Automated SmCCNet

Weixuan Liu Katerina Kechris

2024-01-13

Contents

Automated SmCCNet

In this version of the SmCCNet package, we introduce a pipeline known as Automated SmCCNet. This method streamlines the SmCCNet code and significantly reduces computation time. Users are simply required to input a list of omics data and a phenotype variable. The program then automatically determines whether it is dealing with a single-omics or multi-omics problem, and whether to use CCA or PLS for quantitative or binary phenotypes respectively. For details of how each method is established and how parameters and coefficients are set, we recommend the user to refer to the multi-omics and single-omics vignettes.

Specifically, for multi-omics SmCCNet, if CCA is employed, the program can automatically select the scaling factors (importance of the pair-wise omics or omics-phenotype correlations to the objective function). This is achieved by calculating the pairwise canonical correlation between each pair of omics under the most stringent penalty parameters. The scaling factor for the omics data A, B pair in SmCCA is set to the absolute value of the pairwise canonical correlation between omics A and B divided by the between omics correlation shrinkage parameter. By default, all scaling factors linked to the phenotype-specific correlation structure are set to 1. In Automated SmCCNet, users only need to provide a BetweenShrinkage parameter, a positive real number that helps reduce the significance of the omics-omics correlation component. The larger this number, the more the between omics correlation is shrunk.

Moreover, for multi-omics SmCCNet with a binary phenotype, the scaling factor

is not implemented. However, the user need to provide values for γ_1 (omics-omics connection importance) and γ_2 (omics-phenotype connection importance, see multi-omics vignette section 5 for detail). The automated SmCCNet program offers a method to calculate γ_1 while setting the value of γ_2 to 1. This is generally done by averaging all the pairwise omics-omics canonical correlations in the multi-omics dataset.

The program can also automatically select the percentage of features subsampled. If the number of features from an omics data is less than 300, then the percentage of feature subsampled is set to 0.9, otherwise, it's set to 0.7. The candidate penalty terms range from 0.1 to 0.5 with a step size of 0.1 for single/multi-omics SmCCA, and from 0.5 to 0.9 with a step size of 0.1 for single/multi-omics SPLSDA (for both omics-omics SmCCA step and omics-phenotype classifier, see section 5 in multi-omics vignette for detail).

This automated version of SmCCNet is typically faster than the standard SmCCNet. This is due to the heuristic selection of the scaling factors (see section 4.2 in the multi-omics vignette), and the parallelization of the cross-validation step, resulting in a substantial increase in computational speed. Below is an example of how to implement Automated SmCCNet. For more detailed information, please refer to the **FastAutoSmCCNet()** function help file:

Function Arguments and Tuning Parameters

X: A list of omics matrices with same set and order of subjects

Y: Phenotype variable of either numeric or binary, for binary variable, for binary Y, it should be binarized to 0 and 1 before running this function.

AdjustedCovar: A data frame of covariates of interest to be adjusted for through regressing-out approach, **preprocess** must be set to TRUE.

Kfold: Number of folds for cross-validation, default is set to 5.

EvalMethod: For single or multi-omics with binary phenotype, the evaluation methods used to selected the optimal penalty parameter(s). The selections is among 'accuracy', 'auc', 'precision', 'recall', and 'f1', default is set to 'accuracy'.

subSampNum: Number of subsampling to run, the higher the better in terms of accuracy, but at a cost of computational time, we generally recommend 500-1000 to increase robustness.

BetweenShrinkage: A real number > 0 that helps shrink the importance of omics-omics correlation component, the larger this number is, the greater the shrinkage it is, default is set to 2.

ScalingPen: A numeric vector of length 2 used as the penalty terms for scaling factor selection method, default set to 0.1, and should be between 0 and 1.

DataType: A vector indicating omics type for each element of X, example would be c('gene', 'miRNA').

CutHeight: A numeric value specifying the cut height for hierarchical clustering, should be between 0 and 1, default is set to $1 - 0.1^{10}$.

min_size: Minimally possible subnetwork size after network pruning, default set to 10.

max_size: Maximally possible subnetwork size after network pruning, default set to 100.

summarization: Summarization method used for network pruning and summarization, should be either 'NetSHy' or 'PCA', default is set to 'NetSHy'.

saving_dir: Directory where user would like to store the subnetwork results, default is set to the current working directory.

preprocess: TRUE or FALSE, Whether the data preprocessing step should be conducted, default is set to FALSE. If covariates adjustment is needed, add covariates to the **AdjustedCovar** argument.

ncomp_pls: Number of components for PLS algorithm, only used when binary phenotype is given, default is set to 3.

tuneLength: The total number of candidate penalty term values for each omics data, default is set to 5.

tuneRangeCCA: A vector of length 2 that represents the range of candidate penalty term values for each omics data based on canonical correlation analysis, default is set to c(0.1,0.5).

tuneRangePLS: A vector of length 2 that represents the range of candidate penalty term values for each omics data based on partial least squared discriminant analysis, default is set to c(0.5,0.9).

seed: Random seed for result reproducibility, default is set to 123.

Examples

We present below examples of how to execute Automated SmCCNet using a simulated dataset. In this demonstration, we simulate four datasets: two omics data and one phenotype data. We cover four cases in total, involving combinations of single or multi-omics data with either a quantitative or binary phenotype. The final case demonstrates the use of the regress-out approach for covariate adjustment. If users want to run through the pipeline step-by-step or understand more about the algorithm used, please refer to SmCCNet single or multi-omics vignettes for details.

```
library(SmCCNet)
set.seed(123)
```

```r
data("ExampleData")
Y_binary <- ifelse(Y > quantile(Y, 0.5), 1, 0)
# single-omics PLS
result <- fastAutoSmCCNet(X = list(X1), Y = as.factor(Y_binary),
                          Kfold = 3,
                          subSampNum = 100, DataType = c('Gene'),
                          saving_dir = getwd(), EvalMethod = 'auc',
                          summarization = 'NetSHy',
                          CutHeight = 1 - 0.1^10, ncomp_pls = 5)
# single-omics CCA
result <- fastAutoSmCCNet(X = list(X1), Y = Y, Kfold = 3,
                          preprocess = FALSE,
                          subSampNum = 50, DataType = c('Gene'),
                          saving_dir = getwd(), summarization = 'NetSHy',
                          CutHeight = 1 - 0.1^10)
# multi-omics PLS
result <- fastAutoSmCCNet(X = list(X1,X2), Y = as.factor(Y_binary),
                          Kfold = 3, subSampNum = 50,
                          DataType = c('Gene', 'miRNA'),
                          CutHeight = 1 - 0.1^10,
                          saving_dir = getwd(),
                          EvalMethod = 'auc',
                          summarization = 'NetSHy',
                          BetweenShrinkage = 5,
                          ncomp_pls = 3)
# multi-omics CCA
result <- fastAutoSmCCNet(X = list(X1,X2), Y = Y,
                          K = 3, subSampNum = 50,
                          DataType = c('Gene', 'miRNA'),
                          CutHeight = 1 - 0.1^10,
                          saving_dir = getwd(),
                          summarization = 'NetSHy',
                          BetweenShrinkage = 5)
```

Global network information will be stored in object 'result', and subnetwork information will be stored in the directory user provide. For more information about using Cytoscape to visualize the subnetworks, please refer back to the multi-omics vignette section 3.1.