Package 'iCARE'

November 12, 2026

Title Individualized Coherent Absolute Risk Estimation (iCARE)

Version 1.40.0

Date 2018-12-03

Author Parichoy Pal Choudhury, Paige Maas, William Wheeler, Nilanjan Chatterjee

Description An R package to build, validate and apply absolute risk models

Maintainer Parichoy Pal Choudhury <Parichoy.PalChoudhury@cancer.org>

Depends R (>= 3.3.0), plotrix, gtools, Hmisc

Suggests RUnit, BiocGenerics

License GPL-3 + file LICENSE

LazyData true

biocViews Software, StatisticalMethod, GenomeWideAssociation

NeedsCompilation yes

git_url https://git.bioconductor.org/packages/iCARE

git_branch RELEASE_3_23

git_last_commit 278a074

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

bc_data 2
computeAbsoluteRisk 4
computeAbsoluteRiskSplitInterval 7
iCARE 10
ModelValidation 11
plotModelValidation 15

Index **19**

bc_data *Data for examples*

Description

Example data for computeAbsoluteRisk, computeAbsoluteRiskSplitInterval, ModelValidation, and plotModelValidation.

Details

- bc_model_cov_info: a main list containing information on family history, age at menarche (years), parity, age at first birth (years), age at menopause (years), height (meters), Body Mass Index (kg/sq.m.), use of hormone replacement therapy, use of estrogen and progesterone combined therapy, use of estrogen only therapy, current use of hormone replacement therapy, alcohol (drinks/week), smoking status.; information on each risk factor is given as a list
- bc_model_formula: formula for the specification of the models with risk factors
- bc_72_snps: contains published SNP information from reference: Michailidou K, Lindstrom S, Dennis J, Beesley J, Hui S, Kar S, Lemacon A, Soucy P, Glubb D, Rostamianfar A, et al. (2017) Association analysis identifies 65 new breast cancer risk loci. Nature 551:92-94
- bc_model_log_or: vector of log-odds ratios of family history, age at menarche (years), parity, age at first birth (years), age at menopause (years), height (meters), Body Mass Index (kg/sq.m.), use of hormone replacement therapy, use of estrogen and progesterone combined therapy, use of estrogen only therapy, current use of hormone replacement therapy, alcohol (drinks/week), smoking status.
- bc_model_log_or_post_50: vector of log-odds ratios of family history, age at menarche (years), parity, age at first birth (years), age at menopause (years), height (meters), Body Mass Index (kg/sq.m.), use of hormone replacement therapy, use of estrogen and progesterone combined therapy, use of estrogen only therapy, current use of hormone replacement therapy, alcohol (drinks/week), smoking status. for women 50 years or older
- ref_cov_dat: contains individual level reference dataset of risk factors representative of the underlying population imputed using reference (4) and (5)
- ref_cov_dat_post_50: contains individual level reference dataset of the risk factors for women aged 50 years or older
- bc_inc: contains age-specific incidence rates of breast cancer from reference (3)
- mort_inc: contains age-specific incidence rates of all-cause mortality from reference (1) below
- new_cov_prof: Information on family history, age at menarche (years), parity, age at first birth (years), age at menopause (years), height (meters), Body Mass Index (kg/sq.m.), use of hormone replacement therapy, use of estrogen and progesterone combined therapy, use of estrogen only therapy, current use of hormone replacement therapy, alcohol (drinks/week), smoking status for three women (given for illustration of absolute risk prediction)
- new_snp_prof: Information on 72 breast cancer associated SNPs for three women (given for illustration of absolute risk prediction)

