hosmer-Lemeshow goodness of fit test.*

Description

The function produces a calibration plot and provides Hosmer-Lemeshow goodness of fit test statistics.

Usage

```
plotCalibration(data, cOutcome, predRisk, groups, rangeaxis,
plottitle, xlabel, ylabel, filename, fileplot, plottype)
```

Arguments

data	Data frame or numeric matrix that includes the outcome and predictor variables.
cOutcome	Column number of the outcome variable.
predRisk	Vector of predicted risks of all individuals in the dataset.
groups	Number of groups considered in Hosmer-Lemeshow test. Specification of groups is optional (default groups is 10).
rangeaxis	Range of x-axis and y-axis. Specification of rangeaxis is optional. Default is c(0,1).
plottitle	Title of the plot. Specification of plottitle is optional. Default is "Calibration plot".
xlabel	Label of x-axis Default. Specification of xlabel is optional. Default is "Predicted risk".
ylabel	Label of y-axis. Specification of ylabel is optional. Default is "Observed risk".

filename	Name of the output file in which the calibration table is saved. The file is saved as a txt file in the working directory. When no filename is specified, the output is not saved. Example: filename="calibration.txt"
fileplot	Name of the file that contains the calibation plot. The file is saved in the working directory in the format specified under plottype. Example: fileplot="plotname". Note that the extension is not specified here. When fileplot is not specified, the plot is not saved.
plottype	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. Foe example, plottype="eps" will save the plot in eps format. When plottype is not specified, the plot will be saved in jpg format.

Details

Hosmer-Lemeshow test statistic is a measure of the fit of the model, comparing observed and predicted risks across subgroups of the population. The default number of groups is 10.

The function requires the outcome of interest and predicted risks of all individuals. Predicted risks can be obtained from the functions fitLogRegModel and predRisk or be imported from other packages or methods.

Value

The function creates a calibration plot and returns the following measures:

Chi_square	Chi square value of Hosmer-Lemeshow test
df	Degrees of freedom, which is (groups-2) where groups: number of groups
p_value	p-value of Hosmer-Lemeshow test for goodness of fit

References

Hosmer DW, Hosmer T, Le Cessie S, Lemeshow S. A comparison of goodness-of-fit tests for the logistic regression model. Stat Med 1997; 16:965-980.

See Also

predRisk

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2

# fit a logistic regression model
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel <- ExampleModels()$riskModel2
```

```
# obtain predicted risks
predRisk <- predRisk(riskmodel)

# specify range of x-axis and y-axis
rangeaxis <- c(0,1)
# specify number of groups for Hosmer-Lemeshow test
groups <- 10

# compute calibration measures and produce calibration plot
plotCalibration(data=ExampleData, cOutcome=cOutcome, predRisk=predRisk,
groups=groups, rangeaxis=rangeaxis)
```

`plotDiscriminationBox` *Function for box plots of predicted risks separately for individuals with and without the outcome of interest.*

Description

The function produces box plots of predicted risks for individuals with and without the outcome of interest and calculates the discrimination slope.

Usage

