A software package of optimizing higher-order network structures for maximizing the synchronization of coupled oscillators

(c) 2021 Ying Tang

All rights reserved.

This MATLAB code package optimizes network topology for synchronization of coupled oscillators with higher-order interactions. The current focus is the system with Kuramoto-type coupling function for identical oscillators, the second-order interactions (triangle). The optimization is realized by minimizing the eigenratios or the spread of eigenvalues for the generalized Laplacian matrices. For the undirected network, we rewire the triangle interactions and use simulated annealing to optimize the network synchronizability. For the directed network, we selectively remove directional triangle interactions to optimize synchronizability, and investigate asymmetry for the optimized directed network.

A detailed description on the methods is given in the main text.

Contact: Ying Tang, jamestang23@gmail.com

--------------------------------------------------------------------------------------------------------------------------------------------

System requirements:

All simulations were done using MATLAB® version R2019a.

Third-party packages:

(1) We have used the package of analyzing first-order network “octave-networks-toolbox”, which is available at: <https://github.com/aeolianine/octave-networks-toolbox> . Please make sure to have it in the current folder, and then the scripts optiUndirected\_rewire\_high2.m and optiDirected\_remove\_high2.m will add path of this package.

(2) The simulated annealing algorithm is based on the previous package on the first-order network https://github.com/y-z-zhang/optimize\_sym\_cluster , and is extended to second-order network.

Expected run time: all the expected run time below is evaluated based on a personal desktop with intel(R) core(tm) i7-8700 CPU @ 3.7GHz.

--------------------------------------------------------------------------------------------------------------------------------------------

Guideline:

A step-by-step guideline is given below.

*## Undirected network:*

# 1. optiUndirected\_rewire\_high2.m

# The script is for rewiring the triangle interaction and using simulated annealing to optimize the network synchronizability. It can scan different network sizes, and repeat multiple numerical replicates with different randomly initialized network.

# User input parameters: (a) InitialNet: the two types of initialization as described in text; (b) realization: the number of numerical replicates; (c) NumberNode: the number of network nodes;

# Parameters which can be tuned: (d) Run: the number of independent simulated annealing with different randomly initialized network; (e) NumIncreaseTarget: the number of simulated annealing before the target eigenratio increased by 10%, because the target eigenratio cannot be too small as it may not be reached due to the sparsity for large networks.

# Expected output: A matlab file containing eigenratios, network tensor, generalized Laplacian matrices for the networks before and after optimization, in different sizes.

# Expected run time: depending on the scanned network sizes, the default scan may take 10mins for 1 numerical replicate.

*## Directed network:*

# 1. optiDirected\_remove\_high2.m

# The script is for selectively removing directional triangle interactions to optimize synchronizability.

# User input parameters: (a) InitialNet: the two types of initialization as described in text; (b) realization: the number of numerical replicates; (c) NumberNode: the number of network nodes;

# Parameters which can be tuned: (d) Run: the number of independent simulated annealing with different randomly initialized network; (e) NumIncreaseTarget: the number of simulated annealing before the target spread of eigenvalues increased by 10%, because the target spread of eigenvalues cannot be too small as it may not be reached due to the sparsity for large networks.

# Expected output: A matlab file containing the spread of eigenvalues, network tensor, generalized Laplacian matrices for the networks before and after optimization, in different sizes.

# Expected run time: depending on the scanned network sizes, the default scan may take 2 hours.

*## Scripts for plotting:*

# 1. PlotStatistics2.m

# The script is for plotting the eigenratios and the generalized degree distribution for the optimized undirected network. It requires to first generate the file from optiUndirected\_rewire\_high2.m, scan sufficient network sizes, and put the file into correct folder location.

# 2. PlotEigenratio2.m

# The script is for plotting the eigenratios for the second-order and the corresponding first-order network.

# Both 1,2 require the files generated from optiUndirected\_rewire\_high2.m and need to specify the parameters matching that in optiUndirected\_rewire\_high2.m to load the corresponding saved files. Please make sure the loaded files are in the correct folder location.

# 3. PlotEigenratio\_Directed.m

# The script is for plotting the violin distribution of the asymmetric measures for the optimized directed network. It requires the files generated from optiDirected\_remove\_high2.m and needs to specify the parameters according to optiDirected\_remove\_high2.m to load the corresponding saved files.