Package 'CARNIVAL'

December 11, 2018

Type Package

Title A CAusal Reasoning tool for Network Identification (from gene expression data) using Integer VALue programming.

Version 0.9.1

Author Anika Liu, Enio Gjerga, Panuwat Trairatphisan

Maintainer Panuwat Trairatphisan <panuwat.trairatphisan@outlook.com>

Description An upgraded causal reasoning tool from Melas et al. in R with updated assignments of TFs' weights from PROGENy scores. Cplex parameters can be freely adjusted and multiple solutions from cplex can be obtained and aggregated

License GPL-3

Encoding UTF-8

LazyData true

Imports doParallel, igraph, tidyr, dplyr, readr, tidyverse, viper

Suggests knitr, rmarkdown,

VignetteBuilder knitr

RoxygenNote 6.1.1

BiocViews

NeedsCompilation no

R topics documented:

`AddPerturbationNode` `AddPerturbationNode`

Description

Introduces a perturbation node connecting periphery nodes without a target in the prior knowledge network.

Usage

```
AddPerturbationNode(network)
```

Arguments

`network`	The original prior knowledge network

Value

A list of updated network with perturbation nodes and re-assigned list of input to include only the interactions from the perturbation node.

all_constraints_wLoop all_constraints_wLoop

Description

all_constraints_wLoop

Usage

```
all_constraints_wLoop(c0 = c0, c1 = c1, c2 = c2, c3 = c3,
  c4 = c4, c5 = c5, c6 = c6, c7 = c7, c8 = c8, c9 = c9)
```

Arguments

c0	Constraints number 0
c1	Constraints number 1
c2	Constraints number 2
c3	Constraints number 3
c4	Constraints number 4
c5	Constraints number 5
c6	Constraints number 6
c7	Constraints number 7
c8	Constraints number 8
c9	Constraints number 9

Value

The list of all LP constraints with removed failed-to-write constraints that returns NaN

assignPROGENyScores assignPROGENyScores

Description

assignPROGENyScores

Usage

```
assignPROGENyScores(progeny = progeny, progenyMembers = progenyMembers,
  id = "uniprot")
```

Arguments

progeny	Contains the progeny scores as obtained from runPROGENy.
progenyMembers	Contains the list of members for each PROGENy pathway.
id	Contains the members identifiers (default: uniprot).

Value

This function is used to account for the PROGENy scores in the objective function.

buildDataMatrix buildDataMatrix

Description

buildDataMatrix

Usage

```
buildDataMatrix(data = data, pknList = pknList, inputs = inputs)
```

Arguments

data	Contains the measured data.
pknList	Contains the background network which serves as a prior knowledge and which we train.
inputs	Contains the list of targets as inputs.

Value

This function returns the data matrix containing the data for running CARNIVAL and a set of identifiers for Targets, Measured and Un-measured nodes.

create_variables_all create_variables_all

Description

create_variables_all

Usage

```
create_variables_all(pknList = pknList, dataMatrix = dataMatrix)
```

Arguments

pknList	Contains the background network which serves as a prior knowledge and which we train.
dataMatrix	Contains the matrix which stores the data for running CARNIVAL and a set of identifiers for Targets, Measured and Un-measured nodes.

Value

This function returns the identifiers of all the variables used in the ILP formulation together with an explanation about the meaning of each of them. Also it returns a list of useful identifiers.

`exportResult` `exportResult`

Description

Extract and export the optimisation results from the cplex solution file (XML) as files and variables for further plotting functions

Usage

```
exportResult(cplexSolutionFileName = cplexSolutionFileName,
  variables = variables, conditionIDX = conditionIDX,
  pknList = pknList, dir_name = dir_name, inputs = inputs,
  measurements = measurements, Export_all = Export_all,
  writeIndividualResults = F)
```

Arguments

`cplexSolutionFileName`	Path to the cplex solution file (XML)
`variables`	The list of mapping indices of LP variables
`conditionIDX`	The number of experimental condition to be processed
`pknList`	The provided prior knowledge network
`dir_name`	The name of directory to store results
`inputs`	The list of known or potential target of perturbation
`measurements`	The discretised observations (here transcription factor activities) of values [-1,0,1]
`Export_all`	An option to define whether all detailed mapppd LP variables will be written as individual files
`writeIndividualResults`	An option to define whether the results of individual solutions will be written; if FALSE, only the global combined solution will be written

