Package 'IntOMICS'

September 27, 2023

Type Package

Title Integrative analysis of multi-omics data to infer regulatory networks

Version 1.0.0

URL https://github.com/anna-pacinkova/IntOMICS

BugReports https://github.com/anna-pacinkova/IntOMICS/issues

Description IntOMICS is an efficient integrative framework based on Bayesian networks. IntOMICS systematically analyses gene expression (GE), DNA methylation (METH), copy number variation (CNV) and biological prior knowledge (B) to infer regulatory networks. IntOMICS complements the missing biological prior knowledge by so-called empirical biological knowledge (empB), estimated from the available experimental data. An automatically tuned MCMC algorithm (Yang and Rosenthal, 2017) estimates model parameters and the empirical biological knowledge. Conventional MCMC algorithm with additional Markov blanket resampling (MBR) step (Su and Borsuk, 2016) infers resulting regulatory network structure consisting of three types of nodes: GE nodes refer to gene expression levels, CNV nodes refer to associated copy number variations, and METH nodes refer to associated DNA methylation probe(s).

Imports bnlearn, bnstruct, matrixStats, RColorBrewer, bestNormalize, igraph, gplots, stats, utils, graphics, numbers, SummarizedExperiment, ggplot2, ggraph, methods, cowplot, grid, rlang

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

RoxygenNote 7.2.2

LazyData false

Suggests BiocStyle, knitr, rmarkdown, curatedTCGAData, TCGAutils, testthat

biocViews Software, DNAMethylation, GeneExpression, CopyNumberVariation, SystemsBiology, GeneRegulation, Network, Bayesian

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

git_branch RELEASE_3_17

git_last_commit 334c4bc

git_last_commit_date 2023-04-25

Date/Publication 2023-09-27

Author Pacinkova Anna [cre, aut]

Maintainer Pacinkova Anna <ana.pacinkova@gmail.com>

R topics documented:

`IntOMICS-package` *IntOMICS: Integrative analysis of multi-omics data to infer regulatory networks*

Description

IntOMICS is an efficient integrative framework based on Bayesian networks. IntOMICS systematically analyses gene expression (GE), DNA methylation (METH), copy number variation (CNV) and biological prior knowledge (B) to infer regulatory networks. IntOMICS complements the missing biological prior knowledge by so-called empirical biological knowledge (empB), estimated from the available experimental data. An automatically tuned MCMC algorithm (Yang and Rosenthal, 2017) estimates model parameters and the empirical biological knowledge. Conventional MCMC algorithm with additional Markov blanket resampling (MBR) step (Su and Borsuk, 2016) infers resulting regulatory network structure consisting of three types of nodes: GE nodes refer to gene expression levels, CNV nodes refer to associated copy number variations, and METH nodes refer to associated DNA methylation probe(s).

Author(s)

Maintainer: Pacinkova Anna <ana.pacinkova@gmail.com>

See Also

Useful links:

- https://github.com/anna-pacinkova/IntOMICS
- Report bugs at https://github.com/anna-pacinkova/IntOMICS/issues

---

`acceptance_check` *Acceptance rate checking*

Description

`acceptance_check` This phase verify if the acceptance is in range of 0.28 and 0.6.

Usage

```
acceptance_check(
  first.adapt.phase_net,
  round_check,
  last_iter_check,
  prob_mbr,
  layers_def,
  parent_set_combinations,
  BGe_score_all_configs_node,
  omics,
  annot
)
```

Arguments

`first.adapt.phase_net`	list output of the first.adapt.phase or source_net_def function.
`round_check`	numeric vector after each round_check iterations for which we calculate the beta acceptance rate.
`last_iter_check`	numeric vector number of the acceptance rate for the past last_iter_check iterations.
`prob_mbr`	numeric vector probability of the MBR step.
`layers_def`	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
`parent_set_combinations`	list of all possible parent set configuration for all nodes available.

`BGe_score_all_configs_node`
: list of nodes BGe score for all possible parent set configurations.

`omics`
: named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.

`annot`
: named list containing the associated methylation probes of given gene.

Value

List of 1 element: first adaption phase result before given acceptance rate

---

`annot` *Genes and associated methylation probes*

Description

A named list containing the associated methylation probes of given gene.

Usage

```
annot
```

Format

A named list with 5 components - each component corresponds to one gene:
each component of the list is a character vector with probe names associated with given gene

Source

https://www.cancer.gov/tcga

`beta_tuning` *Beta tuning accessor*

Description

`beta_tuning` This is accessor function for MCMC_sapling_res-class.

Usage

```
beta_tuning(x)
```

Arguments

`x`
: MCMC_sapling_res-class, output from the bn_module function

Value

Matrix, results from adaptive phases that contains hyperparameter beta tuning

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
beta_tuning(BN_mod_res)}
```

---

bge_node *BGe score for specific node*

Description

bge_node Computes the BGe score of given node using precomputed sets of all possible parents.

Usage

```
bge_node(
  node,
  adjacency_matrix,
  parent_set_combinations,
  BGe_score_all_configs_node
)
```

Arguments

node	character vector with given node name.
adjacency_matrix	adjacency matrix of given network.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.

Value

Numeric vector of length 1: BGe score of given node

bge_score *BGe score*

Description

bge_score Computes the BGe score of given network using precomputed sets of possible parents.

Usage

