Package 'SIPI'

January 19, 2024

Type Package

Title SIPI

Version 1.24

Date 2024-01-19

Depends SNPassoc, car, lmtest, parallel, pscl, ggplot2, MASS, ggpubr

Author Hui-Yi Lin and Po-Yu Huang

Maintainer Po-Yu Huang <jettly32@gmail.com>

Description Testing single nucleotide polymorphism (SNP) interactions is considered as a key for overcoming bottlenecks of genetic association studies. SNP Interaction Pattern Identifier (SIPI) evaluates SNP-SNP interactions associated with a binary or continuous outcome. The primary strengths of SIPI are (1) taking non-hierarchical models, reverse coding and inheritance modes (dominant, recessive and additive mode) into consideration and (2) using the Bayesian information criterion (BIC) to search for a best interaction pattern. For each SNP pair, the SIPI evaluates 45 interaction models. The best interaction pattern is the one with the lowest BIC value.

Reference (1) Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167. (2) Lin HY, Huang PY, Chen DT, Tung HY, Sellers TA, Pow-Sang J, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Neal DE, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, consortium P, Park JY. AA9int: SNP Interaction Pattern Search Using Non-Hierarchical Additive Model Set. Bioinformatics. 2018. doi: 10.1093/bioinformatics/bty461. PubMed PMID: 29878078.

LazyData true

License MIT + file LICENSE

RoxygenNote 7.1.2

NeedsCompilation no

R topics documented:

`AA9int` *AA9int (Additive-Additive 9 interaction models): Detect SNP-SNP interactions through testing the 9 models*

Description

`AA9int` (Additive-Additive 9 interaction models), a mini version of `SIPI`, evaluates SNP-SNP interactions associated with a binary or continuous outcome through testing nine interaction models. `AA9int` treats both SNPs as an additive mode and takes reverse coding and non-hierarchical models into consideration. The best interaction pattern is the one with the lowest value of the Bayesian information criterion (BIC). The details of the nine models/patterns are listed in the reference.

Usage

```
AA9int(Outcome, SNPdata, PairInfo, X = NULL, categXNames = NULL,
       TestType = "WaldTest", ModelType = "binomial", OR = FALSE)
```

Arguments

`Outcome`	Binary (`1`: event of interest; `0`: reference) or continuous variable.
`SNPdata`	SNP data: All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. The missing values of SNPs need to be defined as "NA", such as snpdata[snpdata=="–"] = NA, where "–" is originally defined symbol of missing value.
`PairInfo`	3 types of PairInfo: (1) 2d-vector: names of the given SNP pair for one-pair analyses. ex: c("`SNP1`","`SNP2`") (2) 2d-matrix or 2d-dataframe: names for candidate SNP pair.

(3) "all": for pairwise analyses.

(4) a list containing two 1d-vectors: for pairwise between these two vectors.

X — Covariate(s) to be adjusted in the model (for missing values, keep the field blank), NULL=without covariate. Default is NULL.

categXNames — The variable names of categorical variables, NULL=without categorical covariates. Default is NULL.

TestType — Specify the statistical test type: "WaldTest" = the Wald test; "LRT" = the likelihood ratio test. Default is "WaldTest".

ModelType — Model type: "binomial"=logistic regression; "gaussian"=linear regression. Default is "binomial".

OR — If TRUE print the odds ratios, 95% confidence intervals and corresponding p-values.

Value

Returns a list with the following attributes:

selectedModel — The results of the best model with the lowest BIC value among the 9 models.

res9Models — For one-pair analyses only: Detailed results with all 9 models sorted by BIC (lowest first). Output variables: Var1: SNP 1; Var2: SNP 2 (the pattern/model labels are based on this order); Model: interaction model/pattern; Wald_Chisq: the Wald chi-square value of the interaction term; Wald_p: the Wald p-value of the interaction term; LRT_Chisq: the chi-square value of likelihood ratio test (LRT) for the interaction term; LRT_p: the LRT p-value of the interaction term; BIC: the Bayesian information criterion. The model with the lowest BIC value is preferred.

OR — For the "all" pairs analyses, only results of the best model will show. The group coding, please see suppl. Figures 1-3 in the end of this manual.

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167.

Lin HY, Huang PY, Chen DT, Tung HY, Sellers TA, Pow-Sang J, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Neal DE, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, consortium P, Park JY. AA9int: SNP Interaction Pattern Search Using Non-Hierarchical Additive Model Set. Bioinformatics. 2018. doi: 10.1093/bioinformatics/bty461. PubMed PMID: 29878078.

