Package ‘martini’

October 14, 2021

Type Package

Title GWAS Incorporating Networks

Version 1.12.0

Description martini deals with the low power inherent to GWAS studies by using prior knowledge represented as a network. SNPs are the vertices of the network, and the edges represent biological relationships between them (genomic adjacency, belonging to the same gene, physical interaction between protein products). The network is scanned using SConES, which looks for groups of SNPs maximally associated with the phenotype, that form a close subnetwork.

License GPL-3

LazyData TRUE

Imports igraph (>= 1.0.1), Matrix, methods (>= 3.3.2), Rcpp (>= 0.12.8), snpStats (>= 1.20.0), stats, utils,

Suggests biomaRt (>= 2.34.1), circlize (>= 0.4.11), STRINGdb (>= 2.2.0), httr (>= 1.2.1), IRanges (>= 2.8.2), S4Vectors (>= 0.12.2), memoise (>= 2.0.0), knitr, testthat, readr, rmarkdown

Depends R (>= 4.0)

LinkingTo Rcpp, RcppEigen (>= 0.3.3.5.0)

RoxygenNote 7.1.1

Encoding UTF-8

biocViews Software, GenomeWideAssociation, SNP, GeneticVariability, Genetics, FeatureExtraction, GraphAndNetwork, Network

VignetteBuilder knitr

URL `https://github.com/hclimente/martini`

BugReports `https://github.com/hclimente/martini/issues`

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

git_branch RELEASE_3_13

git_last_commit 21f22dd

git_last_commit_date 2021-06-29

Date/Publication 2021-10-14

Author Hector Climente-Gonzalez [aut, cre]
(<https://orcid.org/0000-0002-3030-7471>),
Chloe-Agathe Azencott [aut] (<https://orcid.org/0000-0003-1003-301X>)

Maintainer Hector Climente-Gonzalez <hector.climente@riken.jp>

R topics documented:

get_GI_network 2
get_GM_network 4
get_GS_network 5
gwas2bed 5
ldweight_edges 6
maxflow 6
mincut_c 7
minigwas 7
minippi 8
minisnpMapping 8
plot_ideogram 9
scones 9
scones.cv 11
scones.cv_ 12
scones_ 13
search_cones 14
sigmod 15
sigmod.cv 16
sigmod.cv_ 18
sigmod_ 18
simulate_causal_snps 19
simulate_phenotype 20
subvert 21
wrap_Xy 22

Index **23**

`get_GI_network` *Get gene-interaction network.*

Description

Creates a network of SNPs where each SNP is connected as in the GM network and, in addition, to all the other SNPs pertaining to any interactor of the gene it is mapped to. Corresponds to the gene-interaction (GI) network described by Azencott et al.

Usage

```
get_GI_network(
  gwas,
  organism = 9606,
  snpMapping = snp2ensembl(gwas, organism),
  ppi = get_gxg("biogrid", organism, flush),
  col_ppi = c("gene1", "gene2"),
  col_genes = c("snp", "gene"),
  flush = FALSE
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`organism`	Tax ID of the studied organism. The default is 9606 (human).
`snpMapping`	A data.frame informing how SNPs map to genes. It contains minimum two columns: SNP id and a gene it maps to. Each row corresponds to one gene-SNP mapping. Unless column names are specified using `col_genes`, involved columns must be named `'snp'` and `'gene'`.
`ppi`	A data.frame describing protein-protein interactions with at least two colums. Gene ids must be the contained in snpMapping. Unless column names are specified using `col_ppi`, involved columns must be named `gene1` and `gene2`.
`col_ppi`	Optional, length-2 character vector with the names of the two columns involving the protein-protein interactions.
`col_genes`	Optional, length-2 character vector with the names of the two columns involving the SNP-gene mapping. The first element is the column of the SNP, and the second is the column of the gene.
`flush`	Remove cached results? Boolean value.

Value

An igraph network of the GI network of the SNPs.

References

Azencott, C. A., Grimm, D., Sugiyama, M., Kawahara, Y., & Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29(13), 171-179. https://doi.org/10.1093/bioinformatics/btt238

Examples