```
bge_score(
  adjacency_matrix,
  omics,
  layers_def,
  parent_set_combinations,
  BGe_score_all_configs_node
)
```

Arguments

adjacency_matrix	adjacency matrix of given network.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.

Value

Numeric vector of length 1: BGe score of given adjacency matrix

bn_module *#' BN module*

Description

bn_module Performs automatically tuned MCMC sampling from posterior distribution together with conventional MCMC sampling using empirical biological prior matrix to sample network structures from posterior distribution.

Usage

```
bn_module(
  burn_in = 1e+05,
  thin = 500,
  OMICS_mod_res,
  minseglen = 50000,
  len = 5,
  prob_mbr = 0.07
)
```

Arguments

burn_in	numeric vector the minimal length of burn-in period of the MCMC simulation.
thin	numeric vector thinning frequency of the resulting MCMC simulation.
OMICS_mod_res	list output from the omics_module function.
minseglen	numeric vector minimal number of iterations with the c_rms value below the c_rms threshold.
len	numeric vector initial width of the sampling interval for hyperparameter beta.
prob_mbr	numeric vector probability of the MBR step.

Value

Large List of 3 elements: empirical biological matrix, sampling phase result and hyperparameter beta tuning trace

Examples

```
if(interactive()){data("OMICS_mod_res", package="IntOMICS")
BN_mod_res <- bn_module(burn_in = 500,
    thin = 20, OMICS_mod_res = OMICS_mod_res,
    minseglen = 5, len = 5, prob_mbr = 0.07)}
```

BN_mod_res *IntOMICS MCMC simulation result*

Description

The output from IntOMICS::BN_module function. A named list containing results from the MCMC sampling (resulting sample is thinned and converted into corresponding CPDAGs)

Usage

```
BN_mod_res
```

Format

A named list with 3 components:

B_prior_mat_weighted IntOMICS estimated empirical biological knowledge

sampling.phase_res results from the conventional MCMC sampling - two independent simulations

beta_tuning result from the automatically tuned MCMC algorithm

---

borders_def *Color scales*

Description

`borders_def` Determines the color scale for each modality.

Usage

```
borders_def(node_list, layers_def, omics, omics_meth_original)
```

Arguments

`node_list`	character vector indicating the complete set of nodes in the resulting network structure.
`layers_def`	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
`omics`	named list containing the gene expression (possibly copy number variation and methylation data).
`omics_meth_original`	METH matrix containing original beta values Each component of the list is a matrix with samples in rows and features in columns.

Value

List of 5 elements indicating the color scale for each modality

b_prior_mat *biological prior matrix*

Description

'b_prior_mat' creates the biological prior matrix.

Usage

```
b_prior_mat(
  omics,
  PK,
  layers_def,
  TFtargs,
  annot,
  lm_METH,
  r_squared_thres,
  p_val_thres,
  TFBS_belief,
  nonGE_belief,
  woPKGE_belief
)
```

Arguments

omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
PK	data.frame with known interactions.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
TFtargs	matrix containing the direct interactions between TFs (columns) and their targets (rows).
annot	named list containing the associated methylation probes of given gene.
lm_METH	logical asking whether to use linear regression to filter methylation data (default=TRUE).
r_squared_thres	numeric vector to define the R^2 used as a threshold of significance in linear regression if lm_METH=TRUE (default=0.3).
p_val_thres	numeric vector to define the p-value used as a threshold of significance in linear regression if lm_METH=TRUE (default=0.05).
TFBS_belief	numeric vector to define the belief concerning the TF and its target interaction (default=0.75).
nonGE_belief	numeric vector to define the belief concerning interactions of features except GE-GE (default=0.5).

woPKGE_belief	numeric vector to define the belief concerning GE-GE interactions without prior knowledge (default=0.5).

Value

List of 4 elements: prior biological matrix and data preprocessing

---

`B_prior_mat_weighted` *Empirical biological knowledge accessor*

Description

`B_prior_mat_weighted` This is accessor function for MCMC_sapling_res-class.

Usage

```
B_prior_mat_weighted(x)
```

Arguments

x	MCMC_sapling_res-class, output from the bn_module function

Value

Matrix, empirical biological knowledge

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
B_prior_mat_weighted(BN_mod_res)}
```

`CPDAGs_sim1` *CPDAGs from the first simulation accessor*

Description

`CPDAGs_sim1` This is accessor function for MCMC_sapling_res-class.

Usage

```
CPDAGs_sim1(x)
```

Arguments

x	MCMC_sapling_res-class, output from the bn_module function

Value

List, CPDAGs from the first independent MCMC simulation

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
CPDAGs_sim1(BN_mod_res)}
```

`CPDAGs_sim2` *CPDAGs from the second simulation accessor*

Description

`CPDAGs_sim2` This is accessor function for MCMC_sapling_res-class.

Usage

```
CPDAGs_sim2(x)
```

Arguments

`x`	MCMC_sapling_res-class, output from the bn_module function

Value

List, CPDAGs from the second independent MCMC simulation

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
CPDAGs_sim2(BN_mod_res)}
```

`dag_core_score` *BGe score*

Description

`dag_core_score` The log of the BGe score simplified as much as possible. This function is from BiDAG package.

Usage

```
dag_core_score(j, parentnodes, n, param)
```

Arguments

j	character vector a node to be scored
parentnodes	character vector the parents of the node j
n	numeric vector number of nodes in the netwrok
param	an object of class scoreparameters, which includes all necessary information for calculating the BDe/BGe score

Value

Numeric vector of length 1

---

dens_edge_weights *Density plot of edge weights inferred by IntOMICS*

Description

dens_edge_weights Creates density plot of edge weights.

Usage

```
dens_edge_weights(net)
```

Arguments

net	list output from the weighted_net function.