- validation.cohort.data: Simulated full cohort dataset of 50,000 women for illustration of model validation. The variables are:
  - id: Subject id
  - famhist: Family history; binary indicator of presence/absence of disease among first degree relatives
  - parity: number of child births categorized as nulliparous (ref), 1 births, 2 births, 3 births, 4+ births
  - menarche_dec: categories of age at menarche (years) with levels: less than 11,11-11.5,11.5-12,12-13(ref),13-14,14-15, greater than 15
  - birth_dec: categories of age at first birth (years) with levels: less than 19 (ref), 19-22,22-23,23-25,25-27,27-30,30-34,34-38, greater than 38
  - agemeno_dec: categories of age at menopause (years) with levels: less than 40 (ref), 40-45, 45-47, 47-48, 48-50, 50-51, 51-52, 52-53, 53-55, greater than 55
  - height_dec: categories of height (meters) with levels: less than 1.55, 1.55-1.57, 1.57-1.60, 1.60-1.61, 1.61-1.63, 1.63-1.65, 1.65-1.66, 1.66-1.68, 1.68-1.71
  - bmi_dec: categories of body mass index (kg/sq.m.) with levels: less than 21.5 (ref), 21.5-23, 23-24.2, 24.2-25.3, 25.3-26.5, 26.5-27.8, 27.8-29.3, 29.3-31.4, 31.4-34.6
  - rd_menohrt: use of hormone replacement therapy with levels: premenopausal (ref), postmenopausal and never HRT user, postmenopausal and ever HRT user
  - rd2_everhrt_c: binary indicator for postmenopausal and ever user of estrogen and progesterone combined therapy
  - rd2_everhrt_e: binary indicator for postmenopausal and ever user of estrogen only therapy
  - rd2_currhrt: binary indicator of postmenopausal and current HRT user
  - alcoholdweek_dec: alcohol in drinks per week categorized into levels: none (ref), 0-0.4, 0.4-0.8, 0.8-1.5, 1.5-3.2, 3.2-5.7, 5.7-9.8, >9.8
  - ever_smoke: binary indicator for ever smoker
  - study.entry.age: age of study entry
  - study.exit.age: age of study exit
  - observed.outcome: binary indicator of disease status (yes/no)
  - time.of.onset: time (in years) since study entry to the development of disease; for subjects who have not developed disease beyond the observed followup, it is set to Inf
  - observed.followup: number of years the subject is followed up in the study (difference between the age of study exit and age of study entry)
- validation.nested.case.control.data: A simulated example of a case-control study of 5285 women, nested within the full cohort. In addition to the variables given above, it has information on the 72 breast cancer associated SNPs with variable names being the rs-identifiers.
- output: object returned from computeAbsoluteRisk

References

(1) Centers for Disease Control and Prevention (CDC), National Center for Health Statistics (NCHS). Underlying Cause of Death 1999-2011 on CDC WONDER Online Database, released 2014. Data are from the Multiple Cause of Death Files, 1999-2011, as compiled from data provided by the 57

vital statistics jurisdictions through the Vital Statistics Cooperative Program.
Accessed at http://wonder.cdc.gov/ucd-icd10.html on Aug 26, 2014.

(2) Michailidou K, Beesley J, Lindstrom S, et al. Genome-wide association analysis of more than 120,000 individuals identifies 15 new susceptibility loci for breast cancer. Nature genetics 2015;47:373-80.

(3) Surveillance, Epidemiology, and End Results (SEER) Program SEER*Stat Database: Incidence - SEER 18 Regs Research Data, Nov 2011 Sub, Vintage 2009 Pops (2000-2009) <Katrina/Rita Population Adjustment> - Linked To County Attributes - Total U.S., 1969-2010 Counties. In: National Cancer Institute D, Surveillance Research Program, Surveillance Systems Branch, ed. SEER18 ed.

(4) 2010 National Health Interview Survey (NHIS) Public Use Data Release, NHIS Survey Description. 2011.
(Accessed at ftp://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NHIS/2010/srvydesc.pdf.)

(5) Centers for Disease Control and Prevention (CDC). National Center for Health Statistics (NCHS). National Health and Nutrition Examination Survey Questionnaire. Hyattsville, MD: U.S. Department of Health and Human Services, Centers for Disease Control and Prevention; 2010.

Examples

```
temp <- data(bc_data, package="iCARE")

# Display the object names
temp
```

`computeAbsoluteRisk` *Building and Applying an Absolute Risk Model*

Description

This function is used to build absolute risk models and apply them to estimate absolute risks.

Usage

```
computeAbsoluteRisk(model.formula = NULL, model.cov.info = NULL,
  model.snp.info = NULL, model.log.RR = NULL, model.ref.dataset = NULL,
  model.ref.dataset.weights = NULL, model.disease.incidence.rates,
  model.competing.incidence.rates = NULL, model.bin.fh.name = NA,
  n.imp = 5, apply.age.start, apply.age.interval.length,
  apply.cov.profile = NULL, apply.snp.profile = NULL, use.c.code = 1,
  return.lp = FALSE, return.refs.risk = FALSE)
```

Arguments

`model.formula` an object of class `formula`: a symbolic description of the model to be fitted, e.g. Y~Parity+FamilyHistory.