See Also

parAA9int, SIPI, parSIPI

Examples

```
##load data
data(simData)

#### define SNP data
SNPdata = simData[,3:12]

########################
### run AA9int one-pair analyses <Wald test>
########################

##  For a SNP pair of SNP2 and SNP8
res_snp_2v8 = AA9int(simData$D,SNPdata,c("SNP2","SNP8"))

## print out the best model
res_snp_2v8$selectedModel

## list of 9 models for one SNP pair
res_snp_2v8$res9Models


########################
### run AA9int for a list of multiple SNP pairs <Wald test>
########################

##  For 5 SNP pairs(1 vs. 6, 2 vs. 7, 3 vs. 8, 4 vs. 9, and 5 vs. 10)
pairMatrix = c("SNP1","SNP2","SNP3","SNP4","SNP5","SNP6","SNP7","SNP8","SNP9","SNP10")

pairMatrix = matrix(pairMatrix,5)

pairMatrix = as.data.frame(pairMatrix)

res_snp = AA9int(simData$D,SNPdata,pairMatrix)


########################
### run SIPI for all possible combinations of SNP pairs between two vectors <Wald test>
########################
##  For 9 SNP pairs(1 vs. 6, 2 vs. 6, 3 vs. 6, 1 vs. 7, and 2 vs. 7, ..., 3 vs. 8)
vetor_1 = c("SNP1","SNP2","SNP3")
vetor_2 = c("SNP6","SNP7","SNP8")

res_snp = AA9int(simData$D,SNPdata,list(vetor_1,vetor_2))


########################
### run AA9int pairwise analyses <Wald test>
########################

res_all = AA9int(simData$D,SNPdata,"all")
```

```
########################
### run AA9int pairwise analyses <Wald test> adjusted for covariates
###  [age(numeric), gender(binary), and group(categorical)]
########################

X1 = simData[,c("age","gender","group")]

res_all_cov = AA9int(simData$D,SNPdata,"all",X1,c("gender","group"))

########################
## export ORs and p-values
########################

try = AA9int(simData$D,SNPdata,"all",OR=TRUE)
write_OR_csv(try$OR,'D:/OR.csv')
write_OR_csv(try$selectedModel,'D:/p.csv')
```

`boot3p_SIPI` *Identification of SNP-SNP interactions using the SIPI method with the 3 p-value rule and the Bootstrap approach*

Description

In order to increase accuracy and reduce false positivity in SNP-SNP interactions identified by SIPI, the 3pRule + bootstrap internal validation is suggested. Using the bootstrap approach, subjects are randomly selected with replacements based on the same sample size from the data. This bootstrap procedure mimics the sampling variation in the population from which the sample was drawn. Significance of SNP-SNP interaction pairs is based on the 3pRule: (1) p_pair < pvalCutoff, (2) p_pair < p_SNP1, and (3) p_pair < p_SNP2.

Usage

```
boot3p_SIPI(data, PairList, outcomeName, XNames = NULL, categXNames = NULL,
            ModelType = "binomial", pvalCutoff, n.boot = 100, seed = NULL)
```

Arguments

`data` a data.frame/data.table/matrix with all SNP variables, covariates to be adjusted and outcome variable(s)(binary or continuous). All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). The missing values of SNPs need to be defined as "NA.".
The data type for variables used as categorical in modeling, which will be included in "categXNames", should be a character (chr)/categorical variable. Other covariates in modeling should be numeric (num).
Example codes
check data structure
Str(dat)
example codes of changing variable type from "chr" to "num" or a variable
dat$new_var = as.numeric(as.character(dat$old_var))

PairList	SNP Pairs: must be a data frame with two variables. 1st variable is the list of names of 1st SNP of the pairs, and 2nd variable is the name of 2nd SNP of the pairs. See the following example for details.
outcomeName	Outcome variable name. Outcome types could be binary (1: event of interest; 0: reference) or continuous.
XNames	Name(s) of Covariate(s) to be adjusted in the model (for missing values, keep the field blank),NULL=without covariate. Default is NULL
categXNames	The variable names of categorical variables, NULL=without categorical covariates. Default is NULL.
ModelType	Model type: "binomial"=logistic regression; "gaussian"=linear regression. The default is "binomial."
pvalCutoff	Cut-point of p-value to define promising SNP-SNP interaction pairs, such as the Bonferroni criterion (0.05 divided by the number of pairs).
n.boot	Set up the number of bootstrap datasets. The Default is 100.
seed	Use a specific seed to reproduce the results later. Default is NULL.

Value

Pairs	Names of SNP-SNP interaction pairs
Prop_sig	The proportion of significance based on the 3pRule

Author(s)

Hui-Yi Lin and Harun Or Rashid Mazumder

See Also

SIPI

Examples

```
## load data
data(simData)

## Perform bootstrap validation for 5 SNP-interaction pairs:
## SNP1-SNP3, SNP1-SNP4, SNP1-SNP5, SNP1-SNP6, and SNP1-SNP7
## define SNP pairs
snp1 = c("SNP1", "SNP1", "SNP1","SNP1","SNP1")
snp2 = c("SNP3", "SNP4", "SNP5", "SNP6", "SNP7")
snp_pairs = data.frame(snp1, snp2)

## Perform bootstrap validation with 50 runs for the selected 5 pairs
## without adjusting for other variables
boot3p_SIPI(PairList=snp_pairs, outcomeName="D", data=simData,
            pvalCutoff = 10^(-4), n.boot = 100, seed=100)

## Perform bootstrap validation with 50 runs for the selected 5 pairs
## adjusting 2 variables: age (continuous) and group (categorical)
boot3p_SIPI(PairList=snp_pairs, outcomeName="D", XNames=c("age","group"),
            categXNames="group", data=simData, ModelType="binomial",
            pvalCutoff = 10^(-4),n.boot = 100, seed=100)
```

`bootData` *This bootData function allows users to extract bootstrap data based on a set of selected SNPs with a given seed*

Description

`bootData` is represented as bootstrap data. This bootData function allows users to extract bootstrap data based on selected SNPs with a given seed. With the same seed, the same bootstrap data used for boot3p_SIPI can be obtained.