Value

density plot of edge weights

Examples

```
data(list=c("OMICS_mod_res", "BN_mod_res", "gene_annot", "TFtarg_mat",
"PK"), package="IntOMICS")
res_weighted <- edge_weights(mcmc_res = BN_mod_res, burn_in = 10000,
 thin = 500, edge_freq_thres = 0.3)
weighted_net_res <- weighted_net(cpdag_weights = res_weighted,
 gene_annot = gene_annot, PK = PK, OMICS_mod_res = OMICS_mod_res,
 gene_ID = "gene_symbol", TFtargs = TFtarg_mat,
 B_prior_mat_weighted = B_prior_mat_weighted(BN_mod_res))
dens_edge_weights(weighted_net_res)
```

edge_proposal *Markov Chain conventional single edge proposal move*

Description

edge_proposal This function samples a conventional single edge proposal moves (identify those edges that are possible to change in given network structure)

Usage

```
edge_proposal(net, candidates, layers_def, ge_nodes, omics, B_prior_mat)
```

Arguments

net	adajcency matrix of given network.
candidates	numeric vector with IDs of potential edge to be changed.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
ge_nodes	character vector with GE node names
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
B_prior_mat	a biological prior matrix.

Value

List of 6 elements needed to define candidates for conventional single edge proposal move

edge_types *Resulting edge types definition*

Description

edge_types Defines the resulting network structure.

Usage

```
edge_types(
  B_prior_mat_weighted,
  PK = NULL,
  gene_annot,
  edge_list,
  node_list,
  OMICS_mod_res,
  edge_weights,
  TFtargs = NULL
)
```

Arguments

B_prior_mat_weighted	matrix one of the outputs of the bn_module function.
PK	data.frame with known interactions.
gene_annot	data.frame containing the entrez ID and corresponding gene symbol for conversion.
edge_list	matrix indicating the interaction between nodes, the first column indicates the source node, the second column indicates the target node.
node_list	character vector indicating the complete set of nodes in the resulting network structure.
OMICS_mod_res	list output from the omics_module function.
edge_weights	character vector includes either "MCMC_freq" to reflect the edge weights frequency over the final set of network structures or "empB" to reflect the empirical biological knowledge estimated by IntOMICS.
TFtargs	matrix containing the direct interactions between TFs (columns) and their targets (rows).

Value

List of 6 elements needed to plot the final regulatory network edges

---

edge_weights *Edge weights of MCMC simulation*

Description

edge_weights Returns list of edges with corresponding posterior probabilities (possibly filtered low reliable edges).

Usage

```
edge_weights(mcmc_res, burn_in, thin, edge_freq_thres = NULL)
```

Arguments

mcmc_res	MCMC_sapling_res output from the bn_module function.
burn_in	numeric vector the minimal length of burn-in period of the MCMC simulation.
thin	numeric vector thinning frequency of the resulting MCMC simulation.
edge_freq_thres	numerical vector the quantile of all edge weights used to filter the most reliable edges.

Value

data.frame with edges and corresponding edge weights; edge_freq_thres used to filter relevant edges

Examples

```
data("BN_mod_res", package="IntOMICS")
res_weighted <- edge_weights(mcmc_res = BN_mod_res, burn_in = 10000,
       thin = 500, edge_freq_thres = 0.3)
```

---

emp_b_heatmap *Heatmap of empB - B*

Description

emp_b_heatmap plot a heatmap with empB - B values (depicts the difference between prior knowledge and the empirical knowledge)

Usage

```
emp_b_heatmap(mcmc_res, OMICS_mod_res, gene_annot, TFtargs)
```

Arguments

mcmc_res	MCMC_sapling_res output from the bn_module function.
OMICS_mod_res	list output from the omics_module function.
gene_annot	data.frame containing the entrez ID and corresponding gene symbol for conversion.
TFtargs	matrix containing the direct interactions between TFs (columns) and their targets (rows).

Value

Figure heatmap

Examples

```
data(list=c("TFtarg_mat", "gene_annot", "OMICS_mod_res",
"BN_mod_res"), package="IntOMICS")
emp_b_heatmap(mcmc_res = BN_mod_res, OMICS_mod_res = OMICS_mod_res,
    gene_annot = gene_annot, TFtargs = TFtarg_mat)
```

energy_function_node_specific *Node energy function*

Description

energy_function_node_specific For each node returns its energy over all parent set configurations, the empty parent set is included.

Usage

```
energy_function_node_specific(all_parents_config, B_prior_mat, int_node)
```

Arguments

all_parents_config	matrix with all possible parent set configurations (column indicates parents of given int_node).
B_prior_mat	a biological prior matrix.
int_node	character vector with given node name.

Value

Numeric vector of length 1

epsilon *Epsilon*

Description

epsilon This function returns the epsilon value for each variable/node of the network. The sum of the epsilons of all variables/nodes in the network gives us the energy of given network.

Usage

```
epsilon(net, B_prior_mat)
```

Arguments

net	adjacency matrix of given network.
B_prior_mat	a biological prior matrix.

Value

Numeric vector of length 1: epsilon of given adjacency matrix (needed to compute energy of given adjacency matrix)

estimated_beta *Estimated beta accessor*

Description

estimated_beta This is accessor function for MCMC_sapling_res-class.