```
plotDiscriminationBox(data, cOutcome, predrisk, labels, plottitle,
ylabel, fileplot, plottype)
```

Arguments

`data`	Data frame or matrix that includes the outcome and predictors variables.
`cOutcome`	Column number of the outcome variable.
`predrisk`	Vector of predicted risks.
`labels`	Labels given to the groups of individuals without and with the outcome of interest. Specification of `label` is optional. Default is `c("Without disease", "With disease")`.
`plottitle`	Title of the plot. Specification of `plottitle` is optional. Default is "Box plot".
`ylabel`	Label of y-axis. Specification of `ylabel` is optional. Default is "Predicted risks".
`fileplot`	Name of the file that contains the plot. The file is saved in the working directory in the format specified under `plottype`. Example: `fileplot="name"`. Note that the extension is not specified here. When `fileplot` is not specified, the plot is not saved.
`plottype`	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. Foe example, `plottype="eps"` will save the plot in eps format. When `plottype` is not specified, the plot will be saved in jpg format.

Details

The discrimination slope is the difference between the mean predicted risks of individuals with and without the outcome of interest. Predicted risks can be obtained using the `fitLogRegModel` and `predRisk` or be imported from other programs. The difference between discrimination slopes of two separate risk models is equivalent to (`IDI`) which is discussed in details in the `reclassification` function.

Value

The function creates a box plots of predicted risks for individuals with and without the outcome of interest and returns the discrimination slope.

References

Yates JF. External correspondence: decomposition of the mean probability score. Organizational Behavior and Human Performance 1982;30:132-156.

See Also

`reclassification`, `predRisk`

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of outcome variable
cOutcome <- 2

# fit a logistic regression model
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel <- ExampleModels()$riskModel2

# obtain predicted risks
predRisk <- predRisk(riskmodel)
# specify labels for the groups without and with the outcome of interest
labels <- c("Without disease", "With disease")

# produce discrimination box plot
plotDiscriminationBox(data=ExampleData, cOutcome=cOutcome, predrisk=predRisk,
labels=labels)
```

`plotPredictivenessCurve`

Function for predictiveness curve.

Description

The function creates a plot of cumulative percentage of individuals to the predicted risks.

Usage

```
plotPredictivenessCurve(predrisk, rangeyaxis, labels, plottitle,
xlabel, ylabel, fileplot, plottype)
```

Arguments

predrisk	Vector of predicted risk. When multiple curves need to be presented in one plot, specify multiple vectors of predicted risks as predrisk=cbind(predrisk1, predrisk2,...,predriskn
rangeyaxis	Range of the y axis. Default rangeyaxis is c(0,1).
labels	Label(s) given to the predictiveness curve(s). Specification of labels is optional. When specified, the labels should be in the same order as specified in predrisk.
plottitle	Title of the plot. Specification of plottitle is optional. Default is "Predictiveness curve".
xlabel	Label of x-axis. Specification of xlabel is optional. Default is "Cumulative percentage".
ylabel	Label of y-axis. Specification of ylabel is optional. Default is "Predicted risks".
fileplot	Name of the output file that contains the plot. The file is saved in the working directory in the format specified under plottype. Example: fileplot="plotname". Note that the extension is not specified here. When fileplot is not specified, the plot is not saved.
plottype	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. For example, plottype="eps" will save the plot in eps format. When plottype is not specified, the plot will be saved in jpg format.

Details

The Predictiveness curve is a plot of cumulative percentage of individuals to the predicted risks. Cumulative percentage indicates the percentage of individual that has a predicted risk equal or lower than the risk value. Predicted risks can be obtained using the functions fitLogRegModel and predRisk or be imported from other methods or packages.

Value

The function creates a predictiveness curve.

References

Pepe MS, Feng Z, Huang Y, et al. Integrating the predictiveness of a marker with its performance as a classifier. Am J Epidemiol 2008;167:362-368.

See Also

predRisk

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)

# fit logistic regression models
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel1 <- ExampleModels()$riskModel1
riskmodel2 <- ExampleModels()$riskModel2

# obtain predicted risks
predRisk1 <- predRisk(riskmodel1)
predRisk2 <- predRisk(riskmodel2)

# specify range of y-axis
rangeyaxis <- c(0,1)
# specify labels of the predictiveness curves
labels <- c("without genetic factors", "with genetic factors")

# produce predictiveness curves
plotPredictivenessCurve(predrisk=cbind(predRisk1,predRisk2),
rangeyaxis=rangeyaxis, labels=labels)
```

---

plotPriorPosteriorRisk

Function to plot posterior risks against prior risks.

Description

Function to plot posterior risks against prior risks.

Usage

