Package ‘cpmr’

October 6, 2024

Title Connectome Predictive Modelling in R

Version 0.1.0

Description Connectome Predictive Modelling (CPM) (Shen et al. (2017) <doi:10.1038/nprot.2016.178>) is a method to predict individual differences in behaviour from brain functional connectivity. 'cpmr' provides a simple yet efficient implementation of this method.

License MIT + file LICENSE

URL `https://github.com/psychelzh/cpmr`, `https://psychelzh.github.io/cpmr/`

BugReports `https://github.com/psychelzh/cpmr/issues`

Imports generics, Rfast, stats, tibble

Suggests testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Liang Zhang [aut, cre] (<`https://orcid.org/0000-0001-9041-1150`>)

Maintainer Liang Zhang <`psychelzh@outlook.com`>

Repository CRAN

Date/Publication 2024-10-06 07:30:02 UTC

Contents

`cpm` *Perform Connectome-based Predictive Modeling (CPM)*

Description

The connectome-based predictive modeling (CPM) is a data-driven approach to predict individual behavior from brain connectivity data. Originally proposed by Shen et al. (2017), the CPM has been widely used in various studies. This function implements the CPM algorithm and provides a convenient interface to use it.

Usage

```
cpm(
  conmat,
  behav,
  ...,
  confounds = NULL,
  thresh_method = c("alpha", "sparsity"),
  thresh_level = 0.01,
  kfolds = NULL,
  bias_correct = TRUE,
  return_edges = c("sum", "none", "all"),
  na_action = c("fail", "exclude")
)
```

Arguments

`conmat` A matrix of connectome data. Observations in row, edges in column (assumed that duplicated edges are removed).

`behav` A numeric vector contains behavior data. Length must equal to number of observations in `conmat`. Note `behav` could also be a row/column matrix, which will be converted to a vector using `drop()`.

`...` For future extension. Currently ignored.

`confounds` A matrix of confounding variables. Observations in row, variables in column. If `NULL`, no confounding variables are used. Note if a vector is provided, it will be converted to a column matrix.

`thresh_method, thresh_level`
The threshold method and level used in edge selection. If method is set to be `"alpha"`, the edge selection is based on the critical value of correlation coefficient. If method is set to be `"sparsity"`, the edge selection is based on the quantile of correlation coefficient, thus network sparsity is controlled.

`kfolds` Folds number of cross-validation. If `NULL`, it will be set to be equal to the number of observations, i.e., leave-one-subject-out.

`bias_correct` Logical value indicating if the connectome data should be bias-corrected. If `TRUE`, the connectome data will be centered and scaled to have unit variance based on the training data before model fitting and prediction. See Rapuano et al. (2020) for more details.

`return_edges` A character string indicating the return value of the selected edges. If `"none"`, no edges are returned. If `"sum"`, the sum of selected edges across folds is returned. If `"all"`, the selected edges for each fold is returned, which is a 3D array and memory-consuming.

`na_action` A character string indicating the action when missing values are found in `behav`. If `"fail"`, an error will be thrown. If `"exclude"`, missing values will be excluded from the analysis but kept in the output. Note complete cases are intersection of `conmat`, `behav` and `confounds` if provided.

Value

A list with the following components:

`folds` The corresponding fold for each observation when used as test group in cross-validation.

`real` The real behavior data. This is the same as the input `behav` if `confounds` is `NULL`, otherwise it is the residual of `behav` after regressing out `confounds`.

`pred` The predicted behavior data, with each column corresponding to a model, i.e., both edges, positive edges, negative edges, and the row names corresponding to the observation names (the same as those of `behav`).

`edges` The selected edges, if `return_edges` is not `"none"`. If `return_edges` is `"sum"`, it is a matrix with rows corresponding to edges and columns corresponding to networks. If `return_edges` is `"all"`, it is a 3D array with dimensions corresponding to edges, networks and folds.

`call` The matched call.

`params` A list of parameters used in the function, including:

- `confounds` indicating if confounds are used
- `thresh_method` indicating the threshold method
- `thresh_level` indicating the threshold level
- `kfolds` indicating the number of folds in cross-validation
- `bias_correct` indicating if bias correction is used

References

Shen, X., Finn, E. S., Scheinost, D., Rosenberg, M. D., Chun, M. M., Papademetris, X., & Constable, R. T. (2017). Using connectome-based predictive modeling to predict individual behavior from brain connectivity. Nature Protocols, 12(3), 506–518. https://doi.org/10.1038/nprot.2016.178

Rapuano, K. M., Rosenberg, M. D., Maza, M. T., Dennis, N. J., Dorji, M., Greene, A. S., Horien, C., Scheinost, D., Todd Constable, R., & Casey, B. J. (2020). Behavioral and brain signatures of substance use vulnerability in childhood. Developmental Cognitive Neuroscience, 46, 100878. https://doi.org/10.1016/j.dcn.2020.100878

Examples

```
conmat <- matrix(rnorm(100 * 100), nrow = 100)
behav <- rnorm(100)
cpm(conmat, behav)
# use different threshold method and level
cpm(conmat, behav, thresh_method = "sparsity", thresh_level = 0.05)
# use a 10-fold cross-validation
cpm(conmat, behav, kfolds = 10)
```

summary.cpm *Summary of a cpm object.*

Description

This function provides a summary of a cpm object, including the prediction performance and the selected edges.

Usage

```
## S3 method for class 'cpm'
summary(object, ..., method = c("pearson", "spearman"), edge_level = 0.5)

## S3 method for class 'cpm_summary'
print(x, ...)
```

Arguments

object	An object of class cpm.
...	Other parameters passed to the function.
method	A character vector indicating the method used to calculate the correlation between the real and predicted values.
edge_level	A numeric value between 0 and 1 indicating the proportional threshold for edge selection.
x	An object of class cpm_summary.

Value

A list of class cpm_summary containing two elements:

performance	A matrix of prediction performance, including the correlation between the real and predicted values for both edges, positive edges only, and negative edges only.
edges	A logical vector indicating whether each edge is selected based on the edge_level.
params	A list of parameters used in the summary.

tidy.cpm *Tidy a cpm object*

Description

Tidy a cpm object

Usage

```
## S3 method for class 'cpm'
tidy(x, ..., component = c("performance", "edges"))
```

Arguments

x	A cpm object.
...	Additional arguments passed to summary().
component	A character vector indicating the component to tidy.

Value

A tibble with columns storing parameters of the cpm() object and further columns depending on the component argument:

For component = "performance":

method	The method used to calculate the correlation between the real and predicted values.
pos	The correlation between the real and predicted values for positive edges.
neg	The correlation between the real and predicted values for negative edges.

For component = "edges":

level	The proportional threshold for edge selection.
pos	A logical vector indicating whether each edge is selected based on the edge_level (positive).
neg	A logical vector indicating whether each edge is selected based on the edge_level (negative).

Index