Usage

```
bootData(data, SNPselect, Xname = NULL, IDname = NULL, outcomeName = NULL,
         n.boot = 100, bindMethod = "rowbind", seed = NULL)
```

Arguments

`data`	A data.frame/data.table/matrix with all SNP variables, covariates to be adjusted and outcome variable(s)(binary or continuous). All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). The missing values of SNPs need to be defined as "NA.".
`SNPselect`	List of selected SNPs, such as SNPselect = c("SNP1","SNP3", "SNP5", "SNP6") to be included in the bootstrap data.
`Xname`	Covariate(s) to be adjusted in the model (for missing values, keep the field blank), `NULL`=without covariate. Default is NULL.
`IDname`	Include subject ID column (e.g., PatientID) in the bootstrap data. `NULL`=no subject ID to show in bootstrap samples; Default is NULL.
`outcomeName`	Outcome variable name. Outcome types could be binary (`1`: event of interest; `0`: reference) or continuous.
`n.boot`	Set up the number of bootstrap datasets. The Default is 100.
`bindMethod`	Method of merging multiple bootstrap datasets. `'rowbind'` appends bootstrap datasets one after another in the row direction. `'colbind'` merges bootstrap datasets side by side in the column direction. The default is `'rowbind'`.
`seed`	Set a specific seed to fix the bootstrap process so the same results can be reproduced. Default is NULL.

Value

`sample_index`	Indicator of bootstrap datasets (such as 1, 2,...) for bindMethod=`'rowbind'`.
`_1, _2, _3, ...`	For bindMethod=`'colbind'`, suffix (_1, _2, _3,...) is added to the original variable names to indicate bootstrap datasets.

Author(s)

Hui-Yi Lin and Harun Or Rashid Mazumder

See Also

`boot3p_SIPI`

Examples

```
## load data
data(simData)

## Generate bootstrap datasets with SNPs and other non-SNP variables by appending
## datasets one after another in the row direction.
d = bootData(data=simData, SNPselect = c("SNP1","SNP2","SNP4"),
             Xname = c("age", "group"), IDname = "id", outcomeName="D",
             bindMethod="rowbind", n.boot=2, seed=100)
head(d)
dim(d)

## data with SNP data only
d1 = bootData(data=simData, SNPselect = c("SNP1","SNP2"), n.boot=2, seed=100)
head(d1)
dim(d1)

## Generate bootstrap datasets with SNPs and other non-SNP variables by merging
## datasets side by side in the column direction.
d2 = bootData(data=simData, SNPselect = c("SNP1","SNP2","SNP4"),
              Xname = c("age", "group"), IDname = "id", outcomeName="D",
              bindMethod="colbind", n.boot=2, seed=100)
head(d2)
dim(d2)
```

`eval3pRule` *Define significant SNP-SNP interaction pairs based on 3 p-value rules to reduce false positivity*

Description

The `"eval3pRule"`method define significant SNP-SNP interaction pairs based on 3 p-value rules: (1) p-pair < p-pair-criterion, (2) p-pair < p-SNP1, and (3) p-pair < p-SNP2. p-pair-criterion is the selected p-value criterion of promising SNP-SNP interaction pairs. p-SNP1 is the p-value of 1st composite SNP, and p-SNP2 is the p-value of 2nd composite SNP of the same SNP-SNP interaction pair. The significance of SNP-SNP interactions is affected by the significance level of composite SNPs' individual or main effects. The 3pRule application is suggested for reducing false-positive findings.

Usage

```
eval3pRule(MAINres, SIPIres, pvalCutoff)
```

Arguments

`MAINres`	SNP main effect results are generated using the `SNPmain()` function.
`SIPIres`	SIPI SNP-SNP interaction results, generated using `SIPI()$selectedModel` function
`pvalCutoff`	A cut-point of the p-value (Bonferroni criterion) to define promising SNP-SNP interactions

Value

Var1	SNP1, which is the 1st composite SNP in an SNP pair
Var2	SNP2, which is the 2nd composite SNP in an SNP pair (the pattern labels are based on the order of SNP1 and SNP2)
Pattern	interaction model/pattern
p_pair	the Wald p-value of the interaction term
p_Var1	the Wald p-value of individual effect of SNP1
p_Var2	the Wald p-value of individual effect of SNP2
sig.status	significance status based on qualification of all 3 rules (1) p_pair < pvalCutoff, (2) p_pair < p_SNP1, and (3) p_pair < p_SNP2

Author(s)

Hui-Yi Lin and Harun Or Rashid Mazumder

See Also

SNPmain, SIPI

Examples

```
##load data
data(simData)

## Calculate SNP individual or main effects associated with the outcome variable D
MAINres = SNPmain(simData$D, SNPdata=simData[, 3:7], PairInfo="all")

## Perform SNP-SNP interactions associated with the outcome variable
SIPIres = SIPI(simData$D, SNPdata=simData[, 3:7], PairInfo="all")$selectedModel

## Perform 3pRule to identify significant SNP-SNP interactions
res = eval3pRule (MAINres, SIPIres, pvalCutoff = 0.05/10)
res
```

Grid3by3 *Outcome proportions by genotype combinations of two SNPs*

Description

Outcome proportions by the 3-by-3 genotype combinations of a give SNP pair.

Usage

```
Grid3by3(Outcome, SNPdata, PairInfo)
```

Arguments

Outcome	Binary outcome variable name: a binary variable with "1" as the event of interest and "0" as the reference.
SNPdata	SNP data. All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. An invalid character or blank field are considered to be missing values.
PairInfo	c("SNP1","SNP2"): names of the given SNP pairs for one-pair analyses

Value

maj_min	Major and minor allele
table3by3	Present outcome proportions by genotype combinations
table3by3Freq	Sample size by genotype combinations

Author(s)

Hui-Yi Lin and Po-Yu Huang

See Also

plot3by3

Examples

```
##load data
data(simData)

#### define SNP data
SNPdata = simData[,3:12]

########################
### run Grid3by3
########################
Grid3by3(simData$D,SNPdata,c("SNP1","SNP2"))
```

GridSNPxE — *Outcome proportions by the combinations of SNP and environmental factor*

Description

Outcome proportions by the combinations of a give SNP and environmental factor.