```
plotPriorPosteriorRisk(data, priorrisk, posteriorrisk, cOutcome, plottitle,
xlabel, ylabel, rangeaxis, plotAll=TRUE, labels, filename, fileplot, plottype)
```

Arguments

data	Data frame or matrix that includes the outcome and predictors variables.
priorrisk	Vector of predicted risks based on initial model.
posteriorrisk	Vector of predicted risks based on updated model.
cOutcome	Column number of the outcome variable.
plottitle	Title of the plot. Specification of plottitle is optional. Default is "PriorPosteriorRisk plot".
xlabel	Label of x-axis. Specification of xlabel is optional. Default is "Prior risk".
ylabel	Label of y-axis. Specification of ylabel is optional. Default is "Posterior risk".

rangeaxis	Range of x-axis and y-axis. Specification of rangeaxis is optional. Default is c(0,1).
plotAll	plotAll=TRUE will create one plot for the total population. When plotAll=FALSE separate plots will be created for individuals with and without the outcome of interest. means two separate plots for with and without outcome of interest.
labels	Labels given to the groups of individuals without and with the outcome of interest. Default labels is c("without outcome", "with outcome"). Note that when plotAll=TRUE, specification of labels is not necessary.
filename	Name of the output file in which prior and posterior risks for each individual with the outcome will be saved. If no directory is specified, the file is saved in the working directory as a txt file. When no filename is specified, the output is not saved.
fileplot	Name of the output file that contains the plot. The file is saved in the working directory in the format specified under plottype. Example: fileplot="plotname". Note that the extension is not specified here. When fileplot is not specified, the plot is not saved.
plottype	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. For example, plottype="eps" will save the plot in eps format. When plottype is not specified, the plot will be saved in jpg format.

Details

The function creates a plot of posterior risks (predicted risks using the updated model) against prior risks (predicted risks using the initial model). Predicted risks can be obtained using the functions fitLogRegModel and predRisk or be imported from other packages or methods.

Value

The function creates a plot of posterior risks against prior risks.

See Also

predRisk

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of outcome variable
cOutcome <- 2

# fit logistic regression models
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel1 <- ExampleModels()$riskModel1
riskmodel2 <- ExampleModels()$riskModel2

# obtain predicted risks
```

```
predRisk1 <- predRisk(riskmodel1)
predRisk2 <- predRisk(riskmodel2)

# specify label of x-axis
xlabel <- "Prior risk"
# specify label of y-axis
ylabel <- "Posterior risk"
# specify title for the plot
titleplot <- "Prior versus posterior risk"
# specify range of the x-axis and y-axis
rangeaxis <- c(0,1)
# labels given to the groups without and with the outcome of interest
labels<- c("without outcome", "with outcome")

# produce prior risks and posterior risks plot
plotPriorPosteriorRisk(data=ExampleData, priorrisk=predRisk1,
posteriorrisk=predRisk2, cOutcome=cOutcome, xlabel=xlabel, ylabel=ylabel,
rangeaxis=rangeaxis, plotAll=TRUE, plottitle=titleplot, labels=labels)
```

`plotRiskDistribution` *Function to plot histogram of risks separated for individuals with and without the outcome of interest.*

Description

Function to plot histogram of risks separated for individuals with and without the outcome of interest.

Usage

```
plotRiskDistribution(data, cOutcome, risks, interval, rangexaxis,
rangeyaxis, plottitle, xlabel, ylabel, labels, fileplot, plottype)
```

Arguments

`data`	Data frame or numeric matrix that includes the outcome and predictor variables.
`cOutcome`	Column number of the outcome variable.
`risks`	Risk of each individual. It is specified by either a vector of risk scores or a vector of predicted risks.
`interval`	Size of the risk intervals. For example, `interval=.1` will construct the following intervals for predicted risks: `0-0.1, 0.1-0.2,..., 0.9-1`.
`rangexaxis`	Range of the x-axis. Specification of `rangexaxis` is optional.
`rangeyaxis`	Range of the y-axis.
`plottitle`	Title of the plot. Specification of `plottitle` is optional. Default is "Histogram of risks".
`xlabel`	Label of x-axis. Specification of `xlabel` is optional. Default is "Risk score".

ylabel	Label of y-axis. Specification of ylabel is optional. Default is "Percentage".
labels	Labels given to the groups of individuals without and with the outcome of interest. Specification of labels is optional. Default is c("Without outcome", "With outcome").
fileplot	Name of the output file that contains the plot. The file is saved in the working directory in the format specified under plottype. Example: fileplot="plotname". Note that the extension is not specified here. When fileplot is not specified, the plot is not saved.
plottype	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. For example, plottype="eps" will save the plot in eps format. When plottype is not specified, the plot will be saved in jpg format.