Usage

```
estimated_beta(x)
```

Arguments

x	MCMC_sapling_res-class, output from the bn_module function

Value

Numeric, trace of root mean square used for c_rms measure

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
estimated_beta(BN_mod_res)}
```

estimated_len *Estimated len accessor*

Description

estimated_len This is accessor function for MCMC_sapling_res-class.

Usage

```
estimated_len(x)
```

Arguments

x	MCMC_sapling_res-class, output from the bn_module function

Value

Numeric, width of the sampling interval for hyperparameter beta

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
estimated_len(BN_mod_res)}
```

fan_in_reverse *Number of reverse edge candidates*

Description

fan_in_reverse Determine the number of edges that can be reversed using the fan-in restriction in the largest layer.

Usage

```
fan_in_reverse(positions, net_layer_max, layers_def)
```

Arguments

positions	character vector indicating the interaction between two nodes (the first string indicates the source node, the second string indicates the target node).
net_layer_max	adjacency matrix of the network containing only GE nodes.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.

Value

Numeric vector of length 1: reverse edge candidates

first.adapt.phase_net *IntOMICS first adaption phase result*

Description

The output from IntOMICS::first_adapt_phase function. A named list containing results from the MCMC sampling of the first adaption phase.

Usage

```
first.adapt.phase_net
```

Format

A named list with 10 components:

source.net initial adjacency matrix

beta.source initial beta value

partition_func_UB_beta_source partition function upper bound

acceptance_saved acceptance ratio

B_prior_mat biological prior matrix

acceptance_beta_saved acceptance ratio of beta value

betas simulated beta values

method_choice_saved MCMC method used to sample network structure

nets simulated networks

energy_all_configs_node energy for all possible parent set configurations

---

`first_adapt_phase` *1st adaption phase*

Description

`first_adapt_phase` 1st adaption phase of the adaptive MCMC: the variance of the proposal distribution is changed to achieve the MC acceptance rate of 0.44.

Usage

```
first_adapt_phase(
  omics,
  B_prior_mat,
  energy_all_configs_node,
  len,
  layers_def,
  prob_mbr,
  BGe_score_all_configs_node,
  parent_set_combinations,
  annot
)
```

Arguments

`omics`	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
`B_prior_mat`	a biological prior matrix.
`energy_all_configs_node`	list of nodes energy for all possible parent set configurations.
`len`	numeric vector initial width of the sampling interval for hyperparameter beta.
`layers_def`	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
`prob_mbr`	numeric vector probability of the MBR step.
`BGe_score_all_configs_node`	list of nodes BGe score for all possible parent set configurations.
`parent_set_combinations`	list of all possible parent set configuration for all nodes available.
`annot`	named list containing the associated methylation probes of given gene.

Value

List of 1 element: first adaption phase result

`gene_annot` *Gene ID conversion table*

Description

A data.frame containing the entrez ID and corresponding gene symbol.

Usage

```
gene_annot
```

Format

A data.frame with 8 rows and 2 variables:

entrezID Entrez ID

gene_symbol gene symbol

`ggraph_weighted_net` *Regulatory network plot with edge labels*

Description

`ggraph_weighted_net` Figure of the regulatory network.

Usage

```
ggraph_weighted_net(
  net,
  node_size = 10,
  node_label_size = 4,
  edge_label_size = 4
)
```

Arguments

`net`	list output from the trace_plots function.
`node_size`	numeric node size
`node_label_size`	numeric node label size
`edge_label_size`	numeric edge label size

Value

Figure of weighted network

Examples

```
if(interactive()){data(list=c("OMICS_mod_res", "BN_mod_res", "gene_annot", "TFtarg_mat",
"PK"), package="IntOMICS")
res_weighted <- edge_weights(mcmc_res = BN_mod_res, burn_in = 10000,
 thin = 500, edge_freq_thres = 0.3)
weighted_net_res <- weighted_net(cpdag_weights = res_weighted,
 gene_annot = gene_annot, PK = PK, OMICS_mod_res = OMICS_mod_res,
 gene_ID = "gene_symbol", TFtargs = TFtarg_mat,
 B_prior_mat_weighted = B_prior_mat_weighted(BN_mod_res))
library(ggraph)
ggraph_weighted_net(weighted_net_res)}
```

---

init_net_mcmc *Random initial network*

Description

`init_net_mcmc` This function is used to sample random initial network. The edges are sampled only between GE nodes.

Usage

```
init_net_mcmc(omics, layers_def, B_prior_mat)
```

Arguments

`omics`	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
`layers_def`	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
`B_prior_mat`	a biological prior matrix.

Value

List of 2 elements: random adjacency network and empty network

is_acyclic *Acyclic network identification.*

Description

`is_acyclic` This function is from bnstruct R package. Check if the directed graph is acyclic.

Usage

```
is_acyclic(g)
```

Arguments

`g`	adajcency matrix of given network/graph.

Value

boolean of length 1

layers_def *Layers definition of all omics data*

Description

A data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.

Usage

```
layers_def
```

Format

A data.frame with 3 rows and 3 variables:

omics modality

layer layer ID

fan_in_ge maximal number of parents from given layer to single GE node

legend_custom_ggplot *Node color legend*

Description

legend_custom_ggplot The color scale for each modality.

Usage

```
legend_custom_ggplot(net)
```

Arguments

net	list output from the trace_plots function.

Value

Figure with color key

lm_meth *Linear regression GE~METH*

Description

lm_meth The linear regression model for a dependent variable GE and explanatory variable METH. Returns METH with significant coefficient, R^2 > threshold and R~Gaussian residuals.