Usage

```
GridSNPxE(Outcome, SNPdata, Env)
```

Arguments

Outcome	Binary outcome variable name: a binary variable with "1" as the event of interest and "0" as the reference.
SNPdata	SNP data. SNP variable should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. An invalid character or blank field are considered to be missing values.
Env	Environment variable

Value

maj_min	Major and minor allele
table3by3	Present outcome proportions by genotype combinations
table3by3Freq	Sample size by genotype combinations

Author(s)

Hui-Yi Lin and Po-Yu Huang

See Also

plotSNPxE

Examples

```
##load data
data(simData2)

#### define SNP data
SNPdata = simData2[,9]

#### define Env data
Env = simData2$env_g2

########################
### run GridSNPxE
########################
GridSNPxE(simData2$D,SNPdata,Env)
```

MAFinfo *SNP minor allele frequency (MAF)*

Description

Obtain minor allele frequency (MAF) and major and minor allele.

Usage

```
MAFinfo(SNPdata)
```

Arguments

SNPdata	SNP data

Value

maj/min	major and minor allele
MAF	minor allele frequency
Missing (%)	missing vales
No_genotype	number of genotypes

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Gonzalez JR, Armengol L, Sole X, Guino E, Mercader JM, Estivill X, Moreno V. SNPassoc: an R package to perform whole genome association studies. Bioinformatics, 2007;23(5):654-5.

Examples

```
data(simData)
SNPdata = simData[,3:12]
MAFinfo(SNPdata)
```

---

parAA9int *Parallel computing for AA9int*

Description

parAA9int is a parallel computing version of AA9int. This function can decrease computing time, which is useful for large-scale data.

Usage

```
parAA9int(Outcome, SNPdata, PairInfo, X = NULL, categXNames = NULL,
          TestType = "WaldTest", ModelType = "binomial",
             core_ratio = 0.9, OR = FALSE)
```

Arguments

Outcome	Binary (1: event of interest; 0: reference) or continuous variable.
SNPdata	SNP data: All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. No other letters or numbers should be used. The missing values of SNPs need to be defined as "NA", such as snpdata[snpdata=="–"] = NA, where "–" is originally defined symbol of missing value.
PairInfo	3 types of PairInfo: (1) 2d-vector: names of the given SNP pair for one-pair analyses. ex: c("SNP1","SNP2") (2) 2d-matrix or 2d-dataframe: names for candidate SNP pair.

	(3) "all": for pairwise analyses.
	(4) a list containing two 1d-vectors: for pairwise between these two vectors.
X	Covariate(s) to be adjusted in the model (for missing values, keep the field blank), NULL=without covariate. Default is NULL.
categXNames	The variable names of categorical variables, NULL=without categorical covariates. Default is NULL.
TestType	Specify the statistical test type: "WaldTest" = the Wald test; "LRT" = the likelihood ratio test. Default is "WaldTest".
ModelType	Model type: "binomial"=logistic regression; "gaussian"=linear regression. Default is "binomial".
core_ratio	The ratio of total cores for parallel computing. Default is 0.9.
OR	If TRUE print the odds ratios, 95% confidence intervals and corresponding p-values.

Value

Returns a data frame of the results of the best model with the lowest BIC value among the 9 models. For the "all" pairs analyses, only OR results of the best model will show. The group coding, please see suppl. Figures 1-3 in the end of this manual.)

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167.

Lin HY, Huang PY, Chen DT, Tung HY, Sellers TA, Pow-Sang J, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Neal DE, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, consortium P, Park JY. AA9int: SNP Interaction Pattern Search Using Non-Hierarchical Additive Model Set. Bioinformatics. 2018. doi: 10.1093/bioinformatics/bty461. PubMed PMID: 29878078.

See Also

AA9int, SIPI, and parSIPI

Examples

```
##load data
data(simData)

#### define SNP data
```

```
SNPdata = simData[,3:12]

########################
### run parAA9int pairwise analyses <Wald test>
########################

res_all = parAA9int(simData$D,SNPdata,"all")


########################
###run parAA9int pairwise analyses <Wald test> adjusted for covariates using SIPI with pa
###[age(numeric), gender(binary), and group(categorical)]
########################

X1 = simData[,c("age","gender","group")]
res_all_X = parAA9int(simData$D,SNPdata,"all",X1,c("gender","group"))

########################
## export ORs and p-values
########################

try = parAA9int(simData$D,SNPdata,"all",OR=TRUE)
write_OR_csv(try$OR,'D:/OR.csv')
write_OR_csv(try$selectedModel,'D:/p.csv')
```

`parSIPI` *Parallel computing for SIPI*

Description

`parSIPI` is a parallel computing version of `SIPI`. This function can decrease computing time, which is useful for large-scale data.

