Package ‘coglasso’

July 22, 2025

Type Package

Title Collaborative Graphical Lasso - Multi-Omics Network Reconstruction

Version 1.0.2

Description Reconstruct networks from multi-omics data sets with the collaborative graphical lasso (coglasso) algorithm described in Albanese, A., Kohlen, W., and Behrouzi, P. (2024) <doi:10.48550/arXiv.2403.18602>. Build multiple networks using the coglasso() function, select the best one with stars_coglasso().

URL `https://github.com/DrQuestion/coglasso`, `https://drquestion.github.io/coglasso/`

BugReports `https://github.com/DrQuestion/coglasso/issues`

License GPL (>= 2)

Imports Matrix, Rcpp (>= 1.0.11), stats, utils

LinkingTo Rcpp, RcppEigen

Depends R (>= 2.10)

LazyData true

Encoding UTF-8

RoxygenNote 7.2.3

Suggests igraph, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Alessio Albanese [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0003-1783-5613`>), Pariya Behrouzi [aut] (ORCID: <`https://orcid.org/0000-0001-6762-5433`>)

Maintainer Alessio Albanese <`alessio.albanese@wur.nl`>

Repository CRAN

Date/Publication 2024-04-03 20:02:59 UTC

Contents

coglasso *Estimate networks from a multi-omics data set*

Description

coglasso() estimates multiple multi-omics networks with the algorithm *collaborative graphical lasso*, one for each combination of input values for the hyperparameters λ_w, λ_b and c.

Usage

```
coglasso(
  data,
  pX,
  lambda_w = NULL,
  lambda_b = NULL,
  c = NULL,
  nlambda_w = NULL,
  nlambda_b = NULL,
  nc = NULL,
  lambda_w_max = NULL,
  lambda_b_max = NULL,
  c_max = NULL,
  lambda_w_min_ratio = NULL,
  lambda_b_min_ratio = NULL,
  c_min_ratio = NULL,
  cov_output = FALSE,
  verbose = TRUE
)
```

Arguments

data	The input multi-omics data set. Rows should be samples, columns should be variables. Variables should be grouped by their assay (i.e. transcripts first, then metabolites). data is a required parameter.
pX	The number of variables of the first data set (i.e. the number of transcripts). pX is a required parameter.
lambda_w	A vector of values for the parameter λ_w, the penalization parameter for the "within" interactions. Overrides nlambda_w.
lambda_b	A vector of values for the parameter λ_b, the penalization parameter for the "between" interactions. Overrides nlambda_b.

`c` A vector of values for the parameter c, the weight given to collaboration. Overrides `nc`.

`nlambda_w` The number of requested λ_w parameters to explore. A sequence of size `nlambda_w` of λ_w parameters will be generated. Defaults to 8. Ignored when `lambda_w` is set by the user.

`nlambda_b` The number of requested λ_b parameters to explore. A sequence of size `nlambda_b` of λ_b parameters will be generated. Defaults to 8. Ignored when `lambda_b` is set by the user.

`nc` The number of requested c parameters to explore. A sequence of size `nc` of c parameters will be generated. Defaults to 8. Ignored when `c` is set by the user.

`lambda_w_max` The greatest generated λ_w. By default it is computed with a data-driven approach. Ignored when `lambda_w` is set by the user.

`lambda_b_max` The greatest generated λ_b. By default it is computed with a data-driven approach. Ignored when `lambda_b` is set by the user.

`c_max` The greatest generated c. Defaults to 10. Ignored when `c` is set by the user.

`lambda_w_min_ratio`
The ratio of the smallest generated λ_w over the greatest generated λ_w. Defaults to 0.1. Ignored when `lambda_w` is set by the user.

`lambda_b_min_ratio`
The ratio of the smallest generated λ_b over the greatest generated λ_b. Defaults to 0.1. Ignored when `lambda_b` is set by the user.

`c_min_ratio` The ratio of the smallest generated c over the greatest generated c. Defaults to 0.1. Ignored when `c` is set by the user.