Value

The function creates the histogram of risks separated for individuals with and without the outcome of interest.

See Also

plotROC, riskScore

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2

# fit a logistic regression model
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel <- ExampleModels()$riskModel2

# obtain predicted risks
predRisk <- predRisk(riskmodel)

# specify the size of each interval
interval <- .05
# specify label of x-axis
xlabel <- "Predicted risk"
# specify label of y-axis
ylabel <- "Percentage"
# specify range of x-axis
xrange <- c(0,1)
# specify range of y-axis
yrange <- c(0,40)
# specify title for the plot
maintitle <- "Distribution of predicted risks"
# specify labels
labels <- c("Without outcome", "With outcome")
```

```
# produce risk distribution plot
plotRiskDistribution(data=ExampleData, cOutcome=cOutcome,
risks=predRisk, interval=interval, plottitle=maintitle, rangexaxis=xrange,
rangeyaxis=yrange, xlabel=xlabel, ylabel=ylabel, labels=labels)
```

`plotRiskscorePredrisk` *Function to plot predicted risks against risk scores.*

Description

This function is used to make a plot of predicted risks against risk scores.

Usage

```
plotRiskscorePredrisk(data, riskScore, predRisk, plottitle, xlabel,
ylabel, rangexaxis, rangeyaxis, filename, fileplot, plottype)
```

Arguments

`data`	Data frame or matrix that includes the outcome and predictors variables.
`riskScore`	Vector of (weighted or unweighted) genetic risk scores.
`predRisk`	Vector of predicted risks.
`plottitle`	Title of the plot. Specification of `plottitle` is optional. Default is "Risk score predicted risk plot".
`xlabel`	Label of x-axis. Specification of `xlabel` is optional. Default is "Risk score".
`ylabel`	Label of y-axis. Specification of `ylabel` is optional. Default is "Predicted risk".
`rangexaxis`	Range of the x axis. Specification of `rangexaxis` is optional.
`rangeyaxis`	Range of the y axis. Specification of `rangeyaxis` is optional. Default is `c(0,1)`.
`filename`	Name of the output file in which risk scores and predicted risks for each individual will be saved. If no directory is specified, the file is saved in the working directory as a txt file. When no `filename` is specified, the output is not saved.
`fileplot`	Name of the output file that contains the plot. The file is saved in the working directory in the format specified under `plottype`. Example: `fileplot="plotname"`. Note that the extension is not specified here. When `fileplot` is not specified, the plot is not saved.
`plottype`	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. For example, `plottype="eps"` will save the plot in eps format. When `plottype` is not specified, the plot will be saved in jpg format.

Details

The function creates a plot of predicted risks against risk scores. Predicted risks can be obtained using the functions `fitLogRegModel` and `predRisk` or be imported from other methods or packages. The function `riskScore` can be used to compute unweighted or weighted risk scores.

Value

The function creates a plot of predicted risks against risk scores.

See Also

riskScore, predRisk

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)

# fit a logistic regression  model
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel <- ExampleModels()$riskModel2

# obtain predicted risks
predRisk <- predRisk(riskmodel)

# specify column numbers of genetic predictors
cGenPred <- c(11:16)

# function to compute unweighted genetic risk scores
riskScore <- riskScore(weights=riskmodel, data=ExampleData,
cGenPreds=cGenPred, Type="unweighted")

# specify range of x-axis
rangexaxis <- c(0,12)
# specify range of y-axis
rangeyaxis <- c(0,1)
# specify label of x-axis
xlabel <- "Risk score"
# specify label of y-axis
ylabel <- "Predicted risk"
# specify title for the plot
plottitle <- "Risk score versus predicted risk"

# produce risk score-predicted risk plot
plotRiskscorePredrisk(data=ExampleData, riskScore=riskScore, predRisk=predRisk,
plottitle=plottitle, xlabel=xlabel, ylabel=ylabel, rangexaxis=rangexaxis,
rangeyaxis=rangeyaxis, filename="RiskscorePredRisk.txt")
```