```
get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
```

`get_GM_network` *Get gene membership network.*

Description

Creates a network of SNPs where each SNP is connected as in the GS network and, in addition, to all the other SNPs pertaining to the same gene. Corresponds to the gene membership (GM) network described by Azencott et al.

Usage

```
get_GM_network(
  gwas,
  organism = 9606,
  snpMapping = snp2ensembl(gwas, organism),
  col_genes = c("snp", "gene")
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`organism`	Tax ID of the studied organism. The default is 9606 (human).
`snpMapping`	A data.frame informing how SNPs map to genes. It contains minimum two columns: SNP id and a gene it maps to. Each row corresponds to one gene-SNP mapping. Unless column names are specified using `col_genes`, involved columns must be named `'snp'` and `'gene'`.
`col_genes`	Optional, length-2 character vector with the names of the two columns involving the SNP-gene mapping. The first element is the column of the SNP, and the second is the column of the gene.

Value

An igraph network of the GM network of the SNPs.

References

Azencott, C. A., Grimm, D., Sugiyama, M., Kawahara, Y., & Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29(13), 171-179. https://doi.org/10.1093/bioinformatics/btt238

Examples

```
get_GM_network(minigwas, snpMapping = minisnpMapping)
```

`get_GS_network` *Get genomic sequence network*

Description

Creates a network of SNPs where each SNP is connected to its adjacent SNPs in the genome sequence. Corresponds to the genomic sequence (GS) network described by Azencott et al.

Usage

```
get_GS_network(gwas)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.

Value

An igraph network of the GS network of the SNPs.

References

Azencott, C. A., Grimm, D., Sugiyama, M., Kawahara, Y., & Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29(13), 171-179. `https://doi.org/10.1093/bioinformatics/btt238`

Examples

```
get_GS_network(minigwas)
```

`gwas2bed` *Converts a MAP data.frame to a BED data.frame*

Description

Takes a map file and:

- column 1: Used as the chromosome column in the BED file.
- column 4: Used as start and end in the BED data.frame (as we work with SNPs).

Usage

```
gwas2bed(gwas)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.

Value

A BED data.frame.

`ldweight_edges` *Include linkage disequilibrium information in the network.*

Description

Include linkage disequilibrium information in the SNP network. The weight of the edges will be lower the higher the linkage is.

Usage

```
ldweight_edges(net, ld, method = "inverse")
```

Arguments

`net`	A SNP network.
`ld`	A `dsCMatrix` or `dgCMatrix` containing linkage disequilibrium measures, like the output of `ld`.
`method`	How to incorporate linkage-disequilibrium values into the network.

Value

An copy of net where the edges weighted according to linkage disequilibrium.

Examples

```
ld <- snpStats::ld(minigwas[['genotypes']], depth = 2, stats = "R.squared")
# don't weight edges for which LD cannot be calculated
ld[is.na(ld)] <- 0
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
ldGi <- ldweight_edges(gi, ld)
```

`maxflow` *Maxflow algorithm*

Description

Run the maxflow algorithm.

Usage

```
maxflow(A, As, At)
```

Arguments

`A`	A sparse matrix with the connectivity.
`As`	A vector containing the edges to the source.
`At`	A vector containing the edges to the sink.

Value

A list with vector indicating if the feature was selected and the objective score.

`mincut_c` *Min-cut algorithm*

Description

Run the mincut algorithm.

Usage

```
mincut_c(c, eta, lambda, W)
```

Arguments

`c`	A vector with the association of each SNP with the phenotype.
`eta`	A numeric with the value of the eta parameter.
`lambda`	A numeric with the value of the eta parameter.
`W`	A sparse matrix with the connectivity.

Value

A list with vector indicating if the feature was selected and the objective score.

`minigwas` *Description of the minigwas dataset.*

Description

Small GWAS example.

Format

A list with 3 items:

genotypes Genotype and phenotype information.
fam Simulated network.
map Result of runing `find_cones` with gwas and net.

Examples

```
data(minigwas)

# access different elements
minigwas[["genotypes"]]
minigwas[["map"]]
minigwas[["fam"]]
```

`minippi` *PPIs for the minigwas dataset.*

Description

data.frame describing pairs of proteins that interact for minigwas.

