BrainNetClass Toolbox v1.0

Manual

(Version 05-01-2019)

Overview

Researches have used either static or dynamic functional connectivity networks as metrics to perform classification between patients and normal controls. The aim of this toolbox is to make it easier for neuroscientists and clinicians to conduct state-of-the-art brain network construction and rigorous machine learning-based classifications.

This toolbox is developed by Zhen Zhou, Xiaobo Chen, Yu Zhang, Han Zhang, and Dinggang Shen[1]. The brain network construction algorithms were contributed by Lishan Qiao, Renping Yu, Xiaobo Chen, Yu Zhang, and Han Zhang.

Contact:
Zhen Zhou: zzstefan@email.unc.edu
Han Zhang: hanzhang@med.unc.edu
Dinggang Shen: dgshen@med.unc.edu

Image Display, Enhancement, and Analysis (IDEA) Group
Department of Radiology and Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

If writing papers using our toolbox, please cite the following:
To be added.

[1] This work is supported by NIH grants EB022880, AG049371, AG042599, and AG041721.

Installation

The recommended environment is:
MATLAB version 2017a and higher, running in Windows 10 and Ubuntu 16.04 (this toolbox has been tested successfully based on these two environments).

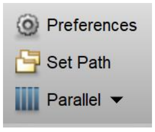
The installation of BrainNetClass is easy. Simply download and unzip the package and add the its main folder by using *addpath* to the MATLAB working path. There are two options to addpath:

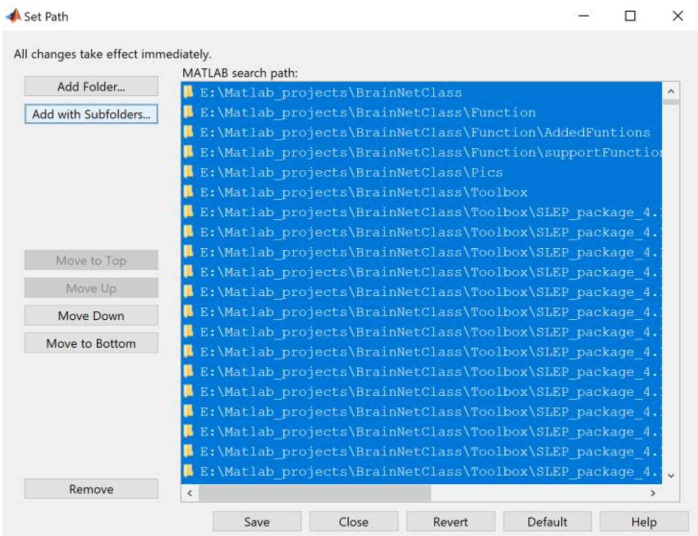
- Command line

Type the following command line in the MATLAB command window:
>> addpath(genpath('D:\BrainNetClass'));
Note: 'D: \BrainNetClass' is the exemplary path pf BrainNetClass on your computer.

- Interface

Click 'Set Path' on the MATLAB panel, or type 'pathtool' in the MATLAB command window. Click 'Add with Subfolders…' button, and select your path, i.e., 'D:\BrainNetClass'.





Click 'Save' to save your change. If you don't have permission to save your changes on your computer (e.g., on the server), please save pathdef.m to another location where you will often launch MATLAB.

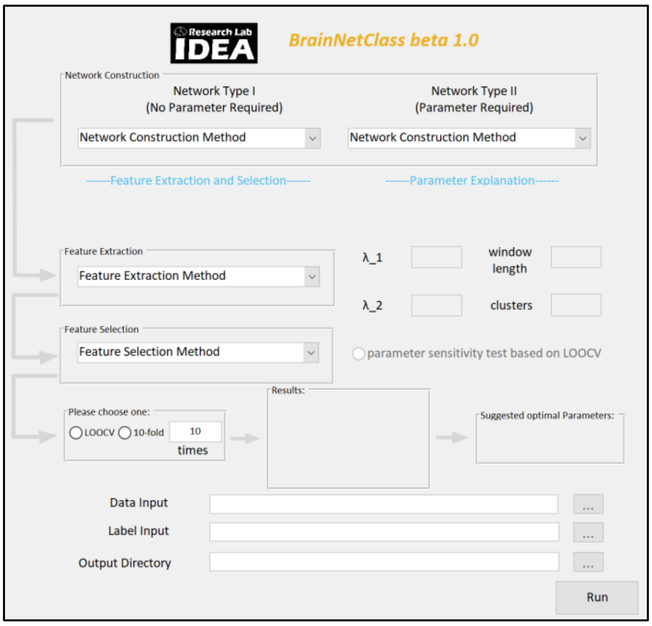
Warning: Please make sure your BrainNetClass path DOES NOT include any space or special character.