`model.cov.info` contains information about the risk factors in the model ; a main list containing a list for each covariate, which must have the fields:

- `"name"` : a string with the covariate name, matching name in model.formula
- `"type"` : a string that is either "continuous" or "factor".

If factor variable, then:

- `"levels"` : vector with strings of level names
- `"ref"` : optional field, string with name of referent level

`model.snp.info` dataframe with three columns, named: ["snp.name", "snp.odds.ratio", "snp.freq"]

`model.log.RR` vector with log odds ratios corresponding to the model params; no intercept; names must match design matrix arising from `model.formula` and `model.cov.info`; check names using function `check_design_matrix()`.

`model.ref.dataset` dataframe of risk factors for a sample of subjects representative of underlying population, no missing values. Variables must be in same order with same names as in `model.formula`.

`model.ref.dataset.weights` optional vector of sampling weights for `model.ref.dataset`.

`model.disease.incidence.rates` two column matrix [integer ages, incidence rates] or three column matrix [start age, end age, rate] with incidence rate of disease. Must fully cover age interval for estimation.

`model.competing.incidence.rates` two column matrix [integer ages, incidence rates] or three column matrix [start age, end age, rate] with incidence rate of competing events. Must fully cover age interval for estimation.

`model.bin.fh.name` string name of family history variable, if in model. This must refer to a variable that only takes values 0,1, NA.

`n.imp` integer value for number of imputations for handling missing SNPs.

`apply.age.start` single integer or vector of integer ages for the start of the interval over which to compute absolute risk.

`apply.age.interval.length` single integer or vector of integer years over which absolute risk should be computed.

`apply.cov.profile`
: dataframe containing the covariate profiles for which absolute risk will be computed. Covariates must be in same order with same names as in `model.formula`.

`apply.snp.profile`
: data frame with observed SNP data (coded 0,1, 2, or NA). May have missing values.

`use.c.code`
: binary indicator of whether to run the c program for fast computation.

`return.lp`
: binary indicator of whether to return the linear predictor for each subject in apply.cov.profile.

`return.refs.risk`
: binary indicator of whether to return the absolute risk prediction for each subject in `model.ref.dataset`.

Details

Individualized Coherent Absolute Risk Estimators (iCARE) is a tool that allows researchers to quickly build models for absolute risk and apply them to estimate individuals' risk based on a set of user defined input parameters. The software gives users the flexibility to change or update models rapidly based on new risk factors or tailor models to different populations based on the specification of simply three input arguments:

- (1) a model for relative risk assumed to be externally derived
- (2) an age-specific disease incidence rate and
- (3) the distribution of risk factors for the population of interest.

The tool can handle missing information on risk factors for risk estimation using an approach where all estimates are derived from a single model through appropriate model averaging.

Value

This function returns a list of results objects, including:

- `risk` : absolute risk estimates over the specified interval for subjects given by `apply.cov.profile`
- `details`: dataframe with the start of the interval, the end of the interval, the covariate profile, and the risk estimates for each individual
- `beta.used` : the log odds ratios used in the model
- `lps` : linear predictors for subjects in `model.cov.profile`, if requested by `return.lp`
- `refs.risk` : absolute risk estimates for subjects in `model.ref.dataset`, if requested by `return.refs.risk`; computes for first age interval provided

Examples

```
data(bc_data, package="iCARE")
results = computeAbsoluteRisk(model.formula=bc_model_formula,
                              model.cov.info    = bc_model_cov_info,
                              model.snp.info    = bc_72_snps,
                              model.log.RR      = bc_model_log_or,
                              model.ref.dataset = ref_cov_dat,
                              model.disease.incidence.rates   = bc_inc,
                              model.competing.incidence.rates = mort_inc,
                              model.bin.fh.name = "famhist",
                              apply.age.start    = 50,
                              apply.age.interval.length = 30,
                              apply.cov.profile  = new_cov_prof,
                              apply.snp.profile  = new_snp_prof,
                              return.refs.risk   = TRUE)
summary(results)
plot(results, main="Risk")
boxplot(results$risk ~ new_cov_prof$famhist, na.rm=TRUE)
```

`computeAbsoluteRiskSplitInterval`

Building and Applying an Absolute Risk Model: Compute Risk over Interval Split in Two Parts

---

Description

This function is used to build an absolute risk model that incorporates different input parameters before and after a given time point. The model is then applied to estimate absolute risks.