---

plotROC *Function for a receiver operating characteristic curve (ROC) plot and area under the ROC curve (AUC) value.*

Description

The function produces ROC curve and corresponding AUC value with 95% CI. The function can plot one or multiple ROC curves in a single plot.

Usage

```
plotROC(data, cOutcome, predrisk, labels, plottitle, xlabel, ylabel,
fileplot, plottype)
```

Arguments

`data`	Data frame or matrix that includes the outcome and predictors variables.
`cOutcome`	Column number of the outcome variable.
`predrisk`	Vector of predicted risk. When multiple curves need to be presented in one plot, specify multiple vectors of predicted risks as `predrisk=cbind(predrisk1, predrisk2,...,predriskn`
`labels`	Label(s) given to the ROC curve(s). Specification of `labels` is optional. When specified, the `labels` should be in the same order as specified in `predrisk`.
`plottitle`	Title of the plot. Specification of `plottitle` is optional. Default is "ROC plot".
`xlabel`	Label of x-axis. Specification of `xlabel` is optional. Default is "1- Specificity".
`ylabel`	Label of y-axis. Specification of `ylabel` is optional. Default is "Sensitivity".
`fileplot`	Name of the output file that contains the plot. The file is saved in the working directory in the format specified under `plottype`. Example: `fileplot="plotname"`. Note that the extension is not specified here. When `fileplot` is not specified, the plot is not saved.
`plottype`	The format in which the plot is saved. Available formats are wmf, emf, png, jpg, jpeg, bmp, tif, tiff, ps, eps or pdf. For example, `plottype="eps"` will save the plot in eps format. When `plottype` is not specified, the plot will be saved in jpg format.

Details

The function requirs predicted risks or risk scores and the outcome of interest for all individuals. Predicted risks can be obtained using the functions `fitLogRegModel` and `predRisk` or be imported from other methods or packages.

Value

The function creates ROC plot and returns AUC value with 95% CI.

References

Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 1982;143:29-36.

Tobias Sing, Oliver Sander, Niko Beerenwinkel, Thomas Lengauer. ROCR: visualizing classifier performance in R. Bioinformatics 2005;21(20):3940-3941.

See Also

predRisk, plotRiskDistribution

Examples

```
# specify the arguments in the function to produce ROC plot
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2

# fit logistic regression models
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel1 <- ExampleModels()$riskModel1
riskmodel2 <- ExampleModels()$riskModel2

# obtain predicted risks
predRisk1 <- predRisk(riskmodel1)
predRisk2 <- predRisk(riskmodel2)

# specify label of the ROC curve
labels <- c("without genetic factors", "with genetic factors")

# produce ROC curve
plotROC(data=ExampleData, cOutcome=cOutcome,
predrisk=cbind(predRisk1,predRisk2), labels=labels)
```

---

predRisk *Function to compute predicted risks for all individuals in the dataset.*

Description

Function to compute predicted risks for all individuals in the (new)dataset.

Usage

```
predRisk(riskModel, data, cID, filename)
```

Arguments

riskModel	Name of logistic regression model that can be fitted using the function fitLogRegModel.
data	Data frame or matrix that includes the ID number and predictor variables.
cID	Column number of ID variable. The ID number and predicted risks will be saved under filename. When cID is not specified, the output is not saved.
filename	Name of the output file in which the ID number and estimated predicted risks will be saved. The file is saved in the working directory as a txt file. Example: filename="name.txt". When no filename is specified, the output is not saved.

Details

The function computes predicted risks from a specified logistic regression model. The function `fitLogRegModel` can be used to construct such a model.

Value

The function returns a vector of predicted risks.