Usage

```
lm_meth(ge_mat, meth_mat, gene, meth_probes, r_squared_thres, p_val_thres)
```

Arguments

ge_mat	matrix of gene expression with samples in rows and features in columns.
meth_mat	matrix of DNA methylaton with samples in rows and features in columns.
gene	character vector with given node name.
meth_probes	character vector methylation probes associated with a gene.
r_squared_thres	numeric vector to define the R^2 used as a threshold of significance in linear regression if lm_METH=TRUE (default=0.3).
p_val_thres	numeric vector to define the p-value used as a threshold of significance in linear regression if lm_METH=TRUE (default=0.05).

Value

Character vector with methylation probes

mbr *Markov Blanket Resampling*

Description

mbr This function performs the markov blanket resampling method according to Su and Borsuk, 2016.

Usage

```
mbr(
  source_net_adjacency,
  layers_def,
  omics,
  BGe_score_all_configs_node,
  parent_set_combinations
)
```

Arguments

source_net_adjacency	adajcency matrix of given network.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
parent_set_combinations	list of all possible parent set configuration for all nodes available.

Value

List of 10 elements needed to define adjacency matrix with markov blanket resampling

mc3 *Markov Chain conventional single edge proposal move*

Description

mc3 This function samples a conventional single edge proposal move.

Usage

```
mc3(
  source_net,
  omics,
  layers_def,
  B_prior_mat,
  beta.source,
  partition_func_UB_beta_source,
  parent_set_combinations,
  BGe_score_all_configs_node,
  annot
)
```

Arguments

source_net	list with adjacency matrix and other parameters needed for MCMC simulation.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
B_prior_mat	a biological prior matrix.
beta.source	named list with hyperparameter beta value and other parameters needed for MCMC simulation.
partition_func_UB_beta_source	numeric vector the upper bound of the partition function needed to define the prior distribution of network structure.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
annot	named list containing the associated methylation probes of given gene.

Value

List of 10 elements needed to define adjacency matrix

---

mc3_constant_bge *Markov Chain conventional single edge proposal move with BGe score fixed*

Description

mc3_constant_bge This function samples a conventional single edge proposal move with ficed BGe score.

Usage

```
mc3_constant_bge(
  source_net,
  omics,
  layers_def,
  B_prior_mat,
  beta.source,
  partition_func_UB_beta_source,
  parent_set_combinations,
  BGe_score_all_configs_node,
  annot
)
```

Arguments

source_net	list with adjacency matrix and other parameters needed for MCMC simulation.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
B_prior_mat	a biological prior matrix.
beta.source	named list with hyperparameter beta value and other parameters needed for MCMC simulation.
partition_func_UB_beta_source	numeric vector the upper bound of the partition function needed to define the prior distribution of network structure.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
annot	named list containing the associated methylation probes of given gene.

Value

List of 10 elements needed to define adjacency matrix with conventional single edge move

---

MCMC_sapling_res *The MCMC_sapling_res class*

Description

Container of an MCMC sampling phase results generated by the function bn_module.

Slots

`estimated_beta` Numeric, estimated value of hyperparameter beta

`estimated_len` Numeric, estimated width of the sampling interval for hyperparameter beta

`B_prior_mat_weighted` Empirical biological knowledge matrix, interactions from the biological prior knowledge and TFs-target interactions are constant (unless if "TFBS_belief" is not equal to "woPKGE_belief").

`CPDAGs_sim1` List of CPDAGs from the first independent MCMC simulation (thinned DAGs from the MCMC simulation converted into CPDAGs, ducplicated CPDAGs discarded)

`CPDAGs_sim2` List of CPDAGs from the second independent MCMC simulation (thinned DAGs from the MCMC simulation converted into CPDAGs, ducplicated CPDAGs discarded)

`beta_tuning` Matrix of results from adaptive phases that contains hyperparameter beta tuning

`value` trace of hyperparameter beta

len trace of width of the sampling interval for hyperparameter beta

`rms` Numeric, trace of root mean square used for c_rms measure to evaluate the convergence of MCMC simulation

Examples

```
# A MCMC_sapling_res object created by the bn_module function.
if(interactive()){data("OMICS_mod_res", package="IntOMICS")
BN_mod_res <- bn_module(burn_in = 500,
    thin = 20, OMICS_mod_res = OMICS_mod_res,
    minseglen = 5, len = 5, prob_mbr = 0.07)}
```

---

mcmc_simulation_sampling_phase

Sampling phase

Description

`mcmc_simulation_sampling_phase` This function performs the final sampling of network structures with estimated hyperparameters. It if part of sampling_phase function.

Usage

```
mcmc_simulation_sampling_phase(
  first,
  last,
  sim_init,
  prob_mbr,
  B_prior_mat,
  omics,
  parent_set_combinations,
  BGe_score_all_configs_node,
```

```
  layers_def,
  len,
  thin,
  energy_all_configs_node,
  annot
)
```

Arguments

first	numeric vector iteration to start.
last	numeric vector iteration to stop.
sim_init	list output from the source_net_def function or from two independent simulations from the mcmc_simulation_sampling_phase function.
prob_mbr	numeric vector probability of the MBR step.
B_prior_mat	a biological prior matrix.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
len	numeric vector initial width of the sampling interval for hyperparameter beta.
thin	numeric vector thinning frequency of the resulting MCMC simulation.
energy_all_configs_node	list of nodes energy for all possible parent set configurations.
annot	named list containing the associated methylation probes of given gene.

Value

List of 1 element: sampling phase result before MCMC convergence

---

neighborhood_size *Neighborhood size*

Description

neighborhood_size This function is determines number of network structures that can be reached from the current network structure.