Value

Output files of ILP solutions and a list of networks and node activities to be written into figures

---

`files2res` `files2res`

Description

Read in a list of exported/written network and node activities files into a variable for plotting

Usage

```
files2res(counterlist)
```

Arguments

counterlist	Indices of model solutions to be read-in

Value

A list of variables for plotting functions

---

generate_measfile generate_measfile

Description

This function generates the measurement file for each condition which is required as one of the input file

Usage

```
generate_measfile(measurements, topnumber = NULL,
  write2folder = "./measurements")
```

Arguments

measurements	A dataframe describing the measurements (e.g. inferred TF activities). Columns/colnames should correspond to perturbations, row/rownames to measurements.
topnumber	If given, only the top number of measurements will be written out.
write2folder	Path to outputfolder

Value

A n*m dataframe describing the inferred TF activities, where n is the number of Tfs and m the number of conditions.

---

genesymbol2uniprot genesymbol2uniprot

Description

This function converts the gene symbol rownames of a dataframe to uniprot rownames. In case one gene symbol maps to two uniprot IDs, the row will be duplicated.

Usage

```
genesymbol2uniprot(df, map, geneID = 1, uniprotID = 2)
```

Arguments

df	A vector of genes.
map	A dataframe with a uniprot ID column and a genesymbol column.
geneID	Column index of the gene symbol column in map.
uniprotID	Column index of the uniprot column in map.

Value

A vector of genes or uniprot IDs.

`load_CARNIVAL_examples` `load_CARNIVAL_examples`

Description

`load_CARNIVAL_examples`

Usage

```
load_CARNIVAL_examples(CARNIVAL_example)
```

Arguments

`CARNIVAL_example`	The number of CARNIVAL example to be loaded; 1 = Toy model, 2 = EGF-SBVimprover,3 = APAP-TGGs

Value

The input variables for CARNIVAL including network, measurement, input target +/- pathways scores

---

`runPROGENy` `runPROGENy` *This function is designed to compute progeny pathway scores and assess there significance using a gene sampling based permutation strategy, for a series of experimental samples/contrasts.*

Description

`runPROGENy` This function is designed to compute progeny pathway scores and assess there significance using a gene sampling based permutation strategy, for a series of experimental samples/contrasts.

Usage

```
runPROGENy(df, weight_matrix, k = 10000, z_scores = T,
  get_nulldist = F)
```

Arguments

df	A data.frame of n*m+1 dimension, where n is the number of omic features to be considered and m is the number of samples/contrasts. The first column should be the identifiers of the omic features. These identifiers must be coherent with the identifers of the weight matrix.
weight_matrix	A progeny coeficient matrix. the first column should be the identifiers of the omic features, and should be coherent with the identifiers provided in df.
k	The number of permutations to be preformed to generate the null-distribution used to estimate significance of progeny scores. Default value is 10000.
z_scores	If true, provides z-scores. If false, provides significance scores (1-pval).
get_nulldist	If get_nulldist is true, then the function will return the null model dataframe that was used.

Value

This function returns a list of two elements. The first element is a dataframe of p*m+1 dimensions, where p is the number of progeny pathways, and m is the number of samples/contrasts. Each cell represent the significance of a progeny pathway score for one sample/contrast. The signifcance ranges between -1 and 1. The significance is equal to x*2-1, x being the quantile of the progeny pathway score with respect to the null distribution. Thus, this significance can be interpreted as the equivalent of 1-p.value (two sided test over an empirical distribution) with the sign indicating the direction of the regulation. The sceond element is the null distribution list (a null distribution is generated for each sample/contrast).

`run_dorothea` `run_dorothea`

Description

This function infers TF activities based on the DoRothEA regulon and the viper function. The regulon can be filtered by confidence level (A-E).

Usage

```
run_dorothea(df, regulon, confidence_level = c("A", "B", "C"),
  write2file = NULL)
```

Arguments

df	A n*m dataframe describing the contrast t-statistics, where n is the number of genes and m the number of conditions.
regulon	DoRothEA regulon list
confidence_level	A vector describing which confidence levels of the DoRothEA regulon to use
write2file	Path to outputfile for all TF_activities if desired

Value

A n*m dataframe describing the inferred TF activities, where n is the number of Tfs and m the number of conditions.