Usage

```
computeAbsoluteRiskSplitInterval(apply.age.start, apply.age.interval.length,
      apply.cov.profile, model.formula, model.disease.incidence.rates,
      model.log.RR, model.ref.dataset, model.ref.dataset.weights=NULL,
      model.cov.info, use.c.code=1, model.competing.incidence.rates=NULL,
      return.lp=FALSE, apply.snp.profile=NULL, model.snp.info=NULL,
      model.bin.fh.name=NULL, cut.time=NULL, apply.cov.profile.2=NULL,
      model.formula.2=NULL, model.log.RR.2=NULL, model.ref.dataset.2=NULL,
      model.ref.dataset.weights.2=NULL, model.cov.info.2=NULL,
      model.bin.fh.name.2=NULL, n.imp=5, return.refs.risk=FALSE)
```

Arguments

`apply.age.start`
: single integer or vector of integer ages for the start of the interval over which to compute absolute risk.

`apply.age.interval.length`
: single integer or vector of integer years over which absolute risk should be computed.

`apply.cov.profile`
: dataframe containing the covariate profiles for which absolute risk will be computed. Covariates must be in same order with same names as in `model.formula`.

`model.formula`
: an object of class `formula`: a symbolic description of the model to be fitted, e.g. Y~Parity+FamilyHistory.

`model.disease.incidence.rates`
: two column matrix [integer ages, incidence rates] or three column matrix [start age, end age, rate] with incidence rate of disease. Must fully cover age interval for estimation.

`model.log.RR`
: vector with log odds ratios corresponding to the model params; no intercept; names must match design matrix arising from `model.formula` and `model.cov.info`; check names using function `check_design_matrix()`.

`model.ref.dataset`
: dataframe of risk factors for a sample of subjects representative of underlying population, no missing values. Variables must be in same order with same names as in `model.formula`.

`model.ref.dataset.weights`
: optional vector of sampling weights for `model.ref.dataset`.

`model.cov.info`
: contains information about the risk factors in the model ; a main list containing a list for each covariate, which must have the fields:

  - `"name"` : a string with the covariate name, matching name in model.formula
  - `"type"` : a string that is either "continuous" or "factor".

  If factor variable, then:

  - `"levels"` : vector with strings of level names
  - `"ref"` : optional field, string with name of referent level

`use.c.code`
: binary indicator of whether to run the c program for fast computation.

`model.competing.incidence.rates`
: two column matrix [integer ages, incidence rates] or three column matrix [start age, end age, rate] with incidence rate of competing events. Must fully cover age interval for estimation.

`return.lp`
: binary indicator of whether to return the linear predictor for each subject in apply.cov.profile.

`apply.snp.profile`
: data frame with observed SNP data (coded 0,1, 2, or NA). May have missing values.

`model.snp.info`
: dataframe with three columns [rs number, odds ratio, allele frequency]

`model.bin.fh.name`
: string name of family history variable, if in model. This must refer to a variable that only takes values 0,1, NA.

`cut.time`
: integer age for which to split computation into before and after

`apply.cov.profile.2`
: see `apply.cov.profile`, to be used for estimation in ages after the cutpoint

`model.formula.2`
: see `model.formula`, to be used for estimation in ages after the cutpoint

`model.log.RR.2`
: see `model.log.RR`, to be used for estimation in ages after the cutpoint

`model.ref.dataset.2`
: see `model.ref.dataset`, to be used for estimation in ages after the cutpoint

`model.ref.dataset.weights.2`
: see `model.ref.dataset.weights`, to be used for estimation in ages after the cutpoint

`model.cov.info.2`
: see `model.cov.info`, to be used for estimation in ages after the cutpoint

`model.bin.fh.name.2`
: see `model.bin.fh.name`, to be used for estimation in ages after the cutpoint

`n.imp`
: integer value for number of imputations for handling missing SNPs.

`return.refs.risk`
: binary indicator of whether to return the absolute risk prediction for each subject in `model.ref.dataset`.

Details

Individualized Coherent Absolute Risk Estimators (iCARE) is a tool that allows researchers to quickly build models for absolute risk and apply them to estimate individuals' risk based on a set of user defined input parameters. The software gives users the flexibility to change or update models rapidly based on new risk factors or tailor models to different populations based on the specification of simply three input arguments:

- (1) a model for relative risk assumed to be externally derived
- (2) an age-specific disease incidence rate and
- (3) the distribution of risk factors for the population of interest.

The tool can handle missing information on risk factors for risk estimation using an approach where all estimates are derived from a single model through appropriate model averaging.