Usage

```
neighborhood_size(net, layers_def, B_prior_mat, omics)
```

Arguments

net	adajcency matrix of given network.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
B_prior_mat	a biological prior matrix.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.

Value

Numeric of length 1: neighborhood size

---

normalise *Arrow of directed edges tuning*

Description

normalise This function is from the ambient package. It is used to determine the position of directed edge arrows.

Usage

```
normalise(x, from = range(x), to = c(0, 1))
```

Arguments

x	numeric vector to be modified.
from	numeric vector range of x.
to	numeric vector range of normalised x.

Value

Numeric vector

omics *Omics data*

Description

A MultiAssayExperiment with names same as in layers_def$omics column containing the gene expression, copy number variation and methylation data.

Usage

```
omics
```

Format

A MultiAssayExperiment with 3 components - each component corresponds to one omics data:
MultiAssayExperiment with variable number of columns

Source

https://www.cancer.gov/tcga

omics_module *omics_module*

Description

omics_module data preprocessing + B_prior_mat definition + partition function upper bound estimation + all possible parent sets per node definition + BGe score computation for all possible parent sets

Usage

```
omics_module(
  omics,
  PK = NULL,
  layers_def,
  TFtargs = NULL,
  annot = NULL,
  lm_METH = TRUE,
  r_squared_thres = 0.3,
  p_val_thres = 0.05,
  TFBS_belief = 0.75,
  nonGE_belief = 0.5,
  woPKGE_belief = 0.5,
  gene_annot
)
```

Arguments

omics	MultiAssayExperiment or named list containing the gene expression (possibly copy number variation and methylation data). If using named list, be aware rownames (samples) match across all objects.
PK	data.frame with known interactions.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
TFtargs	matrix containing the direct interactions between TFs (columns) and their targets (rows).
annot	named list containing the associated methylation probes of given gene.
lm_METH	logical asking whether to use linear regression to filter methylation data (default=TRUE).
r_squared_thres	numeric vector to define the R^2 used as a threshold of significance in linear regression if lm_METH=TRUE (default=0.3).
p_val_thres	numeric vector to define the p-value used as a threshold of significance in linear regression if lm_METH=TRUE (default=0.05).
TFBS_belief	numeric vector to define the belief concerning the TF and its target interaction (default=0.75).
nonGE_belief	numeric vector to define the belief concerning interactions of features except GE-GE (default=0.5).
woPKGE_belief	numeric vector to define the belief concerning GE-GE interactions without prior knowledge (default=0.5).
gene_annot	data.frame containing the entrez ID and corresponding gene symbol for conversion.

Value

List of 6 elements needed to init MCMC simulation

Examples

```
data(list=c("PK", "TFtarg_mat", "annot", "layers_def", "omics",
"gene_annot"), package="IntOMICS")
OMICS_mod_res <- omics_module(omics = omics, PK = PK,
    layers_def = layers_def, TFtargs = TFtarg_mat, annot = annot,
    gene_annot = gene_annot, r_squared_thres = 0.3, lm_METH = TRUE)
```

OMICS_mod_res *preprocessed IntOMICS input data*

Description

The output from IntOMICS::OMICS_module function. A named list containing preprocessed input data.

Usage

```
OMICS_mod_res
```

Format

A named list with 6 components:

pf_UB_BGe_pre output from IntOMICS::pf_UB_est function

B_prior_mat biological prior matrix

annot genes and associated methylation probes

omics a named list containing the gene expression, copy number variation and methylation data

layers_def layers definition of all omics data

omics_meth_original original methylation data

omics_to_list *Convert omics MultiAssayExperiment to list*

Description

`omics_to_list` converts omics MultiAssayExperiment to list

Usage

```
omics_to_list(omics, layers_def, gene_annot)
```

Arguments

`omics`	MultiAssayExperiment containing the gene expression (possibly copy number variation and methylation data).
`layers_def`	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
`gene_annot`	data.frame containing the entrez ID and corresponding gene symbol for conversion.

Value

List of omics modalities

parent_sets_sum_scores_children_x *MBR sum of children scores*

Description

parent_sets_sum_scores_children_x This function determines the sum of BGe scores of given node's children.

Usage

```
parent_sets_sum_scores_children_x(
  parent_set_combinations,
  selected_node,
  children_selected_node,
  child_order,
  dag_tmp_bn,
  new_parent_set,
  source_net_adjacency,
  BGe_score_all_configs_node
)
```

Arguments

parent_set_combinations	list of all possible parent set configuration for all nodes available.
selected_node	character vector with given node name.
children_selected_node	character vector with children of selected_node in given network structure.
child_order	numeric vector random order of children_selected_node.
dag_tmp_bn	object of class bn reflecting given network structure.
new_parent_set	logical asking whether to define new parent set for selected_node.
source_net_adjacency	adajcency matrix of given network.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.

Value

List of 3 elements

parent_sets_sum_scores_x *MBR sum of scores*

Description

parent_sets_sum_scores_x This function determines the sum of BGe scores of given node's parents.

Usage

```
parent_sets_sum_scores_x(
  parent_set_combinations,
  selected_node,
  descendants,
  parent_set,
  BGe_score_all_configs_node
)
```

Arguments

parent_set_combinations	list of all possible parent set configuration for all nodes available.
selected_node	character vector with given node name.
descendants	character vector with descendants of selected_node in given network structure.
parent_set	character vector with parents of selected_node in given network structure.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.

Value

List of 3 elements

pf_ub_est *Partition function upper bound*

Description

pf_ub_est Partition function upper bound estimation with beta = 0. For each node returns energy over all possible parent set configurations and BGe score.