`Uniprot2GeneSymbol` `Uniprot2GeneSymbol`

Description

Conversion of Uniprot ID (e.g. from Omnipath) to official gene symbol in the plotting step

Usage

```
Uniprot2GeneSymbol(res)
```

Arguments

`res`	A list of variables in Uniprot ID (network and node activities)

Value

A list of mapped variables from Uniprot to official gene symbol for a better readibility

`WriteDOTfig` `WriteDOTfig`

Description

This function takes results from ILP optimisation and write out a figure in DOT format

Usage

```
WriteDOTfig(res, idxModel = 0, dir_name, inputs, measurements,
  UP2GS = F)
```

Arguments

`res`	A list containing optimised network structures and node activities from ILP optimisation
`idxModel`	(optional) The index or indices of optimised model to be plotted (Default idxModel=0; plot the combined and average network)
`dir_name`	The directory name that stores CARNIVAL results (as a subfolder of the main "results" folder)
`inputs`	A named vector of inputs and their node states
`measurements`	A named vector of measurements and their state values
`UP2GS`	(T/F) A parameter defining whether to convert the input format UniprotID to Gene Symbol

Value

DOT figures stored in the directory "results/dir_name" with the prefix "ActivityNetwork_model"

writeLPFile writeLPFile

Description

Write a list of linear programming (LP) constraints into a file while will be read by interactive cplex solver to perform network optimisation.

Usage

```
writeLPFile(data = data, pknList = pknList, inputs = inputs,
  alphaWeight = 1, betaWeight = 0.2, scores = scores, mipGAP = 0.1,
  poolrelGAP = 0.01, limitPop = 100, poolCap = 100,
  poolIntensity = 0, poolReplace = 2, timelimit = 1800,
  measWeights = NULL, repIndex, condition = "")
```

Arguments

data	The discretised observations (here transcription factor activities) of values [-1,0,1]
pknList	The prior knowledge network
inputs	The list of known or potential target of perturbation
alphaWeight	The weight for mismatch penalty between discretised observations and predicted model states
betaWeight	The weight for network size (node) penalty
scores	The continuous pathway scores from PROGENy
mipGAP	The minimal integer programming percentage gap to be accepted as a solution
poolrelGAP	The allowed relative percentage gap between the best solution and the equivalent solutions in the solution pool
limitPop	The number of allowed solutions to be populated
poolCap	The number of solutions to be kept in the pool of solution
poolIntensity	The intensity of the search in solution space
poolReplace	The replacement strategy of the solutions in the solution pool
timelimit	The allowed amount of time (in seconds) for the optimisation
measWeights	The countinous weight of observations (here transcription factor activities) - to replace the default alphaWeight if assigned; Note: take only positive values!
repIndex	The indexing of optimisation - useful in case more than one experiment is performed
condition	The free variable which could be assigned for additional study e.g. to vary the efect of betaWeight in a loop

Value

An integer programming file containing the description of ILP optimisation problem and a cplex command file to communicate with the interactive version of cplex solver

write_binaries write_binaries

Description

write_binaries

Usage

```
write_binaries(variables = variables)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.

Value

This code writes the list of binary variables (xp, xm, up & um).

write_boundaries write_boundaries

Description

write_boundaries

Usage

```
write_boundaries(variables = variables, oF = oF)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
oF	Is the objective function of the formulation.

Value

This code writes the boundaries of each variable.

`write_constraints_1_all` `write_constraints_1_all`

Description

`write_constraints_1_all`

Usage

```
write_constraints_1_all(variables = variables)
```

Arguments

`variables`	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.

Value

This code writes the list of constraints (1) of the ILP problem for all the conditions.

`write_constraints_2_all` `write_constraints_2_all`

Description

`write_constraints_2_all`

Usage

```
write_constraints_2_all(variables = variables)
```

Arguments

`variables`	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.

Value

This code writes the list of constraints (2) of the ILP problem for all the conditions.

write_constraints_3_all
write_constraints_3_all

Description

write_constraints_3_all

Usage

```
write_constraints_3_all(variables = variables)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.

Value

This code writes the list of constraints (3) of the ILP problem for all the conditions.

write_constraints_4_all
write_constraints_4_all

Description

write_constraints_4_all

Usage

```
write_constraints_4_all(variables = variables)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.