Note: to run the toolbox, we need a compiled libsvm library and SLEP library. Although the compiled version of them are included in the toolbox, we highly recommend user make a compiled version by themselves. To compile libsvm, after adding path, please type in the command window:
>> cd 'D:\BrainNetClass';
>> cd 'toolbox/libsvm-3.23';

>> mex -setup
To compile SLEP,
>> cd ‘D:\BrainNetClass’;
>> cd ‘toolbox/SLEP_package_4.1’;
>> mex -setup

Note: the compiling of these two package may require some compliers, please refer to https://www.mathworks.com/support/requirements/supported-compilers.html.

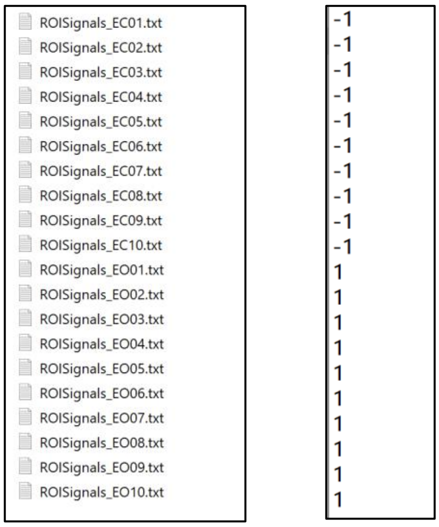
After installation, to run the toolbox, type ‘BrainNetClass’ in the MATLAB command window. The following GUI window will pop up.



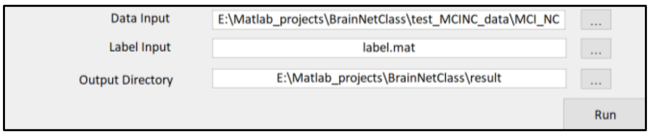
Inputs

First, user should specify the input folder that contains the input data files. They should both be in the *.txt file format.

The format of regional time series text file must be a matrix with rows denoting the time points and columns representing the ROIs. Each file for each subject (see the left panel of the figure below) It can be obtained by the DPABI software or other softwares using the extracting ROI time courses module. The label text file should be prepared like the following figure (right panel), with each label located in one line and corresponding to the data file. E.g., -1 represents patient and 1 represents control.



The output directory should also be specified at the beginning. After inputs being specified, it should be like this:



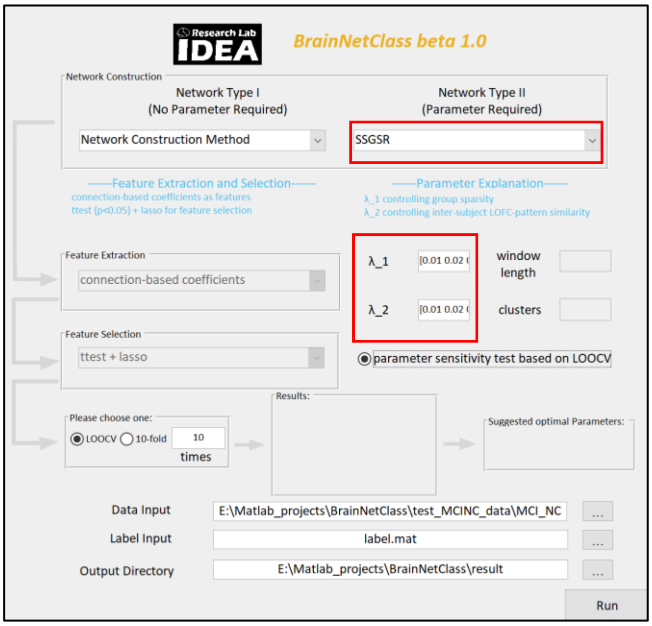
Brain network construction

Then, user should choose which brain construction method to use. The available methods can be categorized into two types: those with No Parameter Required (Network Type I), e.g., PC, aHOFC and tHOFC, and those with Parameter Required (Network Type II), e.g., SR, GSR, WSR, WSGR, SGR, SSGSR, SLR and dHOFC. For details, see below and please refer to the mentioned original papers.

It is not required to pre-specify parameters for the Network Type I, but when choosing the method in Network Type II, user needs to specify the related hyper-parameter by giving the ranges of them. Although the user could use the default parameter ranges given by the toolbox, it is recommended to carefully choose them as the parameters could significantly affect the constructed brain networks and the classification results. The toolbox will provide "suggested" parameters for users but they are allowed to change the default parameters based on their own preferences.