Usage

```
pf_ub_est(omics, B_prior_mat, layers_def, annot)
```

Arguments

omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
B_prior_mat	a biological prior matrix.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
annot	named list containing the associated methylation probes of given gene.

Value

List of 4 elements needed to simulate MCMC sampling

PK *Wnt signalling pathway*

Description

A dataset containing known direct interactions between 7 genes.

Usage

```
PK
```

Format

A data.frame with 6 rows and 3 variables:

src_entrez the parent node

dest_entrez the child node

edge_type the edge from parent node to child node is present or missing

Source

https://www.kegg.jp/entry/map04310

range_01 *Range between 0 and 1*

Description

range_01 This function re-scales a numeric vector so that it ranges between 0 and 1.

Usage

```
range_01(x)
```

Arguments

x	numeric vector.

Value

Numeric vector with normalised values

rms *c_rms trace accessor*

Description

rms This is accessor function for MCMC_sapling_res-class.

Usage

```
rms(x)
```

Arguments

x	MCMC_sapling_res-class, output from the bn_module function

Value

Numeric, trace of root mean square used for c_rms measure

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
rms(BN_mod_res)}
```

sample_chain *Random initial network edge generation*

Description

sample_chain This function is used to sample random initial network. The edges are sampled only between GE nodes.

Usage

```
sample_chain(empty_net, omics_ge)
```

Arguments

empty_net	adjacency matrix of an empty network/graph (all values are 0).
omics_ge	matrix with gene expression data (samples in rows and features in columns).

Value

BN object with conditional probabilities

sampling_phase *Sampling phase*

Description

sampling_phase Now we apply 2 MCMC simulations and check the RMS value. After the burn-in period, we discard the values from the first half of this phase.

Usage

```
sampling_phase(
  second.adapt.phase_net,
  omics,
  layers_def,
  prob_mbr,
  thin,
  minseglen,
  burn_in,
  annot
)
```

Arguments

second.adapt.phase_net	list output of the second.adapt.phase function.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
prob_mbr	numeric vector probability of the MBR step.
thin	numeric vector thinning frequency of the resulting MCMC simulation.
minseglen	numeric vector minimal number of iterations with the c_rms value below the c_rms threshold.
burn_in	numeric vector the minimal length of burn-in period of the MCMC simulation.
annot	named list containing the associated methylation probes of given gene.

Value

List of 2 elements: sampling phase result; RMS used to evaluate MCMC convergence

---

score_parameters_bidag_bge

BGe score parameters

Description

score_parameters_bidag_bge Returns parameters needed for calculation of the BGe score. This function is from BiDAG package.

Usage

```
score_parameters_bidag_bge(n, data, bgepar = list(am = 1, aw = NULL))
```

Arguments

n	numeric number of columns in data matrix.
data	matrix with features in columns and a number of rows equal to the number of samples.
bgepar	list which contains parameters for BGe score computation.

Value

Object of class scoreparameters, which includes all necessary information for calculating the BDe/BGe score

second_adapt_phase *Second adaption phase*

Description

second_adapt_phase This phase identifies the proposal distribution that has a similar covariance structure with the target distribution.

Usage

```
second_adapt_phase(
  transient.phase_net,
  omics,
  layers_def,
  B_prior_mat,
  energy_all_configs_node,
  prob_mbr,
  BGe_score_all_configs_node,
  parent_set_combinations,
  annot,
  woPKGE_belief = 0.5
)
```

Arguments

transient.phase_net	list output of the transient.phase function.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
B_prior_mat	a biological prior matrix.
energy_all_configs_node	list of nodes energy for all possible parent set configurations.
prob_mbr	numeric vector probability of the MBR step.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
annot	named list containing the associated methylation probes of given gene.
woPKGE_belief	numeric vector to define the belief concerning GE-GE interactions without prior knowledge (default=0.5).

Value

List of 1 element: first adaption phase + transient phase + second adaption phase result

---

show,MCMC_sapling_res-method *MCMC_sampling_res-methods*

Description

set show method for MCMC_sampling_res-class objects.

Usage

```
## S4 method for signature 'MCMC_sapling_res'
show(object)
```

Arguments

object	given MCMC_sampling_res-class object

Value

Get summary of the properties of MCMC_sampling_res-class object.

---

source_net_def *Source network for MCMC simulation*

Description

source_net_def This function is used to create the initial network with its features necessary for MCMC simulation.

Usage

```
source_net_def(
  init_net_mcmc.output,
  parent_set_combinations,
  omics,
  BGe_score_all_configs_node,
  B_prior_mat,
  layers_def,
  energy_all_configs_node,
  len
)
```

Arguments

init_net_mcmc.output	list output of the init_net_mcmc function.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
B_prior_mat	a biological prior matrix.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
energy_all_configs_node	list of nodes energy for all possible parent set configurations.
len	numeric vector initial width of the sampling interval for hyperparameter beta.

Value

List of 10 elements needed to define the initial adjacency matrix

---

squared_jumping *Squared jumping of adaptive MCMC algorithm*

Description

squared_jumping Squared jumping of adaptive MCMC algorithm is used to tune the variance of the beta parameter.

Usage