`cov_output` Add the estimated variance-covariance matrix to the output.

`verbose` Print information regarding current `coglasso` run on the console.

Value

`coglasso()` returns a list containing several elements:

- `loglik` is a numerical vector containing the log likelihoods of all the estimated networks.
- `density` is a numerical vector containing a measure of the density of all the estimated networks.
- `df` is an integer vector containing the degrees of freedom of all the estimated networks.
- `convergence` is a binary vector containing whether a network was successfully estimated for the given combination of hyperparameters or not.
- `path` is a list containing the adjacency matrices of all the estimated networks.
- `icov` is a list containing the inverse covariance matrices of all the estimated networks.
- `nexploded` is the number of combinations of hyperparameters for which `coglasso()` failed to converge.
- `data` is the input multi-omics data set.
- `hpars` is the ordered table of all the combinations of hyperparameters given as input to `coglasso()`, with $\alpha(\lambda_w + \lambda_b)$ being the key to sort rows.

- lambda_w is a numerical vector with all the λ_w values coglasso() used.
- lambda_b is a numerical vector with all the λ_b values coglasso() used.
- c is a numerical vector with all the c values coglasso() used.
- pX is the number of variables of the first data set.
- cov optional, returned when cov_output is TRUE, is a list containing the variance-covariance matrices of all the estimated networks.

Examples

```
# Typical usage: set the number of hyperparameters to explore
cg <- coglasso(multi_omics_sd_micro, pX = 4, nlambda_w = 3, nlambda_b = 3, nc = 3, verbose = FALSE)
```

---

multi_omics_sd *Multi-omics dataset of sleep deprivation in mouse*

Description

A dataset containing transcript and metabolite values analysed in Albanese et al. 2023, subset of the multi-omics data set published in Jan, M., Gobet, N., Diessler, S. et al. A multi-omics digital research object for the genetics of sleep regulation. Sci Data 6, 258 (2019).

multi_omics_sd_small is a smaller version, limited to the transcript Cirbp and the transcripts and metabolites belonging to its neighborhood as described in Albanese et al. 2023

multi_omics_sd_micro is a minimal version with Cirbp and a selection of its neighborhood.

Usage

```
multi_omics_sd

multi_omics_sd_small

multi_omics_sd_micro
```

Format

multi_omics_sd**:**

A data frame with 30 rows and 238 variables (162 transcripts and 76 metabolites):

Plin4 to Tfrc log2 CPM values of 162 transcripts in mouse cortex under sleep deprivation (-4.52–10.46)

Ala to SM C24:1 abundance values of 76 metabolites (0.02–1112.67)

multi_omics_sd_small**:**

A data frame with 30 rows and 19 variables (14 transcripts and 5 metabolites)

Cirbp to Stip1 log2 CPM values of 14 transcripts in mouse cortex under sleep deprivation (4.24–9.31)

Phe to PC ae C32:2 Abundance values of 5 metabolites (0.17–145.33)

`multi_omics_sd_micro`:

A data frame with 30 rows and 6 variables (4 transcripts and 2 metabolites)

Cirbp to Dnajb11 log2 CPM values of 4 transcripts in mouse cortex under sleep deprivation (4.78–9.31)

Trp to PC aa C36:3 Abundance values of 2 metabolites (58.80–145.33)

Source

Jan, M., Gobet, N., Diessler, S. et al. A multi-omics digital research object for the genetics of sleep regulation. Sci Data 6, 258 (2019) doi:10.1038/s415970190171x

Figshare folder of the original manuscript: `https://figshare.com/articles/dataset/Input_data_for_systems_genetics_of_sleep_regulation/7797434`

`stars_coglasso` *Stability selection of the best* `coglasso` *network*

Description

`stars_coglasso()` selects the combination of hyperparameters given to `coglasso()` yielding the most stable, yet sparse network. Stability is computed upon network estimation from subsamples of the multi-omics data set, allowing repetition. Subsamples are collected for a fixed amount of times (`rep_num`), and with a fixed proportion of the total number of samples (`stars_subsample_ratio`).