Value

This function returns a list of results objects, including:

- `risk` : absolute risk estimates over the specified interval for subjects given by `apply.cov.profile`
- `details`: dataframe with the start of the interval, the end of the interval, the covariate profile, and the risk estimates for each individual
- `beta.used` : the log odds ratios used in the model

- `lps.1` : linear predictors based on first set of parameters for subjects in `model.cov.profile`, if requested by `return.lp`

- `lps.2` : linear predictors based on second set of parameters for subjects in `model.cov.profile`, if requested by `return.lp`

- `refs.risk` : absolute risk estimates for subjects in `model.ref.dataset`, if requested by `return.refs.risk`; computes for first age interval provided

Examples

```
data(bc_data, package="iCARE")

results <- computeAbsoluteRiskSplitInterval(model.formula=bc_model_formula,
                              cut.time = 50,
                              model.cov.info       = bc_model_cov_info,
                              model.snp.info       = bc_72_snps,
                              model.log.RR         = bc_model_log_or,
                              model.log.RR.2       = bc_model_log_or_post_50,
                              model.ref.dataset    = ref_cov_dat,
                              model.ref.dataset.2  = ref_cov_dat_post_50,
                              model.disease.incidence.rates   = bc_inc,
                              model.competing.incidence.rates = mort_inc,
                              model.bin.fh.name = "famhist",
                              apply.age.start    = 30,
                              apply.age.interval.length = 40,
                              apply.cov.profile  = new_cov_prof,
                              apply.snp.profile  = new_snp_prof,
                              return.refs.risk   = TRUE)
summary(results)
plot(results)
boxplot(results$risk ~ new_cov_prof$famhist, na.rm=TRUE)
```

`iCARE` *A Tool for Individualized Coherent Absolute Risk Estimation (iCARE)*

Description

Individualized Coherent Absolute Risk Estimators (iCARE) is a tool that allows researchers to quickly build models for absolute risk and apply them to estimate individuals' risk based on a set of user defined input parameters. The software gives users the flexibility to change or update models rapidly based on new risk factors or tailor models to different populations based on the specification of simply three input arguments: (1) a model for relative risk assumed to be externally derived (2) an age-specific disease incidence rate and (3) the distribution of risk factors for the population of interest. The tool can handle missing information on risk factors for risk estimation using an approach where all estimates are derived from a single model through appropriate model averaging.

Details

The main functions for building and applying an absolute risk model are `computeAbsoluteRisk` and `computeAbsoluteRiskSplitInterval`. The first of these computes absolute risks over the specified time interval using a single set of paramters. The second provides more advanced functionality and computes absolute risk over the interval in two parts.

`computeAbsoluteRiskSplitInterval` allows the user compute absolute risk over the interval in two parts, incorporating two different sets of paramters before and after a specified cutpoint. This function allows a different cutpoint for each covariate profile if desired. The function for validating an absolute risk model is `ModelValidation`, and `plotModelValidation` can be called for producing plots for model calibration, model discrimination and incidence rates.

Author(s)

Paige Maas, Parichoy Pal Choudhury, Nilanjan Chatterjee and William Wheeler <wheelerb@imsweb.com>

`ModelValidation` *Model Validation*

Description

This function is used to validate absolute risk models.

Usage

```
ModelValidation(study.data,
                total.followup.validation = FALSE,
                predicted.risk = NULL,
                predicted.risk.interval = NULL,
                linear.predictor = NULL,
                iCARE.model.object =
                  list(model.formula = NULL,
                       model.cov.info = NULL,
                       model.snp.info = NULL,
                       model.log.RR = NULL,
                       model.ref.dataset = NULL,
                       model.ref.dataset.weights = NULL,
                       model.disease.incidence.rates = NULL,
                       model.competing.incidence.rates = NULL,
                       model.bin.fh.name = NA,
                       apply.cov.profile  = NULL,
                       apply.snp.profile = NULL,
                       n.imp = 5, use.c.code = 1,
                       return.lp = TRUE,
                       return.refs.risk = TRUE),
                number.of.percentiles = 10,
                reference.entry.age = NULL,
```

```
                reference.exit.age = NULL,
                predicted.risk.ref = NULL,
                linear.predictor.ref = NULL,
                linear.predictor.cutoffs = NULL,
                dataset = "Example Dataset",
                model.name = "Example Risk Prediction Model")
```

Arguments

`study.data` Data frame which includes the variables below.