Value

This code writes the list of constraints (4) of the ILP problem for all the conditions.

write_constraints_5_all write_constraints_5_all

Description

write_constraints_5_all

Usage

```
write_constraints_5_all(variables = variables)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.

Value

This code writes the list of constraints (5) of the ILP problem for all the conditions.

write_constraints_6 write_constraints_6

Description

write_constraints_6

Usage

```
write_constraints_6(variables = variables, dataMatrix = dataMatrix,
  inputs = inputs)
```

Arguments

variables	The list of mapping indices of LP variables.variables Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
dataMatrix	Contains the matrix which stores the data for running CARNIVAL and a set of identifiers for Targets, Measured and Un-measured nodes.
inputs	Contains the list of targets as inputs.

Value

This code writes the list of constraints (6) of the ILP problem for all the conditions.

write_constraints_7 write_constraints_7

Description

write_constraints_7

Usage

```
write_constraints_7(variables = variables, dataMatrix = dataMatrix,
  inputs = inputs)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
dataMatrix	Contains the matrix which stores the data for running CARNIVAL and a set of identifiers for Targets, Measured and Un-measured nodes.
inputs	Contains the list of targets as inputs.

Value

This code writes the list of constraints (7) of the ILP problem for all the conditions.

write_constraints_8 write_constraints_8

Description

write_constraints_8

Usage

```
write_constraints_8(variables = variables, inputs = inputs,
  pknList = pknList)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
inputs	Contains the list of targets as inputs.
pknList	Contains the background network which serves as a prior knowledge and which we train.

Value

This code writes the list of constraints (8) of the ILP problem for all the conditions.

write_constraints_objFunction_all write_constraints_objFunction_all

Description

write_constraints_objFunction_all

Usage

```
write_constraints_objFunction_all(variables = variables,
  dataMatrix = dataMatrix)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indeces.
dataMatrix	Contains the matrix which stores the data for running CARNIVAL and a set of identifiers for Targets, Measured and Un-measured nodes.

Value

This function returns the list of constraints associated with the 'Absolute Difference' variables and which measure the mis-fit between inferred and measured data.

write_generals write_generals

Description

write_generals

Usage

```
write_generals(variables = variables, oF = oF)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
oF	Is the objective function of the formulation.

Value

This code writes all the variables.

write_loop_constraints
write_loop_constraints

Description

write_loop_constraints

Usage

```
write_loop_constraints(variables = variables, pknList = pknList,
  inputs = inputs)
```

Arguments

variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
pknList	Contains the background network which serves as a prior knowledge and which we train.
inputs	Contains the list of targets as inputs.

Value

This function writes the constraints preventing self-activation of nodes in the network due to positive feedback loops.

write_objective_function
write_objective_function

Description

write_objective_function

Usage

```
write_objective_function(dataMatrix = dataMatrix,
  variables = variables, alphaWeight = alphaWeight,
  betaWeight = betaWeight, scores = scores,
  measWeights = measWeights, conditionIDX = conditionIDX)
```

Arguments

dataMatrix	Contains the matrix which stores the information for each node in the PKN, i.e. acivity of the nodes which are measured,at each condition.
variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indices.
alphaWeight	The weightning factor of the measurement.

betaWeight	The weightning factor of the network size.
scores	The provided PROGENy scores.
measWeights	A weightning factor for the measurements.
conditionIDX	The index of the current condition being considered.

Value

This code writes the objective function of the ILP problem for one specific condition.

---

write_objective_function_all

write_objective_function_all

Description

write_objective_function_all

Usage

```
write_objective_function_all(dataMatrix = dataMatrix,
  variables = variables, alphaWeight = alphaWeight,
  betaWeight = betaWeight, scores = scores,
  measWeights = measWeights, conditionIDX = conditionIDX)
```

Arguments

dataMatrix	Contains the matrix which stores the information for each node in the PKN, i.e. acivity of the nodes which are measured,at each condition.
variables	Contains the list of variables as used to formulate the ILP problem, explanations for each variable and a list of useful indeces.
alphaWeight	The weightning factor of the measurement.
betaWeight	The weightning factor of the network size.
scores	The provided PROGENy scores.
measWeights	A weightning factor for the measurements.
conditionIDX	The index of the current condition being considered.

Value

This code writes the objective function of the ILP problem for all the conditions.

Index