For example, as shown in the figure below, the default ranges for both parameters (λ_1 and λ_2) of the SSGSR method are from 0.01 to 0.1 with the increment of 0.01, i.e., [0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1]. The roles of the λ_1 and λ_2 will be shown in the panel above in blue.

For all the SR based methods, we have λ_1 (and λ_2) to be specified, while for the dHOFC method, we have *window length* and *number of clusters* to be specified since it uses dynamic functional connectivity.



Below is a brief explanation of all the available brain network construction methods. For more details, please see our toolbox paper (to be added) and their respective original papers listed behind.

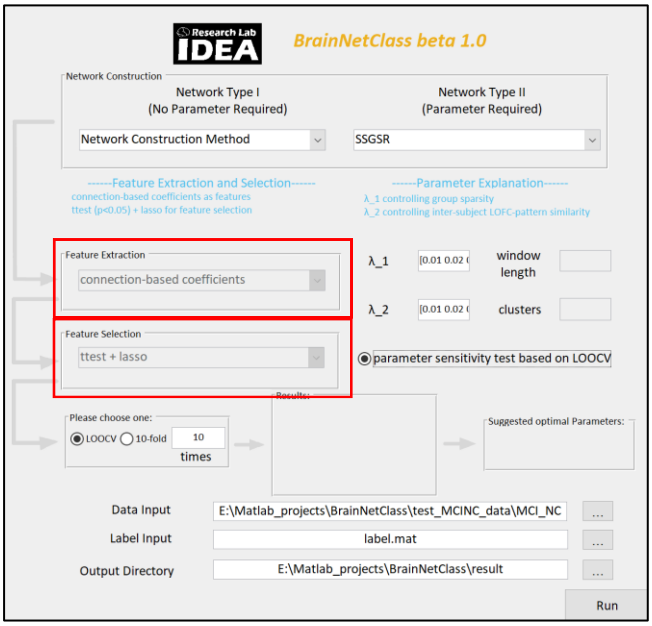
1. PC: Pearson's correlation. The most conventional method. It can be used as baseline method. No parameter is required.
2. SR: Sparse representation with a L1-norm constraint. One parameter is required. When you expect the network is sparse or there is heavy noise in the data, you may use it. It can also be used as baseline method.
3. GSR: Group sparse representation. It makes sure that all subjects have similar FC network pattern. One parameter is required. (Wee et al., 2014)
4. SSGSR: Generate within-group similar networks but retain necessary between-group difference. Two parameters are required. (Zhang et al., 2019)
5. SGR: Sparse group representation. Combining L1-norm and Lq,1-norm to preserve certain structured information in the adjacency matrix. Two parameters are required.
6. WSR: FC-weighted SR. SR-based network construction but with strong FC preserved. One parameter is required. (Yu et al., 2017)
7. WSGR: Similar to SGR but with strong FC preserved. Two parameters are required. (Yu et al., 2017)
8. LSR: Low-rank constraint-based SR. Make sure that the network is both sparse and structured (having some modular structures). Two parameters are required. (Qiao et al., 2016)
9. tHOFC (topographical similarity-based HOFC): A high-order FC metric, inter-regional functional relationship is estimated by the FC topological similarity rather than the BOLD signal similarity. No parameter is required. (Zhang et al., 2016)
10. aHOFC (associated HOFC): A further step from tHOFC measuring inter-level HOFC (the similarity between HOFC and conventional FC topological profiles. No parameter is required. (Zhang et al., 2016)
11. dHOFC: Dynamic FC-based HOFC that measures temporal synchronization of dynamic FC time series. Two parameters are required. (Chen et al., 2016)

Feature extraction and selection

After choosing a network construction method, user needs to choose the methods for feature extraction and selection. We provide two types of feature extraction methods: connection-based coefficients (i.e., directly use the connectivity strength of each link as features) and weighted-graph local clustering coefficient (i.e., one of the widely used nodal character calculated based on a weighted graph). For more details, please see (Chen et al., 2016; Zhang et al., 2019).