- `observed.outcome`: 1 if disease has occurred by the end of followup, 0 if censored
- `study.entry.age`: age (in years) of entering the cohort
- `study.exit.age`: age (in years) of last followup visit
- `time.of.onset`: time (in years) of onset of disease; note that all subjects are disease free at the time of entry and for those who do not develop disease by end of followup it is Inf
- `sampling.weights`: for a case-control study nested within a cohort study, this is a vector of sampling weights for each subject, i.e., probability of inclusion into the sample

`total.followup.validation`
: logical; TRUE if risk validation is performed over the total followup, for all other cases (e.g., 5 year or 10 year risk validation) it is FALSE

`predicted.risk` vector of predicted risks; should be supplied if risk prediction is done by some method other than that implemented in `iCARE`; default is NULL

`predicted.risk.interval`
: scalar or vector denoting the number of years after entering the study over which risk validation is desired (e.g., 5 for validating a model for 5 year risk) if `total.followup.validation = FALSE`; if `total.followup.validation = TRUE`, it can be set to NULL

`linear.predictor`
: vector of risk scores for each subject, i.e. `x*beta`, where `x` is the vector of risk factors and `beta` is the vector of log relative risks; in the current version if both the arguments `predicted.risk` and `linear.predictor` are supplied the function will use the supplied estimates to perform model validation, otherwise the function will compute these estimates using the `computeAbsoluteRisk` function

`iCARE.model.object`
: A named list containing the input arguments to the function `computeAbsoluteRisk`. The names in this list must match the argument names. See `computeAbsoluteRisk`

`number.of.percentiles`
: the number of percentiles of the risk score that determines the number of strata over which the risk prediction model is to be validated, default = 10

`reference.entry.age`
: age of entry to be specified for computing absolute risk of the reference population

`reference.exit.age`
: age of exit to be specified for computing absolute risk of the reference population

`predicted.risk.ref`
: predicted absolute risk in the reference population assuming the entry age to be as specified in `reference.entry.age` and exit age to be as specified in `reference.exit.age`

`linear.predictor.ref`
: vector of risk scores for the reference population

`linear.predictor.cutoffs`
: user specified cut-points for the linear predictor to define categories for absolute risk calibration and relative risk calibration

`dataset`
: name and type of dataset to be displayed in the output, e.g., "PLCO Full Cohort" or "Full Cohort Simulation"

`model.name`
: name of the model to be displayed in output, e.g., "Synthetic Model" or "Simulation Setting"

Value

This function returns a list of the following objects:

- `Subject_Specific_Observed_Outcome`: observed outcome after adjusting the observed followup according to the risk prediction interval: 1 if disease has occurred by the end of followup, 0 if censored
- `Risk_Prediction_Interval`: Character object showing the interval of risk prediction (e.g., 5 years). If the risk prediction is over the total followup of the study, this reads "Observed Followup"
- `Adjusted.Followup`: followup time (in years) after adjusting the observed followup according to the risk prediction interval
- `Subject_Specific_Predicted_Absolute_Risk`: predicted absolute risk of disease for each subject
- `Reference_Absolute_Risk`: predicted absolute risk in the reference population
- `Subject_Specific_Risk_Score`: estimated risk score for each subject; the missing covariates are handled internally using the imputation in `iCARE`
- `Reference_Risk_Score`: risk score for the reference population
- `Population_Incidence_Rate`: age specific disease incidence rate in the population
- `Study_Incidence_Rate`: estimated age specific incidence rate in the study
- `Category_Results`: observed and predicted absolute risks and observed and predicted relative risks in each category defined by the risk score
- `Category_Specific_Observed_Absolute_Risk`: Observed absolute risk in each category defined by the risk score
- `Category_Specific_Predicted_Absolute_Risk`: Predicted absolute risk in each category defined by the risk score
- `Category_Specific_Observed_Relative_Risk`: Observed relative risk in each category defined by the risk score