See Also

`fitLogRegModel`, `plotCalibration`, `plotROC`, `plotPriorPosteriorRisk`

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2
# specify column number of ID variable
cID <- 1
# specify column numbers of non-genetic predictors
cNonGenPred <- c(3:10)
# specify column numbers of non-genetic predictors that are categorical
cNonGenPredCat <- c(6:8)
# specify column numbers of genetic predictors
cGenPred <- c(11,13:16)
# specify column numbers of genetic predictors that are categorical
cGenPredCat <- c(0)

# fit logistic regression model
riskmodel <- fitLogRegModel(data=ExampleData, cOutcome=cOutcome,
cNonGenPreds=cNonGenPred, cNonGenPredsCat=cNonGenPredCat,
cGenPreds=cGenPred, cGenPredsCat=cGenPredCat)

# obtain predicted risks
predRisk <- predRisk(riskModel=riskmodel)
```

---

reclassification *Function for reclassification table and statistics.*

Description

The function creates a reclassification table and provides statistics.

Usage

```
reclassification(data, cOutcome, predrisk1, predrisk2, cutoff)
```

Arguments

data	Data frame or matrix that includes the outcome and predictors variables.
cOutcome	Column number of the outcome variable.
predrisk1	Vector of predicted risks of all individuals using initial model.
predrisk2	Vector of predicted risks of all individuals using updated model.
cutoff	Cutoff values for risk categories. Define the cut-off values as c(0,...,1). Multiple values can be defined and always specify 0 and 1. Example: c(0,.20,.30,1)

Details

The function creates a reclassification table and computes the categorical and continuous net reclassification improvement (NRI) and integrated discrimination improvement (IDI). A reclassification table indicates the number of individuals who move to another risk category or remain in the same risk category as a result of updating the risk model. Categorical NRI equal to x% means that compared with individuals without outcome, individuals with outcome were almost x% more likely to move up a category than down. The function also computes continuous NRI, which does not require any discrete risk categories and relies on the proportions of individuals with outcome correctly assigned a higher probability and individuals without outcome correctly assigned a lower probability by an updated model compared with the initial model. IDI equal to x% means that the difference in average predicted risks between the individuals with and without the outcome increased by x% in the updated model. The function requires predicted risks estimated by using two separate risk models. Predicted risks can be obtained using the functions fitLogRegModel and predRisk or be imported from other methods or packages.

Value

The function returns the reclassification table, separately for individuals with and without the outcome of interest and the following measures:

NRI (Categorical)	Categorical Net Reclassification Improvement with 95% CI and p-value of the test
NRI (Continuous)	Continuous Net Reclassification Improvement with 95% CI and p-value of the test
IDI	Integrated Discrimination Improvement with 95% CI and p-value of the test

References

Cook NR. Use and misuse of the receiver operating characteristic curve in risk prediction. Circulation 2007;115(7):928-935.

Pencina MJ, D'Agostino RB Sr, D'Agostino RB Jr, Vasan RS. Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond. Stat Med 2008;27(2):157-172; discussion 207-212.

See Also

plotDiscriminationBox, predRisk

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column number of the outcome variable
cOutcome <- 2

# fit logistic regression models
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel1 <- ExampleModels()$riskModel1
riskmodel2 <- ExampleModels()$riskModel2

# obtain predicted risks
predRisk1 <- predRisk(riskmodel1)
predRisk2 <- predRisk(riskmodel2)
# specify cutoff values for risk categories
cutoff <- c(0,.10,.30,1)

# compute reclassification measures
reclassification(data=ExampleData, cOutcome=cOutcome,
predrisk1=predRisk1, predrisk2=predRisk2, cutoff)
```

---

riskScore *Function to compute genetic risk scores.*

Description

The function computes unweighted or weighted genetic risk scores. The relative effects (or weights) of genetic variants can either come from beta coefficients of a risk model or from a vector of beta coefficients imported into R, e.g., when beta cofficients are obtained from meta-analysis.