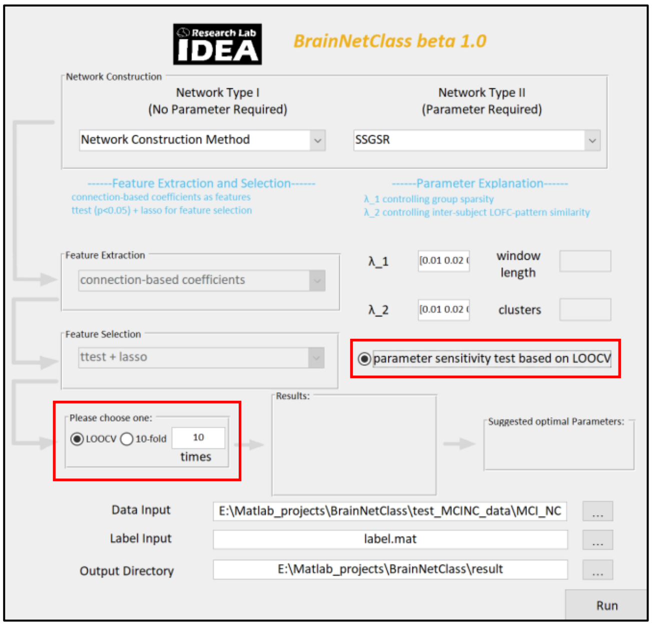
If user chooses the network construction method requiring no parameters, it will go straight to the feature extraction (i.e., connection coefficients and weighted local clustering coefficients) and selection (i.e., *t*-test, LASSO and *t*-test + LASSO) part.

If user chooses the network construction method requiring parameter(s), specific feature extraction/selection method will be automatically suggested to the user and it cannot be changed. For example, for the SR-based methods, users are restricted to use the connection-based coefficients as features combining with *t*-test+*LASSO* to perform feature selection (see the figure below). For the dHOFC method, users are restricted to use weighted-graph local clustering coefficients as features and LASSO to perform feature selection.



Classification

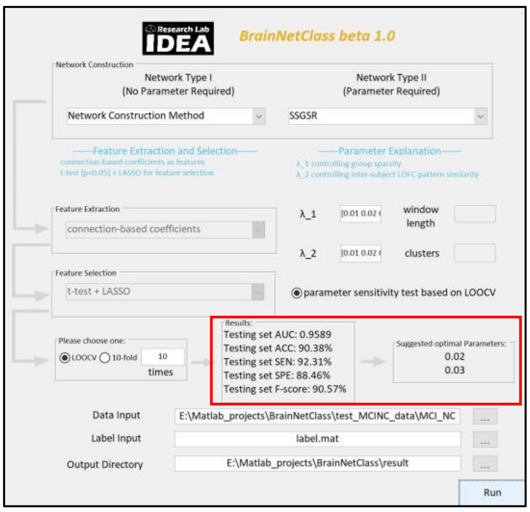
We provided two types of cross validation strategies (see figure below). The first one is leave-one-out cross validation (LOOCV) and the other is 10-fold cross validation. If the sample size is large, we recommend the 10-fold cross validation. With limited sample size, we recommend LOOCV. Of note, Notably, 10-fold cross validation will be run for many times (default: 10 times, user may specify more than 10 times, e.g., 100, but it will increase the processing time) as the result of 10-fold cross validation heavily depends on data partitioning.

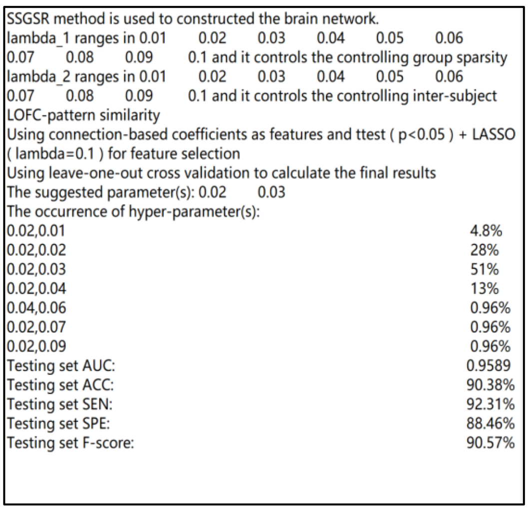


There is also a parameter sensitivity test to assess the effects of the hyper-parameters on classification accuracy using LOOCV with all the subjects. See (Zhang et al., 2019) for the rationale for doing so. It is suggested to include this into the paper. The toolbox also counts the occurrence of each combination of the hyper-parameters. This can be used to test the model robustness and is suggested to include this into the paper.