- Category_Specific_Predicted_Relative_Risk: Predicted relative risk in each category defined by the risk score
- Variance_Matrix_Absolute_Risk: Variance-covariance matrix of the vector of cateogry specific absolute risks
- Variance_Matrix_LogRelative_Risk: Variance-covariance matrix of the vector of cateogry specific relative risks
- Hosmer_Lemeshow_Results: results of the Hosmer-Lemeshow type chisquare test comparing the observed and predicted absolute risks
- HL_pvalue: pvalue of the Hosmer-Lemeshow type chisquare test
- RR_test_result: results of the chisquare test comparing the observed and predicted relative risks
- RR_test_pvalue: pvalue of the chisquare test of relative risk
- AUC: estimate of the Area Under the Curve (AUC) defined as the probability that for a randomly sampled case-control pair the case has a higher risk score than the control; for the full cohort setting we compute the empirical proportion and for the nested case-control setting we compute the inverse probability weighted estimator
- Variance_AUC: estimate of the variance of Area Under the Curve (AUC): for the full cohort setting the regular asymptotic variance is estimated and for the nested case-control setting the influence function based variance estimate of the inverse probability weighted variance estimator is computed
- CI_AUC: 95 percent Wald based confidence interval of Area Under the Curve (AUC) using the asymptotic variance
- Overall_Expected_to_Observed_Ratio: The overall ratio of the expected risk to the observed risk
- CI_Overall_Expected_to_Observed_Ratio: 95 percent Wald based confidence interval of the overall ratio of the expected risk to the observed risk

See Also

computeAbsoluteRisk

Examples

```
data(bc_data, package="iCARE")
validation.cohort.data$inclusion = 0
subjects_included = intersect(validation.cohort.data$id,
                              validation.nested.case.control.data$id)
validation.cohort.data$inclusion[subjects_included] = 1

validation.cohort.data$observed.followup = validation.cohort.data$study.exit.age -
  validation.cohort.data$study.entry.age

selection.model = glm(inclusion ~ observed.outcome
                      * (study.entry.age + observed.followup),
                      data = validation.cohort.data,
                      family = binomial(link = "logit"))
```

```
validation.nested.case.control.data$sampling.weights =
  selection.model$fitted.values[validation.cohort.data$inclusion == 1]

set.seed(50)

data = validation.nested.case.control.data

snpDat     = bc_72_snps
form       = diagnosis ~ famhist + as.factor(parity)
info       = list(bc_model_cov_info[[1]], bc_model_cov_info[[3]])
vars       = all.vars(form)[-1]
risk.model = list(model.formula = form,
                  model.cov.info = info,
                  model.snp.info = snpDat,
                  model.log.RR = bc_model_log_or[c(1, 8:11)],
                  model.ref.dataset = ref_cov_dat[, vars],
                  model.ref.dataset.weights = NULL,
                  model.disease.incidence.rates = bc_inc,
                  model.competing.incidence.rates = mort_inc,
                  model.bin.fh.name = "famhist",
                  apply.cov.profile = data[,vars],
                  apply.snp.profile = data[,snpDat$snp.name],
                  n.imp = 5, use.c.code = 1, return.lp = TRUE,
                  return.refs.risk = TRUE)

# Not run since it can take a few minutes
# output = ModelValidation(study.data = data, total.followup.validation = TRUE,
#      predicted.risk.interval = NULL, iCARE.model.object = risk.model,
#      number.of.percentiles = 10)
output
```

---

plotModelValidation *Model Validation Plot*

Description

This function is used to create plots for model calibration, model discrimination and incidence rates.

Usage

```
plotModelValidation(study.data, validation.results,
                    dataset = "Example Dataset",
                    model.name = "Example Model",
                    x.lim.absrisk = "",
                    y.lim.absrisk = "",
                    x.lab.absrisk = "Expected Absolute Risk (%)",
                    y.lab.absrisk = "Observed Absolute Risk (%)",
                    x.lim.RR = "",
                    y.lim.RR = "", x.lab.RR = "Expected Relative Risk",
```

```
                    y.lab.RR = "Observed Relative Risk",
                    risk.score.plot.kernel = "gaussian",
                    risk.score.plot.bandwidth = "nrd0",
                    risk.score.plot.percent.smooth = 50)
```