Usage

```
stars_coglasso(
  coglasso_obj,
  stars_thresh = 0.1,
  stars_subsample_ratio = NULL,
  rep_num = 20,
  max_iter = 10,
  verbose = TRUE
)
```

Arguments

`coglasso_obj` The object returned by `coglasso()`.

`stars_thresh` The threshold set for variability of the explored networks at each iteration of the algorithm. The λ_w or the λ_b associated to the most stable network before the threshold is overcome is selected.

`stars_subsample_ratio`
The proportion of samples in the multi-omics data set to be randomly subsampled to estimate the variability of the network under the given hyperparameters setting. Defaults to 80% when the number of samples is smaller than 144, otherwise it defaults to $\frac{10}{n}\sqrt{n}$.

`rep_num` The amount of subsamples of the multi-omics data set used to estimate the variability of the network under the given hyperparameters setting. Defaults to 20.

`max_iter` The greatest number of times the algorithm is allowed to choose a new best λ_w. Defaults to 10.

`verbose` Print information regarding the progress of the selection procedure on the console.

Details

StARS for *collaborative graphical regression* is an adaptation of the method published by Liu, H. *et al.* (2010): Stability Approach to Regularization Selection (StARS). *StARS* was developed for network estimation regulated by a single penalty parameter, while collaborative graphical lasso needs to explore three different hyperparameters. In particular, two of these are penalty parameters with a direct influence on network sparsity, hence on stability. For every c parameter, `stars_coglasso()` explores one of the two penalty parameters (λ_w or λ_b), keeping the other one fixed at its previous best estimate, using the normal, one-dimentional *StARS* approach, until finding the best couple. It then selects the c parameter for which the best (λ_w, λ_b) couple yielded the most stable, yet sparse network.

Value

`stars_coglasso()` returns a list containing the results of the selection procedure, built upon the list returned by `coglasso()`.

- ... are the same elements returned by `coglasso()`.
- `merge_lw` and `merge_lb` are lists with as many elements as the number of c parameters explored. Every element is in turn a list of as many matrices as the number of λ_w (or λ_b) values explored. Each matrix is the "merged" adjacency matrix, the average of all the adjacency matrices estimated for those specific c and λ_w (or λ_b) values across all the subsampling in the last path explored before convergence, the one when the final combination of λ_w and λ_b is selected for the given c value.
- `variability_lw` and `variability_lb` are lists with as many elements as the number of c parameters explored. Every element is a numeric vector of as many items as the number of λ_w (or λ_b) values explored. Each item is the variability of the network estimated for those specific c and λ_w (or λ_b) values in the last path explored before convergence, the one when the final combination of λ_w and λ_b is selected for the given c value.
- `opt_adj` is a list of the adjacency matrices finally selected for each c parameter explored.
- `opt_variability` is a numerical vector containing the variabilities associated to the adjacency matrices in `opt_adj`.
- `opt_index_lw` and `opt_index_lb` are integer vectors containing the index of the selected λ_ws (or λ_bs) for each c parameters explored.
- `opt_lambda_w` and `opt_lambda_b` are vectors containing the selected λ_ws (or λ_bs) for each c parameters explored.
- `sel_index_c`, `sel_index_lw` and `sel_index_lb` are the indexes of the final selected parameters c, λ_w and λ_b leading to the most stable sparse network.
- `sel_c`, `sel_lambda_w` and `sel_lambda_b` are the final selected parameters c, λ_w and λ_b leading to the most stable sparse network.

- `sel_adj` is the adjacency matrix of the final selected network.
- `sel_density` is the density of the final selected network.
- `sel_icov` is the inverse covariance matrix of the final selected network.

Examples

```
cg <- coglasso(multi_omics_sd_micro, pX = 4, nlambda_w = 3, nlambda_b = 3, nc = 3, verbose = FALSE)

# Takes around 20 seconds
sel_cg <- stars_coglasso(cg, verbose = FALSE)
```

Index