Usage

```
riskScore(weights, data, cGenPreds, Type)
```

Arguments

weights	The vector that includes the weights given to the genetic variants. See details for more informations.
data	Data frame or matrix that includes the outcome and predictors variables.
cGenPreds	Column numbers of the genetic variables on the basis of which the risk score is computed.
Type	Specification of the type of risk scores that will be computed. Type can be weighted (Type="weighted") or unweighted (Type="unweighted").

Details

The function calculates unweighted or weighted genetic risk scores. The unweighted genetic risk score is a simple risk allele count assuming that all alleles have the same effect. For this calculation, it is required that the genetic variables are coded as the number of risk alleles. Beta coefficients are used to determine which allele is the risk allele. When the sign of the beta coefficient is negative, the allele coding is reversed. The weighted risk score is a sum of the number of risk alleles multiplied by their beta coefficients.

The beta coefficients can come from two different sources, either beta coefficients of a risk model or a vector of beta coefficients imported into R, e.g., when beta cofficients are obtained from meta-analysis. This vector of beta coefficients should be a named vector containing the same names as mentioned in genetic variants. A logistic regression model can be constructed using `fitLogRegModel` from this package.

Value

The function returns a vector of risk scores.

Note

When a vector of beta coefficients is imported, it should be checked whether the DNA strands and the coding of the risk alleles are the same as in the study data. The functions are available in the package GenABEL to accurately compute risk scores when the DNA strands are different or the risk alleles are coded differently in the study data and the data used in meta-analysis.

See Also

`plotRiskDistribution`, `plotRiskscorePredrisk`

Examples

```
# specify dataset with outcome and predictor variables
data(ExampleData)
# specify column numbers of genetic predictors
cGenPred <- c(11:16)

# fit a logistic regression model
# all steps needed to construct a logistic regression model are written in a function
# called 'ExampleModels', which is described on page 4-5
riskmodel <- ExampleModels()$riskModel2

# compute unweighted risk scores
riskScore <- riskScore(weights=riskmodel, data=ExampleData,
cGenPreds=cGenPred, Type="unweighted")
```

`simulatedDataset` *Function to construct a simulated dataset containing individual genotype data, genetic risks and disease status for a hypothetical population.*

Description

Construct a dataset that contains individual genotype data, genetic risk, and disease status for a hypothetical population. The dataset is constructed using simulation in such a way that the frequencies and odds ratios (ORs) of the genetic variants and the population disease risk computed from this dataset are the same as specified by the input parameters.

Usage

```
simulatedDataset(ORfreq, poprisk, popsize, filename)
```

Arguments

`ORfreq`	Matrix with ORs and frequencies of the genetic variants. The matrix contains four columns in which the first two describe ORs and the last two describe the corresponding frequencies. The number of rows in this matrix is same as the number of genetic variants included. Genetic variants can be specified as per genotype, per allele, or as dominant/ recessive effect of the risk allele. When per genotype data are used, OR of the heterozygous and homozygous risk genotypes are mentioned in the first two columns and the corresponding genotype frequencies are mentioned in the last two columns. When per allele data are used, the OR and frequency of the risk allele are specified in the first and third column and the remaining two cells are coded as '1'. Similarly, when dominant/ recessive effects of the risk alleles are used, the OR and frequency of the dominant/ recessive variant are specified in the first and third column, and the remaining two cells are coded as '0'.
`poprisk`	Population disease risk (expressed in proportion).
`popsize`	Total number of individuals included in the dataset.
`filename`	Name of the file in which the dataset will be saved. The file is saved in the working directory as a txt file. When no filename is specified, the output is not saved.

Details

The function will execute when the matrix with odds ratios and frequencies, population disease risk and the number of individuals are specified.

The simulation method is described in detail in the references.

The method assumes that (i) the combined effect of the genetic variants on disease risk follows a multiplicative (log additive) risk model; (ii) genetic variants inherit independently, that is no linkage

disequilibrium between the variants; (iii) genetic variants have independent effects on the disease risk, which indicates no interaction among variants; and (iv) all genotypes and allele proportions are in Hardy-Weinberg equilibrium. Assumption (ii) and (iv) are used to generate the genotype data, and assumption (ii) and (iii) are used to calculate disease risk.