Usage

```
parSIPI(Outcome, SNPdata, PairInfo, X = NULL, categXNames = NULL,
        TestType = "WaldTest", ModelType = "binomial",
        core_ratio = 0.9, OR = FALSE)
```

Arguments

`Outcome`	Binary (`1`: event of interest; `0`: reference) or continuous variable.
`SNPdata`	SNP data: All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. The missing values of SNPs need to be defined as "NA", such as snpdata[snpdata=="–"] = NA, where "–" is originally defined symbol of missing value.
`PairInfo`	3 types of PairInfo: (1) 2d-vector: names of the given SNP pair for one-pair analyses. ex: c(`"SNP1"`,`"SNP2"`) (2) 2d-matrix or 2d-dataframe: names for candidate SNP pair. (3) `"all"`: for pairwise analyses.

	(4) a list containing two 1d-vectors: for pairwise between these two vectors.
X	Covariate(s) to be adjusted in the model (for missing values, keep the field blank), NULL=without covariate. Default is NULL.
categXNames	The variable names of categorical variables, NULL=without categorical covariates. Default is NULL.
TestType	Specify the statistical test type: "WaldTest" = the Wald test; "LRT" = the likelihood ratio test. Default is "WaldTest".
ModelType	Model type: "binomial"=logistic regression; "gaussian"=linear regression. Default is "binomial".
core_ratio	The ratio of total cores for parallel computing. Default is 0.9.
OR	If TRUE print the odds ratios, 95% confidence intervals and corresponding p-values.

Value

Returns a data frame of the results of the best model with the lowest BIC value among the 45 models for each SNP pair. For the "all" pairs analyses, only OR results of the best model will show. The group coding, please see suppl. Figures 1-3 in the end of this manual.)

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167.

See Also

SIPI

Examples

```
##load data
data(simData)

#### define SNP data
SNPdata = simData[,3:12]

########################
### run parSIPI pairwise analyses <Wald test>
########################

res_all = parSIPI(simData$D,SNPdata,"all")


########################
```

```
###run parSIPI pairwise analyses <Wald test> adjusted for covariates using SIPI with para
###[age(numeric),gender (binary), and group(categorical)]
########################

X1 = simData[,c("age","gender","group")]
res_all_X = parSIPI(simData$D,SNPdata,"all",X1,c("gender","group"))

########################
## export ORs and p-values
########################

try = parSIPI(simData$D,SNPdata,"all",OR=TRUE)
write_OR_csv(try$OR,'D:/OR.csv')
write_OR_csv(try$selectedModel,'D:/p.csv')
```

`plot3by3` *Heat table of outcome proportions by genotype combinations*

Description

Create a heat table of sample sizes and outcome proportions by the 3-by-3 genotype combinations for a given SNP pair.

Usage

```
plot3by3(x, SNP_info = T, outcome = T, freq = T, legend = T, monochrome = F,
         scale = "fixed", axis_fs = 1, outcome_fs = 1, freq_fs = 1, lgd_fs = 1,
         marginal = F)
```

Arguments

`x`	List object output from function `Grid3by3`.
`SNP_info`	Put SNP information(SNP name and major/minor allele) on plot. Default is `TRUE`.
`outcome`	Include outcome proportions in each cell of plot. Default is `TRUE`. When no observations is in the given cell, 'NaN' will be shown. If there is no observation and `outcome=FALSE`, a warning will be shown.
`freq`	Include frequency in each cell. Default is `TRUE`.
`legend`	Include legend. Default is `TRUE`.
`monochrome`	Output monochrome plot. Default is `FALSE`
`scale`	A character string specifying the colour gradient scale type. `"fixed"` will lend color to heatmap with fixed color gradient scale from 0 to 1, `"sliding"` will lend color to heatmap with sliding gradient scale between minimum and maximum outcome proportion. Default is `"fixed"`.
`axis_fs`	Axis font size. Adjusted both axis title font size and axis label font size. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is `1`.
`outcome_fs`	Outcome font size. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is `1`.

freq_fs	Frequency font sizr. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is 1.
lgd_fs	Legend font size. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is 1.
marginal	Show details for the genotypes of a individual SNP.

Details

This function creates a heat table based on the output of Grid3by3, which generates outcome proportions by genotype combinations of a given SNP pair.

Value

A heat table of outcome proportions.

Author(s)

Hui-Yi Lin and Heng-Yuan Tung

References

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

See Also

Grid3by3

Examples

```
##load data
data(simData)

#### define SNP data
SNPdata = simData[,3:12]

#########################
### run plot3by3
#########################
x = Grid3by3(simData$D, SNPdata, c('SNP1', 'SNP2'))
plot3by3(x, SNP_info = F, outcome = F, legend = T, scale = "fixed", monochrome = T, lgd_f

x = Grid3by3(simData$D, SNPdata, c('SNP4', 'SNP6'))
plot3by3(x, scale = "sliding", axis_fs = 1.2, outcome_fs = 0.9, freq_fs = 0.8, marginal =
plot3by3(x, scale = "sliding", freq = F, axis_fs = 1.2, outcome_fs = 0.9, marginal = T)
```

`plotSNPxE` *Heat table of outcome proportions by combinations of SNP and environmental factor*

Description

Create a heat table of sample sizes and outcome proportions by the combinations of a give SNP and environmental factor.

