

Package: cpmr (via r-universe)

September 9, 2024

Title Connectome Predictive Modelling in R

Version 0.0.9.9000

Description Connectome Predictive Modelling (CPM) (Shen et al. (2017) [doi:10.1038/nprot.2016.178](https://doi.org/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 Rfast, stats

Suggests testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository <https://psychelzh.r-universe.dev>

RemoteUrl <https://github.com/psychelzh/cpmr>

RemoteRef HEAD

RemoteSha 431ded360a0d4ed9c9122c861d305f126e1cac83

Contents

cpm	2
summary.cpm	4
Index	5

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("none", "sum", "all")
)
```

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 <code>drop()</code> .
...	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.

Value

A list with the following components:

<code>fold</code> s	The corresponding fold for each observation when used as test group in cross-validation.
<code>real</code>	The real behavior data. This is the same as the input <code>behav</code> if <code>confounds</code> is NULL, otherwise it is the residual of <code>behav</code> after regressing out <code>confounds</code> .
<code>pred</code>	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 <code>behav</code>).
<code>edges</code>	The selected edges, if <code>return_edges</code> is not "none". If <code>return_edges</code> is "sum", it is a matrix with rows corresponding to edges and columns corresponding to networks. If <code>return_edges</code> is "all", it is a 3D array with dimensions corresponding to folds, edges, and networks.

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

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

Value

A list of class `cpm_summary` containing two elements:

<code>performance</code>	A matrix of prediction performance, including the correlation between the real and predicted values for both edges, positive edges only, and negative edges only.
<code>edges</code>	A logical vector indicating whether each edge is selected based on the <code>edge_level</code> .

Index

`cpm`, [2](#)

`drop()`, [2](#)

`print.cpm_summary(summary.cpm)`, [4](#)

`summary.cpm`, [4](#)