Simulating the dataset involves three steps: (1) modelling genotype data, (2) modelling disease risks, and (3) modelling disease status. Brief descriptions of these steps are as follows:

(1) Modelling genotype data: For each variant the genotype frequencies are either specified or calculated from the allele frequencies using Hardy-Weinberg equilibrium. Then, the genotypes for each genetic variant are randomly distributed without replacement over all individuals.

(2) Modelling disease risks: For the calculation of the individual disease risk, Bayes' theorem is used, which states that the posterior odds of disease are obtained by multiplying the prior odds by the likelihood ratio (LR) of the individual genotype data. The prior odds are calculated from the population disease risk or disease prevalence (prior odds= prior risk/ (1- prior risk)) and the posterior odds are converted back into disease risk (disease risk= posterior odds/ (1+ posterior odds)). Under the no linkage disequilibrium (LD) assumption, the LR of a genetic profile is obtained by multiplying the LRs of the single genotypes that are included in the risk model. The LR of a single genotype is calculated using frequencies and ORs of genetic variants and population disease risk. See references for more details.

(3) Modelling disease status: To model disease status, we used a procedure that compares the estimated disease risk of each subject to a randomly drawn value between 0 and 1 from a uniform distribution. A subject is assigned to the group who will develop the disease when the disease risk is higher than the random value and to the group who will not develop the disease when the risk is lower than the random value.

This procedure ensures that for each genomic profile, the percentage of people who will develop the disease equals the population disease risk associated with that profile, when the subgroup of individuals with that profile is sufficiently large.

Value

The function returns:

`Dataset` A data frame or matrix that includes genotype data, genetic risk and disease status for a hypothetical population. The dataset contains (4 + number of genetic variants included) columns, in which the first column is the un-weighted risk score, which is the sum of the number of risk alleles for each individual, the third column is the estimated genetic risk, the forth column is the individual disease status expressed as '0' or '1', indicating without or with the outcome of interest, and the fifth until the end column are genotype data for the variants expressed as '0', '1' or '2', which indicate the number of risk alleles present in each individual for the genetic variants.

References

Janssens AC, Aulchenko YS, Elefante S, Borsboom GJ, Steyerberg EW, van Duijn CM. Predictive testing for complex diseases using multiple genes: fact or fiction? Genet Med. 2006;8:395-400.

Kundu S, Karssen LC, Janssens AC: Analytical and simulation methods for estimating the potential predictive ability of genetic profiling: a comparison of methods and results. Eur J Hum Genet. 2012 May 30.

van Zitteren M, van der Net JB, Kundu S, Freedman AN, van Duijn CM, Janssens AC. Genome-based prediction of breast cancer risk in the general population: a modeling study based on meta-analyses of genetic associations. Cancer Epidemiol Biomarkers Prev. 2011;20:9-22.

van der Net JB, Janssens AC, Sijbrands EJ, Steyerberg EW. Value of genetic profiling for the prediction of coronary heart disease. Am Heart J. 2009;158:105-10.

Janssens AC, Moonesinghe R, Yang Q, Steyerberg EW, van Duijn CM, Khoury MJ. The impact of genotype frequencies on the clinical validity of genomic profiling for predicting common chronic diseases. Genet Med. 2007;9:528-35.

Examples

```
# specify the matrix containing the ORs and frequencies of genetic variants
# In this example we used per allele effects of the risk variants
ORfreq<-cbind(c(1.35,1.20,1.24,1.16), rep(1,4), c(.41,.29,.28,.51),rep(1,4))

# specify the population disease risk
popRisk <- 0.3
# specify size of hypothetical population
popSize <- 10000

# Obtain the simulated dataset
Data <- simulatedDataset(ORfreq=ORfreq, poprisk=popRisk, popsize=popSize)

# Obtain the AUC and produce ROC curve
plotROC(data=Data, cOutcome=4, predrisk=Data[,3])
```

Index