Usage

```
plotSNPxE(x, SNP_info = T, outcome = T, freq = T, legend = T, monochrome = F,
          scale = "fixed", axis_fs = 1, outcome_fs = 1, freq_fs = 1, lgd_fs = 1,
          marginal = F)
```

Arguments

`x`	List object output from function `GridSNPxE`.
`SNP_info`	Put SNP information(SNP name and major/minor allele) on plot. Default is `TRUE`.
`outcome`	Include outcome proportions in each cell of plot. Default is `TRUE`. When no observations is in the given cell, 'NaN' will be shown. If there is no observation and `outcome=FALSE`, a warning will be shown.
`freq`	Include frequency in each cell. Default is `TRUE`.
`legend`	Include legend. Default is `TRUE`.
`monochrome`	Output monochrome plot. Default is `FALSE`
`scale`	A character string specifying the colour gradient scale type. `"fixed"` will lend color to heatmap with fixed color gradient scale from 0 to 1, `"sliding"` will lend color to heatmap with sliding gradient scale between minimum and maximum outcome proportion. Default is `"fixed"`.
`axis_fs`	Axis font size. Adjusted both axis title font size and axis label font size. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is `1`.
`outcome_fs`	Outcome font size. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is `1`.
`freq_fs`	Frequency font sizr. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is `1`.
`lgd_fs`	Legend font size. Number greater than 1 will enlarge the font size, less than 1 will reduce the font size. Default is `1`.
`marginal`	Show details for the genotypes of a individual SNP.

Details

This function creates a heat table based on the output of `GridSNPxE`, which generates outcome proportions by the combinations of a give SNP and environmental factor.

Value

A heat table of outcome proportions.

Author(s)

Hui-Yi Lin and Heng-Yuan Tung

References

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

Lin HY, Huang PY, Tseng TS, Park JY. SNPxE: SNP-environment interaction pattern identifier. BMC Bioinformatics. 2021;22(1):425. PubMed PMID: 34493206; PMCID: PMC8425112.

See Also

Grid3by3

Examples

```
##load data
data(simData2)

#### define SNP data
SNPdata = simData2[,9]

#### define Env data
Env = simData2$env_g2

########################
### run plotSNPxE
########################
Outcome = simData2$D
x = GridSNPxE(Outcome,SNPdata,Env)
plotSNPxE(x, SNP_info = T, outcome = T, freq = T, legend = T,
          monochrome = F, scale = "fixed", marginal = T,
          axis_fs = 1, outcome_fs = 1, freq_fs = 1, lgd_fs = 1)
```

---

simData *An example data set for testing SNP-SNP interactions*

Description

simData is an example dataset with one binary outcome variable (D), the 10 SNPs, and three covariates [age (numeric), gender (binary), and group (categorical)].

Usage

```
data(simData)
```

Format

A data frame with 1000 observations on the following 15 variables.

`id` A numeric vector for identification.

`D` A numeric vector for a binary outcome with `"1"` as the event of interest and `"0"` as the reference.

`SNP1 to SNP10` A factor has three genotypes, which are composed of two of the four letters (C, T, A, and G), such as CC, TC and TT.

`age` numeric age.

`gender` `0`: female, `1`: male.

`group` `1`: Group 1; `2`: Group 2; and `3`: Group 3.

Examples

```
data(simData)
```

`simData2` *An example data set for testing SNP-environment interactions*

Description

`simData2` is an example dataset with one binary outcome variable (D), the 5 SNPs, three covariates [cov1 (numeric), cov2 (numeric), and group (categorical)], and three environment factors [env_g2 (binary), env_g3 (categorical), and env_level (numeric)].

Usage

```
data(simData2)
```

Format

A data frame with 2000 observations on the following 13 variables.

`id` A numeric vector for identification.

`D` A numeric vector for a binary outcome with `"1"` as the event of interest and `"0"` as the reference.

`cov1` control numeric covariate 1

`cov2` control numeric covariate 2

`env_g2` evels of environment factor, `0`: low, `1`: high.

`group` `1`: Group 1, `2`: Group 2, and `3`: Group 3.

`env_level` levels of environment factor (continuous)

`env_g3` levels of environment factor, `1`: low, `2`: medium, `3`: high.

`snp1 to snp5` SNP data with 3 genotypes, such as AA, AG and GG.

Examples

```
data(simData2)
```

SIPI *SNP Interaction Pattern Identifier (SIPI): Detect SNP-SNP interactions through testing the 45 models*

Description

SNP Interaction Pattern Identfier (SIPI) evaluates SNP-SNP interactions associated with a binary or continuous outcome. The primary strengths of SIPI are (1) taking non-hierarchical models, reverse coding and inheritance modes (dominant, recessive and additive mode) into consideration and (2) using BIC to search for a best interaction pattern. For each SNP pair, the SIPI evaluates 45 interaction models. The best interaction pattern is the one with the lowest value of the Bayesian information criterion (BIC). The details of the 45 models/patterns are listed in the SIPI published paper.

Usage

```
SIPI(Outcome, SNPdata, PairInfo, X = NULL, categXNames = NULL,
     TestType = "WaldTest", ModelType = "binomial", OR = FALSE)
```

Arguments

`Outcome`	Binary (`1`: event of interest; `0`: reference) or continuous variable.
`SNPdata`	SNP data: All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. The missing values of SNPs need to be defined as "NA", such as snpdata[snpdata=="–"] = NA, where "–" is originally defined symbol of missing value.
`PairInfo`	3 types of PairInfo: (1) 2d-vector: names of the given SNP pair for one-pair analyses. ex: c("SNP1","SNP2") (2) 2d-matrix or 2d-dataframe: names for candidate SNP pair. (3) "all": for pairwise analyses. (4) a list containing two 1d-vectors: for pairwise between these two vectors.
`X`	Covariate(s) to be adjusted in the model (for missing values, keep the field blank), `NULL`=without covariate. Default is `NULL`.
`categXNames`	The variable names of categorical variables, `NULL`=without categorical covariates. Default is `NULL`.
`TestType`	Specify the statistical test type: "WaldTest" = the Wald test; "LRT" = the likelihood ratio test. Default is "WaldTest".
`ModelType`	Model type: "binomial"=logistic regression; "gaussian"=linear regression. Default is "binomial".
`OR`	If `TRUE` print the odds ratios, 95% confidence intervals and corresponding p-values.