Examples

```
data(minippi)

head(minippi)
```

`minisnpMapping` *Genes for the minigwas dataset.*

Description

data.frame that maps SNPs from minigwas to their gene.

Examples

```
data(minisnpMapping)

head(minisnpMapping)
```

`plot_ideogram` *Ideogram of SConES results.*

Description

Create a circular ideogram of the a network results using the circlize package (Gu et al., 2014).

Usage

```
plot_ideogram(gwas, net, covars = data.frame(), genome = "hg19")
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`net`	An igraph network that connects the SNPs.
`covars`	A data frame with the covariates. It must contain a column 'sample' containing the sample IDs, and an additional columns for each covariate.
`genome`	Abbreviations of the genome to use: hg19 for human (default), mm10 for mouse, etc.

Value

A circular ideogram, including the manhattan plot, and the interactions between the selected SNPs.

References

Gu, Z., Gu, L., Eils, R., Schlesner, M., & Brors, B. (2014). circlize Implements and enhances circular visualization in R. Bioinformatics (Oxford, England), 30(19), 2811-2. `https://doi.org/10.1093/bioinformatics/btu393`

`scones` *Find connected explanatory SNPs*

Description

Finds the SNPs maximally associated with a phenotype while being connected in an underlying network.

Usage

```
scones(
  gwas,
  net,
  eta,
  lambda,
  covars = data.frame(),
  score = c("chi2", "glm", "r2"),
  family = c("binomial", "poisson", "gaussian", "gamma"),
  link = c("logit", "log", "identity", "inverse")
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`net`	An igraph network that connects the SNPs.
`eta`	Value of the eta parameter.
`lambda`	Value of the lambda parameter.
`covars`	A data frame with the covariates. It must contain a column 'sample' containing the sample IDs, and an additional columns for each covariate.
`score`	Association score to measure association between genotype and phenotype. Possible values: chi2 (default), glm.
`family`	A string defining the generalized linear model family. This should match one of "binomial", "poisson", "gaussian" or "gamma". See snp.rhs.tests for details.
`link`	A string defining the link function for the GLM. This should match one of "logit", "log", "identity" or "inverse". See snp.rhs.tests for details.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

References

Azencott, C. A., Grimm, D., Sugiyama, M., Kawahara, Y., & Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29(13), 171-179. `https://doi.org/10.1093/bioinformatics/btt238`

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
scones(minigwas, gi, 10, 1)
```

`scones.cv` *Find connected explanatory SNPs.*

Description

Finds the SNPs maximally associated with a phenotype while being connected in an underlying network. Select the hyperparameters by cross-validation.

Usage

```
scones.cv(
  gwas,
  net,
  covars = data.frame(),
  score = c("chi2", "glm", "r2"),
  criterion = c("stability", "bic", "aic", "aicc", "global_clustering",
    "local_clustering"),
  etas = numeric(),
  lambdas = numeric(),
  family = c("binomial", "poisson", "gaussian", "gamma"),
  link = c("logit", "log", "identity", "inverse")
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`net`	An igraph network that connects the SNPs.
`covars`	A data frame with the covariates. It must contain a column 'sample' containing the sample IDs, and an additional columns for each covariate.
`score`	Association score to measure association between genotype and phenotype. Possible values: chi2 (default), glm.
`criterion`	String with the function to measure the quality of a split.
`etas`	Numeric vector with the etas to explore in the grid search. If ommited, it's automatically created based on the association scores.
`lambdas`	Numeric vector with the lambdas to explore in the grid search. If ommited, it's automatically created based on the association scores.
`family`	A string defining the generalized linear model family. This should match one of "binomial", "poisson", "gaussian" or "gamma". See snp.rhs.tests for details.
`link`	A string defining the link function for the GLM. This should match one of "logit", "log", "identity" or "inverse". See snp.rhs.tests for details.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

References

Azencott, C. A., Grimm, D., Sugiyama, M., Kawahara, Y., & Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29(13), 171-179. https://doi.org/10.1093/bioinformatics/btt238