Arguments

study.data	See ModelValidation
validation.results	List returned from ModelValidation
dataset	Name and type of dataset to be displayed in the output, e.g., "PLCO Full Cohort" or "Full Cohort Simulation"
model.name	Name of the model to be displayed in output, e.g., "Synthetic Model" or "Simulation Setting"
x.lim.absrisk	Vector of length two specifying the x-axes limits in the absolute risk calibration plot. If not specified, then default limits will be computed.
y.lim.absrisk	Vector of length two specifying the y-axes limits in the absolute risk calibration plot. If not specified, then default limits will be computed.
x.lab.absrisk	String specifying the x-axes label in the absolute risk calibration plot. The default is "Expected Absolute Risk (%)".
y.lab.absrisk	String specifying the y-axes label in the absolute risk calibration plot. The default is "Observed Absolute Risk (%)."
x.lim.RR	Vector of length two specifying the x-axes limits in the relative risk calibration plot. If not specified, then default limits will be computed.
y.lim.RR	Vector of length two specifying the y-axes limits in the relative risk calibration plot. If not specified, then default limits will be computed.
x.lab.RR	String specifying the x-axes label in the relative risk calibration plot. The default is "Expected Relative Risk".
y.lab.RR	String specifying the y-axes label in the relative risk calibration plot. The default is "Observed Relative Risk".
risk.score.plot.kernel	Character string giving the smoothing kernel to be used by the density function used internally to plot the density of the risk scores. It should be one of "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine", with default "gaussian".
risk.score.plot.bandwidth	The options for bandwidth selection used by the density function internally to plot the density of the risk scores. The options are one of the following: "nrd0", "nrd", "ucv", "bcv", "SJ-ste", "SJ-dpi" with the default being "nrd0". More information on these different options is available in the help pages that can be accessed from R using the command ?bw.nrd.
risk.score.plot.percent.smooth	Percentage of the number of sample points used for determining the number of equally spaced points at which the density of the risk score is to be estimated. This number supplies the input parameter "n" to the density function used internally to plot the densities of the risk score. The default value is 50.

Value

This function returns NULL

See Also

ModelValidation

Examples

```
data(bc_data, package="iCARE")
validation.cohort.data$inclusion = 0
subjects_included = intersect(validation.cohort.data$id,
                              validation.nested.case.control.data$id)
validation.cohort.data$inclusion[subjects_included] = 1

validation.cohort.data$observed.followup = validation.cohort.data$study.exit.age -
  validation.cohort.data$study.entry.age

selection.model = glm(inclusion ~ observed.outcome
                      * (study.entry.age + observed.followup),
                      data = validation.cohort.data,
                      family = binomial(link = "logit"))

validation.nested.case.control.data$sampling.weights =
  selection.model$fitted.values[validation.cohort.data$inclusion == 1]

set.seed(50)

data = validation.nested.case.control.data

snpDat     = bc_72_snps
form       = diagnosis ~ famhist + as.factor(parity)
info       = list(bc_model_cov_info[[1]], bc_model_cov_info[[3]])
vars       = all.vars(form)[-1]
risk.model = list(model.formula = form,
                  model.cov.info = info,
                  model.snp.info = snpDat,
                  model.log.RR = bc_model_log_or[c(1, 8:11)],
                  model.ref.dataset = ref_cov_dat[, vars],
                  model.ref.dataset.weights = NULL,
                  model.disease.incidence.rates = bc_inc,
                  model.competing.incidence.rates = mort_inc,
                  model.bin.fh.name = "famhist",
                  apply.cov.profile = data[,vars],
                  apply.snp.profile = data[,snpDat$snp.name],
                  n.imp = 5, use.c.code = 1, return.lp = TRUE,
                  return.refs.risk = TRUE)

# Not run since it can take a few minutes
#output = ModelValidation(study.data = data, total.followup.validation = TRUE,
#          predicted.risk.interval = NULL, iCARE.model.object = risk.model,
```

```
#          number.of.percentiles = 10)

plot(output)
```

Index

∗ **data**
    bc_data, 2
∗ **package**
    iCARE, 10

bc_72_snps *(*bc_data*)*, 2
bc_data, 2
bc_inc *(*bc_data*)*, 2
bc_model_cov_info *(*bc_data*)*, 2
bc_model_formula *(*bc_data*)*, 2
bc_model_log_or *(*bc_data*)*, 2
bc_model_log_or_post_50 *(*bc_data*)*, 2

computeAbsoluteRisk, *2, 3*, 4, *11, 12*, *14*
computeAbsoluteRiskSplitInterval, *2*, 7, *11*

iCARE, 10

ModelValidation, *2*, *11*, 11, *16, 17*
mort_inc *(*bc_data*)*, 2

new_cov_prof *(*bc_data*)*, 2
new_snp_prof *(*bc_data*)*, 2

output *(*bc_data*)*, 2

plot *(*plotModelValidation*)*, 15
plotModelValidation, *2*, *11*, 15

ref_cov_dat *(*bc_data*)*, 2
ref_cov_dat_post_50 *(*bc_data*)*, 2

validation.cohort.data *(*bc_data*)*, 2
validation.nested.case.control.data *(*bc_data*)*, 2