Value

Returns a list with the following attributes:

`selectedModel`
: The results of the best model with the lowest BIC value among the 45 models.

`res45Models`
: For one-pair analyses only: Detailed results with all 45 models sorted by BIC (lowest first). Output variables: Var1: SNP 1; Var2: SNP 2 (the pattern/model labels are based on this order); Model: interaction model/pattern; Wald_Chisq: the Wald chi-square value of the interaction term; Wald_p: the Wald p-value of the interaction term; LRT_Chisq: the chi-square value of likelihood ratio test (LRT) for the interaction term; LRT_p: the LRT p-value of the interaction term; BIC: the Bayesian information criterion. The model with the lowest BIC value is preferred.

`OR`
: For the "all" pairs analyses, only results of the best model will show. The group coding, please see suppl. Figures 1-3 in the end of this manual.

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167.

See Also

`parSIPI`, `AA9int`, and `parAAint`

Examples

```
##load data
data(simData)

#### define SNP data
SNPdata = simData[,3:12]

########################
### run SIPI one-pair analyses <Wald test>
########################

##  For a SNP pair of SNP2 and SNP8
res_snp_2v8 = SIPI(simData$D,SNPdata,c("SNP2","SNP8"))

## print out the best model
res_snp_2v8$selectedModel

## list of 45 models for one SNP pair
res_snp_2v8$res45Models
```

```
########################
### run SIPI for a list of multiple SNP pairs <Wald test>
########################

##  For 5 SNP pairs(1 v.s. 6, 2 v.s. 7, 3 v.s. 8, 4 v.s. 9, and 5 v.s. 10)
pairMatrix = c("SNP1","SNP2","SNP3","SNP4","SNP5","SNP6","SNP7","SNP8","SNP9","SNP10")

pairMatrix = matrix(pairMatrix,5)

pairMatrix = as.data.frame(pairMatrix)

res_snp = SIPI(simData$D,SNPdata,pairMatrix)


########################
### run SIPI for all possible combinations of SNP pairs between two vectors <Wald test>
########################
##  For 9 SNP pairs(1 vs. 6, 2 vs. 6, 3 vs. 6, 1 vs. 7, and 2 vs. 7, ..., 3 vs. 8)
vetor_1 = c("SNP1","SNP2","SNP3")
vetor_2 = c("SNP6","SNP7","SNP8")

res_snp = SIPI(simData$D,SNPdata,list(vetor_1,vetor_2))

########################
### run SIPI pairwise analyses <Wald test>
########################

res_all = SIPI(simData$D,SNPdata,"all")


########################
### run SIPI pairwise analyses <Wald test> adjusted for covariates
###  [age(numeric), gender(binary), and group(categorical)]
########################

X1 = simData[,c("age","gender","group")]

res_all_cov = SIPI(simData$D,SNPdata,"all",X1,c("gender","group"))

########################
## export ORs and p-values
########################

try = SIPI(simData$D,SNPdata,"all",OR=TRUE)
write_OR_csv(try$OR,'D:/OR.csv')
write_OR_csv(try$selectedModel,'D:/p.csv')
```

SNPmain *Detect SNP main effect through testing the 3 models*

Description

SNPmain evaluates SNP main effect associated with a binary or continuous outcome through testing three models. The best main effect pattern is the one with the lowest value of the p-value. The details of the three models/patterns are listed in the reference.

Usage

```
SNPmain(Outcome, SNPdata, SNPlist, X = NULL, categXNames = NULL,
        ModelType = "binomial")
```

Arguments

Outcome	Binary (1: event of interest; 0: reference) or continuous variable.
SNPdata	SNP data: All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. An invalid character or blank field are considered to be missing values.
SNPlist	3 types of SNPlist: (1) 1d-vector: names of the given SNP. ex: c("SNP1","SNP2","SNP3") (2) "all": for all SNPs.
X	Covariate(s) to be adjusted in the model (for missing values, keep the field blank), NULL=without covariate. Default is NULL.
categXNames	The variable names of categorical variables, NULL=without categorical covariates. Default is NULL.
ModelType	Model type: "binomial"=logistic regression; "gaussian"=linear regression. Default is "binomial".

Value

Returns the results of the best model with the lowest p-value among the 3 models. Output variables includes SNP (SNP name), Model (main effect model/pattern0), main.effect (coefficeint), p-value, OR (odds ratio), CI_2.5% and CI_97.5% (95 % confidence interval).

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167.

Lin HY, Huang PY, Chen DT, Tung HY, Sellers TA, Pow-Sang J, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Neal DE, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, consortium P, Park JY. AA9int: SNP Interaction Pattern Search Using Non-Hierarchical Additive Model Set. Bioinformatics. 2018. doi: 10.1093/bioinformatics/bty461. PubMed PMID: 29878078.