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
scones.cv(minigwas, gi)
scones.cv(minigwas, gi, score = "glm")
```

`scones.cv_` *Find connected explanatory features*

Description

Finds the features maximally associated with a phenotype while being connected in an underlying network. Select the hyperparameters by cross-validation.

Usage

```
scones.cv_(X, y, featnames, net)
```

Arguments

`X`	n x d design matrix
`y`	Vector of length n with the outcomes
`featnames`	Vector of length d with the feature names
`net`	An igraph network that connects the SNPs.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

Examples

```
X <- as(minigwas[['genotypes']], 'numeric')
X <- X + matrix(rnorm(2500, sd = 0.1), nrow(X), ncol(X))
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
scones.cv_(X, minigwas[['fam']]$affected, minigwas[['map']]$snp, gi)
```

`scones_` *Find connected explanatory features*

Description

Finds the features maximally associated with a phenotype while being connected in an underlying network.

Usage

```
scones_(X, y, featnames, net, eta, lambda)
```

Arguments

`X`	n x d design matrix
`y`	Vector of length n with the outcomes
`featnames`	Vector of length d with the feature names
`net`	An igraph network that connects the SNPs.
`eta`	Value of the eta parameter.
`lambda`	Value of the lambda parameter.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

Examples

```
X <- as(minigwas[['genotypes']], 'numeric')
X <- X + matrix(rnorm(2500, sd = 0.1), nrow(X), ncol(X))
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
scones_(X, minigwas[['fam']]$affected, minigwas[['map']]$snp, gi, 10, 1)
```

`search_cones` *Find connected explanatory SNPs.*

Description

Finds the SNPs maximally associated with a phenotype while being connected in an underlying network (Azencott et al., 2013).

Usage

```
search_cones(
  gwas,
  net,
  encoding = "additive",
  sigmod = FALSE,
  covars = data.frame(),
  associationScore = c("chi2", "glm"),
  modelScore = c("stability", "bic", "aic", "aicc", "global_clustering",
    "local_clustering"),
  etas = numeric(),
  lambdas = numeric()
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`net`	An igraph network that connects the SNPs.
`encoding`	SNP encoding (unused argument).
`sigmod`	Boolean. If TRUE, use the Sigmod variant of SConES, meant to prioritize tightly connected clusters of SNPs.
`covars`	A data frame with the covariates. It must contain a column 'sample' containing the sample IDs, and an additional columns for each covariate.
`associationScore`	Association score to measure association between genotype and phenotype.
`modelScore`	String with the function to measure the quality of a split.
`etas`	Numeric vector with the etas to explore in the grid search. If ommited, it's automatically created based on the association scores.
`lambdas`	Numeric vector with the lambdas to explore in the grid search. If ommited, it's automatically created based on the association scores.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

References

Azencott, C. A., Grimm, D., Sugiyama, M., Kawahara, Y., & Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics, 29(13), 171-179. `https://doi.org/10.1093/bioinformatics/btt238`

Examples

```
## Not run: gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
search_cones(minigwas, gi)
search_cones(minigwas, gi, encoding = "recessive")
search_cones(minigwas, gi, associationScore = "skat")
## End(Not run)
```

`sigmod` *Find connected explanatory SNPs*

Description

Finds the SNPs maximally associated with a phenotype while being connected in an underlying network.

Usage

```
sigmod(
  gwas,
  net,
  eta,
  lambda,
  covars = data.frame(),
  score = c("chi2", "glm", "r2"),
  family = c("binomial", "poisson", "gaussian", "gamma"),
  link = c("logit", "log", "identity", "inverse")
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`net`	An igraph network that connects the SNPs.
`eta`	Value of the eta parameter.
`lambda`	Value of the lambda parameter.
`covars`	A data frame with the covariates. It must contain a column 'sample' containing the sample IDs, and an additional columns for each covariate.
`score`	Association score to measure association between genotype and phenotype. Possible values: chi2 (default), glm.
`family`	A string defining the generalized linear model family. This should match one of "binomial", "poisson", "gaussian" or "gamma". See snp.rhs.tests for details.

`link`	A string defining the link function for the GLM. This should match one of "logit", "log", "identity" or "inverse". See snp.rhs.tests for details.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

References

Liu, Y., Brossard, M., Roqueiro, D., Margaritte-Jeannin, P., Sarnowski, C., Bouzigon, E., Demenais, F. (2017). SigMod: an exact and efficient method to identify a strongly interconnected disease-associated module in a gene network. Bioinformatics, 33(10), 1536–1544. `https://doi.org/10.1093/bioinformatics/btx004`

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
sigmod(minigwas, gi, 10, 1)
```