Result display

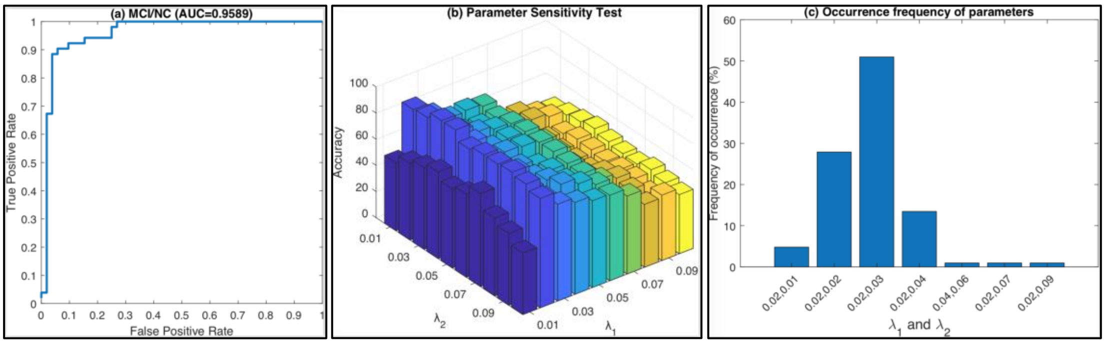
When all the computational processes are finished, the model performance will be displayed on the GUI panel (see the left panel of the figure below). The results are also recorded and saved in a log text file. This log file records the method used to construct the brain network, the hyper-parameters, feature extraction and selection methods, cross validation methods, suggested parameter(s), the occurrence of parameters, and the performance of the classifier (see the right panel of the figure below).




SSGSR method is used to constructed the brain network.
lambda_1 ranges in 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 and it controls the controlling group sparsity
lambda_2 ranges in 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 and it controls the controlling inter-subject LOFC-pattern similarity
Using connection-based coefficients as features and ttest (p<0.05) + LASSO (lambda=0.1) for feature selection
Using leave-one-out cross validation to calculate the final results
The suggested parameter(s): 0.02 0.03
The occurrence of hyper-parameter(s):

0.02,0.01	4.8%
0.02,0.02	28%
0.02,0.03	51%
0.02,0.04	13%
0.04,0.06	0.96%
0.02,0.07	0.96%
0.02,0.09	0.96%
Testing set AUC:	0.9589
Testing set ACC:	90.38%
Testing set SEN:	92.31%
Testing set SPE:	88.46%
Testing set F-score:	90.57%

Furthermore, the toolbox will generate three figures in the result folder, which are the Receiver Operating Characteristic (ROC) curve of the classification performance (a), classification accuracy from the parameter sensitivity test (b), and the parameter occurrence (c). For interpretation, please see our toolbox paper (to be added).



Test data

We provide some exemplary data for user to get familiar with the usage of the toolbox. The data is from http://fcon_1000.projects.nitrc.org/indi/retro/BeijingEOEC.html (Beijing: Eyes Open Eyes Closed Study). The goal is to use resting-state fMRI time series from 116 brain regions to construct brain functional networks and predict whether the subject is in eyes closed state or eyes open state. There is a full version (with longer running time but better classification result) and a simplified version of the data (with shorter running time, for fast running). The labels (EC, EO) are provided with two versions as well (see label.txt and label_simple.txt). For more details, please see our toolbox paper (to be added).

References

Chen, X., Zhang, H., Gao, Y., Wee, C.Y., Li, G., Shen, D., Alzheimer's Disease Neuroimaging, I., 2016. High-order resting-state functional connectivity network for MCI classification. Hum Brain Mapp 37, 3282-3296.

Qiao, L., Zhang, H., Kim, M., Teng, S., Zhang, L., Shen, D., 2016. Estimating functional brain networks by incorporating a modularity prior. Neuroimage 141, 399-407.

Wee, C.Y., Yap, P.T., Zhang, D., Wang, L., Shen, D., 2014. Group-constrained sparse fMRI connectivity modeling for mild cognitive impairment identification. Brain Struct Funct 219, 641-656.

Yu, R., Zhang, H., An, L., Chen, X., Wei, Z., Shen, D., 2017. Connectivity strength-weighted sparse group representation-based brain network construction for MCI classification. Hum Brain Mapp 38, 2370-2383.

Zhang, H., Chen, X., Shi, F., Li, G., Kim, M., Giannakopoulos, P., Haller, S., Shen, D., 2016. Topographical Information-Based High-Order Functional Connectivity and Its Application in Abnormality Detection for Mild Cognitive Impairment. J Alzheimers Dis 54, 1095-1112.

Zhang, Y., Zhang, H., Chen, X., Liu, M., Zhu, X., Lee, S.-W., Shen, D., 2019. Strength and similarity guided group-level brain functional network construction for MCI diagnosis. Pattern Recognition 88, 421-430.