See Also

SIPI

Examples

```
##load data
data(simData)

#### define SNP data
SNPdata = simData[,3:12]
Outcome = simData$D

########################
### run SNPmain analyses <Wald test>
########################

##  For a list of SNPs (SNP1, SNP2, and SNP3)
res_snp_123 = SNPmain(Outcome,SNPdata,c("SNP1","SNP2","SNP3"))

##  For all SNPs
res_snp_all = SNPmain(Outcome,SNPdata,"all")

########################
### run SNPmain analyses adjusted for covariates
###  [age(numeric), gender (binary), and group (categorical)]
########################

X1 = simData[,c("age","gender","group")]
res_all_cov = SNPmain(Outcome,SNPdata,"all",X1,c("gender","group"))
```

SNPxE *Detect SNP-Env interactions through testing the 27 models*

Description

SNPxE evaluates SNP-environment interactions associated with a binary or continuous outcome through testing 27 interaction models. The details of the nine models/patterns are listed in the reference.

Usage

```
SNPxE(Outcome, SNPdata, Env = NULL, Envtype='ord',Envreference=NULL,
      X = NULL, categXNames = NULL,
      ModelType = "binomial", SelectCriteria = "pvalue", OR = FALSE)
```

Arguments

Outcome	Binary (1: event of interest; 0: reference) or continuous variable.

SNPdata	SNP data: All SNP variables should have a character variable attribute and contain two of four letters (C, T, A, and G). No other letters or numbers should be used. The missing values of SNPs need to be defined as "NA", such as snpdata[snpdata=="–"] = NA, where "–" is originally defined symbol of missing value.
Env	A environment variable with an ordinal or categorical feature (such as negative/positive, low/medium/high or levels).
EnvType	ord: ordinal Env feature; categ: categorical Env feature.
Envreference	The reference level of categorical feature.
X	Covariate(s) to be adjusted in the model (for missing values, keep the field blank), NULL=without covariate. Default is NULL.
categXNames	The variable names of categorical variables, NULL=without categorical covariates. Default is NULL.
ModelType	Model type: "binomial"=logistic regression; "gaussian"=linear regression. Default is "binomial".
SelectCriteria	The Criteria of model selection. SelectCriteria: "pvalue"=p-value, "bic"=Bayesian information criterion.
OR	If TRUE print the odds ratios, 95% confidence intervals and corresponding p-values.

Value

Returns the results of the best model with the lowest value of SelectCriteria among the 27 models. Output variables includes SNP (SNP name), Model (interaction model/pattern), Coef (coefficeint), p-value, OR (odds ratio), CI_2.5% and CI_97.5% (95 % confidence interval).

Author(s)

Hui-Yi Lin and Po-Yu Huang

References

Lin HY, Chen DT, Huang PY, Liu YH, Ochoa A, Zabaleta J, Mercante DE, Fang Z, Sellers TA, Pow-Sang JM, Cheng CH, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, Consortium P, Park JY. SNP interaction pattern identifier (SIPI): an intensive search for SNP-SNP interaction patterns. Bioinformatics. 2017;33(6):822-33. PubMed PMID: 28039167.

Lin HY, Huang PY, Chen DT, Tung HY, Sellers TA, Pow-Sang J, Eeles R, Easton D, Kote-Jarai Z, Amin Al Olama A, Benlloch S, Muir K, Giles GG, Wiklund F, Gronberg H, Haiman CA, Schleutker J, Nordestgaard BG, Travis RC, Hamdy F, Neal DE, Pashayan N, Khaw KT, Stanford JL, Blot WJ, Thibodeau SN, Maier C, Kibel AS, Cybulski C, Cannon-Albright L, Brenner H, Kaneva R, Batra J, Teixeira MR, Pandha H, Lu YJ, consortium P, Park JY. AA9int: SNP Interaction Pattern Search Using Non-Hierarchical Additive Model Set. Bioinformatics. 2018. doi: 10.1093/bioinformatics/bty461. PubMed PMID: 29878078.

Lin HY, Huang PY, Tseng TS, Park JY. SNPxE: SNP-environment interaction pattern identifier. BMC Bioinformatics. 2021;22(1):425. PubMed PMID: 34493206; PMCID: PMC8425112.

See Also

SIPI

Examples

```
##load data
data(simData2)

#### define SNP data
SNPdata = simData2[,9:13]

#### define Outcome
Outcome = simData2$D

#### define Env data
Env = simData2$env_g2

#########################
### run SNPxE analyses <Wald test>
#########################

##  Using pvalue Criteria
res_snp_env_pvalue = SNPxE(Outcome,SNPdata,Env=Env,
                           ModelType="binomial",SelectCriteria="pvalue")

##  Using bic Criteria
res_snp_env_bic = SNPxE(Outcome,SNPdata,Env=Env,
                        ModelType="binomial",SelectCriteria="bic")

#########################
### run SNPxE analyses with categorical feature  <Wald test>
#########################
Env = simData2$env_g3
res_snp_env_pvalue = SNPxE(Outcome,SNPdata,
                           Env=Env,Envtype='categ',Envreference="2",
                           ModelType="binomial",SelectCriteria="pvalue")

#########################
### run SNPxE analyses adjusted for covariates
###  [cov1(numeric), cov2(numeric), and group(categorical)]
#########################

X1 = simData2[,c("group","cov1","cov2")]
res_snp_env_pvalue_cov = SNPxE(Outcome,SNPdata,Env=Env,
                               X=X1,categXNames=c("group"),
                               ModelType="binomial",SelectCriteria="pvalue")

#########################
## export ORs and p-values
#########################
write.csv(res_snp_env_pvalue$Res_df,'D:/P.csv',row.names=FALSE)
write.csv(res_snp_env_pvalue$Coef_dfl,'D:/OR.csv',row.names=FALSE)
```