`sigmod.cv` *Find connected explanatory SNPs.*

Description

Finds the SNPs maximally associated with a phenotype while being connected in an underlying network. Select the hyperparameters by cross-validation.

Usage

```
sigmod.cv(
  gwas,
  net,
  covars = data.frame(),
  score = c("chi2", "glm", "r2"),
  criterion = c("stability", "bic", "aic", "aicc", "global_clustering",
    "local_clustering"),
  etas = numeric(),
  lambdas = numeric(),
  family = c("binomial", "poisson", "gaussian", "gamma"),
  link = c("logit", "log", "identity", "inverse")
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`net`	An igraph network that connects the SNPs.
`covars`	A data frame with the covariates. It must contain a column 'sample' containing the sample IDs, and an additional columns for each covariate.
`score`	Association score to measure association between genotype and phenotype. Possible values: chi2 (default), glm.
`criterion`	String with the function to measure the quality of a split.
`etas`	Numeric vector with the etas to explore in the grid search. If ommited, it's automatically created based on the association scores.
`lambdas`	Numeric vector with the lambdas to explore in the grid search. If ommited, it's automatically created based on the association scores.
`family`	A string defining the generalized linear model family. This should match one of "binomial", "poisson", "gaussian" or "gamma". See snp.rhs.tests for details.
`link`	A string defining the link function for the GLM. This should match one of "logit", "log", "identity" or "inverse". See snp.rhs.tests for details.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

References

Liu, Y., Brossard, M., Roqueiro, D., Margaritte-Jeannin, P., Sarnowski, C., Bouzigon, E., Demenais, F. (2017). SigMod: an exact and efficient method to identify a strongly interconnected disease-associated module in a gene network. Bioinformatics, 33(10), 1536–1544. `https://doi.org/10.1093/bioinformatics/btx004`

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
sigmod.cv(minigwas, gi)
sigmod.cv(minigwas, gi, score = "glm")
```

`sigmod.cv_` *Find connected explanatory features*

Description

Finds the features maximally associated with a phenotype while being connected in an underlying network. Select the hyperparameters by cross-validation.

Usage

```
sigmod.cv_(X, y, featnames, net)
```

Arguments

`X`	n x d design matrix
`y`	Vector of length n with the outcomes
`featnames`	Vector of length d with the feature names
`net`	An igraph network that connects the SNPs.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

Examples

```
X <- as(minigwas[['genotypes']], 'numeric')
X <- X + matrix(rnorm(2500, sd = 0.1), nrow(X), ncol(X))
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
sigmod.cv_(X, minigwas[['fam']]$affected, minigwas[['map']]$snp, gi)
```

`sigmod_` *Find connected explanatory features*

Description

Finds the features maximally associated with a phenotype while being connected in an underlying network.

Usage

```
sigmod_(X, y, featnames, net, eta, lambda)
```

Arguments

`X`	n x d design matrix
`y`	Vector of length n with the outcomes
`featnames`	Vector of length d with the feature names
`net`	An igraph network that connects the SNPs.
`eta`	Value of the eta parameter.
`lambda`	Value of the lambda parameter.

Value

A copy of the `SnpMatrix$map` `data.frame`, with the following additions:

- c: contains the univariate association score for every single SNP.
- selected: logical vector indicating if the SNP was selected by SConES or not.
- module: integer with the number of the module the SNP belongs to.

Examples

```
X <- as(minigwas[['genotypes']], 'numeric')
X <- X + matrix(rnorm(2500, sd = 0.1), nrow(X), ncol(X))
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
sigmod_(X, minigwas[['fam']]$affected, minigwas[['map']]$snp, gi, 10, 1)
```

`simulate_causal_snps` *Simulate causal SNPs*

Description

Selects randomly interconnected genes as causal, then selects a proportion of them as causal.