```
squared_jumping(
  second.adapt.phase_net,
  constant,
  fin,
  beta_sd,
  B_prior_mat,
  omics,
  parent_set_combinations,
  BGe_score_all_configs_node,
  layers_def,
  prob_mbr,
  annot
)
```

Arguments

second.adapt.phase_net	list output of the variance_target or squared_jumping function.
constant	numeric vector used to multiply the beta_sd to determine the variance of the distribution of the hyperparameter beta.
fin	numeric vector iteration to stop.
beta_sd	numeric vector used to determine the variance of the distribution of the hyperparameter beta.
B_prior_mat	a biological prior matrix.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
prob_mbr	numeric vector probability of the MBR step.
annot	named list containing the associated methylation probes of given gene.

Value

List of 1 element: second adaptive phase result with stopped MCMC mixing

---

TFtarg_mat *transcription factors and their known targets*

Description

A dataset containing the direct interactions between TFs and their targets.

Usage

```
TFtarg_mat
```

Format

A matrix with 22452 rows and 181 variables:

columns refer to TFs and rows to their targets

Source

https://maayanlab.cloud/Harmonizome/dataset/ENCODE+Transcription+Factor+Targets

trace_plots *Trace plots of MCMC simulation*

Description

trace_plots Create trace plots of MCMC simulation and filter low reliable edges based on the edge_freq_thres parameter.

Usage

```
trace_plots(mcmc_res, burn_in, thin, edge_freq_thres = NULL)
```

Arguments

mcmc_res	MCMC_sapling_res output from the bn_module function.
burn_in	numeric vector the minimal length of burn-in period of the MCMC simulation.
thin	numeric vector thinning frequency of the resulting MCMC simulation.
edge_freq_thres	numerical vector the quantile of all edge weights used to filter the most reliable edges.

Value

MCMC simulation trace plots

Examples

```
if(interactive()){data("BN_mod_res", package="IntOMICS")
res_weighted <- trace_plots(mcmc_res = BN_mod_res, burn_in = 10000,
       thin = 500, edge_freq_thres = 0.3)}
```

transient_phase *transient phase*

Description

transient_phase This phase verify if the chain is moving towards the mode of target distribution.

Usage

```
transient_phase(
  first.adapt.phase_net,
  omics,
  B_prior_mat,
  layers_def,
  energy_all_configs_node,
  prob_mbr,
  BGe_score_all_configs_node,
  parent_set_combinations,
  annot
)
```

Arguments

first.adapt.phase_net	list output of the first.adapt.phase function.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
B_prior_mat	a biological prior matrix.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
energy_all_configs_node	list of nodes energy for all possible parent set configurations.
prob_mbr	numeric vector probability of the MBR step.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
annot	named list containing the associated methylation probes of given gene.

Value

List of 1 element: first adaption phase and transient phase result

---

variance_target *Second adaption phase variance tuning*

Description

variance_target This phase identifies the proposal distribution that has a similar covariance structure with the target distribution. This is part of second_adapt_phase.

Usage

```
variance_target(
  transient.phase_net,
  constant,
  fin,
  B_prior_mat,
  omics,
  parent_set_combinations,
  BGe_score_all_configs_node,
  layers_def,
  prob_mbr,
  annot
)
```

Arguments

transient.phase_net	list output of the variance_target or transient.phase function.
constant	numeric vector used to multiply the beta_sd to determine the variance of the distribution of the hyperparameter beta.
fin	numeric vector iteration to stop.
B_prior_mat	a biological prior matrix.
omics	named list containing the gene expression (possibly copy number variation and methylation data). Each component of the list is a matrix with samples in rows and features in columns.
parent_set_combinations	list of all possible parent set configuration for all nodes available.
BGe_score_all_configs_node	list of nodes BGe score for all possible parent set configurations.
layers_def	data.frame containing the modality ID, corresponding layer in BN and maximal number of parents from given layer to GE nodes.
prob_mbr	numeric vector probability of the MBR step.
annot	named list containing the associated methylation probes of given gene.

Value

Large List of 3 elements: second adaptive phase result with possible MCMC mixing; acceptance rate of hyperparameter beta; SD of hyperparameter beta

weighted_net *Resulting network definition*

Description

weighted_net Defines the resulting network structure and determines the color scale for each modality.

Usage

```
weighted_net(
  cpdag_weights,
  gene_annot,
  PK = NULL,
  OMICS_mod_res,
  edge_weights = "MCMC_freq",
  gene_ID,
  TFtargs = NULL,
  B_prior_mat_weighted
)
```

Arguments

cpdag_weights	data.frame output from the edge_weights function.
gene_annot	data.frame containing the entrez ID and corresponding gene symbol for conversion.
PK	data.frame with known interactions.
OMICS_mod_res	list output from the omics_module function.
edge_weights	character vector includes either "MCMC_freq" to reflect the edge weights frequency over the final set of network structures or "empB" to reflect the empirical biological knowledge estimated by IntOMICS.
gene_ID	character vector includes either "gene_symbol" or "entrezID" to reflect gene identifiers used in the final figure.
TFtargs	matrix containing the direct interactions between TFs (columns) and their targets (rows).
B_prior_mat_weighted	matrix one of the outputs of the bn_module function.

Value

List of 7 elements needed to plot the final regulatory network

Examples

```
data(list=c("OMICS_mod_res", "BN_mod_res", "gene_annot", "TFtarg_mat",
"PK"), package="IntOMICS")
res_weighted <- edge_weights(mcmc_res = BN_mod_res, burn_in = 10000,
 thin = 500, edge_freq_thres = 0.3)
weighted_net_res <- weighted_net(cpdag_weights = res_weighted,
 gene_annot = gene_annot, PK = PK, OMICS_mod_res = OMICS_mod_res,
 gene_ID = "gene_symbol", TFtargs = TFtarg_mat,
 B_prior_mat_weighted = B_prior_mat_weighted(BN_mod_res))
```

Index