Usage

```
simulate_causal_snps(net, ngenes = 20, pcausal = 1)
```

Arguments

`net`	An igraph gene-interaction (GI) network that connects the SNPs.
`ngenes`	Number of causal genes.
`pcausal`	Number between 0 and 1, proportion of the SNPs in causal genes that are causal themselves.

Value

A vector with the ids of the simulated causal SNPs.

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
simulate_causal_snps(gi, ngenes=2)
```

`simulate_phenotype` *Simulate phenotype*

Description

Simulates a phenotype from a GWAS experiment and a specified set of causal SNPs. If the data is qualitative, only controls are used.

Usage

```
simulate_phenotype(
  gwas,
  snps,
  h2,
  model = "additive",
  effectSize = rnorm(length(snps)),
  qualitative = FALSE,
  ncases,
  ncontrols,
  prevalence
)
```

Arguments

`gwas`	A SnpMatrix object with the GWAS information.
`snps`	Character vector with the SNP ids of the causal SNPs. Must match SNPs in gwas[["map"]][["snp.name"]].
`h2`	Heritability of the phenotype (between 0 and 1).
`model`	String specifying the genetic model under the phenotype. Accepted values: "additive".
`effectSize`	Numeric vector with the same lenght as the number of causal SNPs. It indicates the effect size of each of the SNPs; if absent, they are sampled fron a normal distribution.
`qualitative`	Bool indicating if the phenotype is qualitative or not (quantitative).
`ncases`	Integer specifying the number of cases to simulate in a qualitative phenotype. Required if qualitative = TRUE.
`ncontrols`	Integer specifying the number of controls to simulate in a qualitative phenotype. Required if qualitative = TRUE.
`prevalence`	Value between 0 and 1 specifying the population prevalence of the disease. Note that ncases cannot be greater than prevalence * number of samples. Required if qualitative = TRUE.

Value

A copy of the GWAS experiment with the new phenotypes in `gwas[["fam"]][["affected"]]`.

References

Inspired from GCTA simulation tool: `http://cnsgenomics.com/software/gcta/Simu.html`.

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
causal <- simulate_causal_snps(gi, ngenes = 2)
simulate_phenotype(minigwas, causal, h2 = 1)
```

`subvert` *Vertices with an attribute*

Description

Returns the nodes matching some condition.

Usage

```
subvert(net, attr, values, affirmative = TRUE)
```

Arguments

`net`	An igraph network.
`attr`	An attribute of the vertices.
`values`	Possible values of `attr`
`affirmative`	Logical. States if a condition must be its affirmation (e.g. all nodes with gene name "X"), or its negation (all nodes not with gene name "X").

Value

The vertices with attribute equal to any of the values in `values`.

Examples

```
gi <- get_GI_network(minigwas, snpMapping = minisnpMapping, ppi = minippi)
martini:::subvert(gi, "gene", "A")
martini:::subvert(gi, "name", c("1A1", "1A3"))
```

`wrap_Xy` *Make pseudo SnpMatrix object*

Description

Wrap design matrix and outcome vector into a pseudo SnpMatrix object.

Usage

```
wrap_Xy(X, y, featnames, net)
```

Arguments

`X`	n x d design matrix
`y`	Vector of length n with the outcomes
`featnames`	Vector of length d with the feature names
`net`	An igraph network that connects the SNPs.

Index

get_GI_network, 2
get_GM_network, 4
get_GS_network, 5
GM, *2*
GS, *4*
gwas2bed, 5

ld, *6*
ldweight_edges, 6

maxflow, 6
mincut_c, 7
minigwas, 7
minippi, 8
minisnpMapping, 8

plot_ideogram, 9

scones, 9
scones.cv, 11
scones.cv_, 12
scones_, 13
search_cones, 14
sigmod, 15
sigmod.cv, 16
sigmod.cv_, 18
sigmod_, 18
simulate_causal_snps, 19
simulate_phenotype, 20
snp.rhs.tests, *10*, *11*, *15–17*
subvert, 21

wrap_Xy, 22