**Logging into the Cluster**

Terminal session login in the cluster

$ ssh veralicona@hpc1.cam.uchc.edu

Username: veralicona

Password: Entry4Me!

**Paths**

cd to the BrainConnectivityNetworks folder. Type paths and press enter. This will run the paths.m script that adds all the folders of MULAN to MATLAB’s path.

**Generating Erdos Renyi Networks**

Random networks are created by the generateDirectedRandom(nodes, p, iter) function in the Helper-scripts folder. It takes as argument the number of nodes, the probability of an edge p, and a number iter that is appended to the end of the structure file created. The structure file that is created is saved to GenerateData/structureN[nodes]directedRandom[iter].mat.

If you want an adjacency matrix directly, you can also call random\_directed\_graph(nodes,p) which is also in the Helper-scripts folder. This is what generateDirectedRandom calls to get the random networks.

**Generating Scale Free Networks**

Scale free networks are created by the generateScaleFree(nodes, iter) function in the Helper-scripts folder. Nodes is the number of nodes and iter is the integer that is appended to the end of the structure file created. The structure file that is created is saved to GenerateData/structureN[nodes]directedRandom[iter].mat.

If you want an adjacency matrix directly, you can also call the BAgraph\_dir(N,mo,m) function in the ScaleFree folder. This is what generateScaleFree(nodes, iter) calls actually. N is the number of nodes, mo is the size of the seed, and m is the number of outgoing edges per node.

**Saving Adjacency Matrix to Text Edge Lists**

Sometimes, you might need to save adjacency matrices to text file edge lists so that you can view the networks in Cytoscape. To do this, type the following commands into matlab:

[r,c] = find(adj);

edges = [r,c];

dlmwrite('myFile.txt', edges);

**Saving Adjacency Matrix to MULAN Structure File**

Have an adjacency matrix saved as a workspace variable in MATLAB and the adj2str.m function saved in the present working directory. In the MATLAB command line, type:

adj2str [name of the matrix] [# of nodes] [# of edges]

Example: adj2str adjmat 50 80

The function will save a MULAN structure file for that adjacency matrix in the present working directory of the format: structureN[# of nodes]L[# of edges].

**ROC Plots**

We modified mln\_MethodStructuresAUC in the MULAN/Evaluation folder to save the Fpr and Tpr values in the AUC file. We also added a function genTopROC.m in the Helper-scripts folder.

genTopROC(filename, number) plots the ROC. Filename is the name of the AUC file under the AUC folder of the results folder. Number is the number of the top methods by AUC value to plot. It saves the the plot as a pdf file in the AUC folder of the results folder.

**Running Cluster Jobs**

To run MULAN on the cluster, you need to modify the jobs.sh file so that it points to the matlab file you want to run. An example is:

matlab -nodisplay -nosplash -nodesktop -r "run('/home/CAM/veralicona/Parameters/start\_parameter\_estimation.m');" >> /home/CAM/veralicona/Paramet\

ers/output.txt

start\_parameter\_estimation.m is the matlab file that gets run. Also, the file after >> is where the output to the console is written.

In the matlab file that runs first, you should first cd to the directory of the code on the cluster. For example, cd /home/CAM/veralicona/sim/. Then add the paths of the MULAN folders, call mln\_generateParams to create the paramter file, compile some c code that Genie uses,

mex ../GENIE/RT/rtree-c/rtenslearn\_c.c;

ext = mexext;

movefile(fullfile(pwd, ['rtenslearn\_c' '.' ext]), fullfile('../GENIE/RT/', ['rtenslearn\_c' '.' ext]));

and then run the simulation.

**MULAN with NetSim data**

All netsim datasets were downloaded from<http://www.fmrib.ox.ac.uk/analysis/netsim/>. Unzip the downloaded folder and save the sim4.mat file to the MULAN directory. The multisub\_sims.m file should be saved in the ClusterComputation directory if run on the cluster with all the filepaths added, but can also be saved to the MULAN directory if run locally. In the command line, with the MULAN directory as the present working directory, type:

multisub\_sims sim4 1 50.

The multisub\_sims function will isolate all 50 subjects’ individual time series, save them in their own directories, then run MULAN with them. After the function is done running, there should be 50 new directories with the format sim4\_Sub[# iteration] saved in the MULAN directory.

**MULAN with Patient data**

Save the text files of all the ADHD patient and nonpatient time series to a directory named “datafolder” in the MULAN directory. The multisub.m and genData.m files should be saved in the ClusterComputation folder if run on the cluster with all the filepaths added, but can also be saved to the MULAN directory if run locally. In the command line, with the MULAN directory as the present working directory, to run MULAN with the ADHD patient and nonpatient type:

multisub ADHD 1 70

multisub CTRL 71 184

After the function is done running, there should be new directories with the format ADHD\_Sub[# subject] and CTRL\_Sub[# subject] saved in the MULAN directory. Note that the multisub function needs to be run twice, once with ADHD and once with CTRL.

**MULAN with Random/Scalefree Networks**

To repeatedly run MULAN on scale free networks on the cluster, you should modify job.sh so that the first matlab file run is start\_repeated\_scaleFree.m. Check that the first line of the file, with the cd command, points to the correct folder.

This script will do all the aforementioned setup work such as adding the paths, and then generate scalefree network structure files and run MULAN on them in parallel.

Note that before starting a simulation, you should check the mln\_generateParams.m file to make sure the correct parameters are used for the simulation.

To run MULAN with random networks point job.sh to start\_repeated\_random.m. Everything else works the same.

You can modify the parfor loop to change the different iterations that are run. The iteration number gets appended to the name of the results folder so you should modify this so that previous runs are not overwritten.

**Genie and Tigress with 6 Different Parameter Sets**

To run repeatedly test Genie and Tigress on the 6 parameter sets with both scale free and random networks, you need to switch to the mbio\_methods branch of the code. Point job.sh to start\_paramter\_estimation.m file. Check that the first line of the file, with the cd command, points to the correct folder. As before, you can modify the parfor loop to change the different iterations that are run. The iteration number gets appended to the name of the results folder so you should modify this so that previous runs are not overwritten.

**Adding Extra Methods to MULAN**

A number of files need to be changed to add extra methods to MULAN. First in the MULAN/ClusterComputation/mln\_generateParams.m file, you need to add the parameters for your new method. You also need to modify mln\_CalEvaN.m in the same folder and add your method to VGroupMethlog.

Next, go to the MULAN/Calculation folder. mln\_setgroupmethlogparam.m needs to be modified so that your method and its parameters are added. Next go to the MulanCal.m file. Modify param\_df so that your method parameters are read from the .param file. Add a case for your method in the switch GroupMethlog. Also, add a file such as mln\_calcMatGenie.m in the Calculation folder that gets called in this switch statement.

In addition, add your method to istimeM.m. This function returns true if your method is not frequency based. Otherwise, it returns false. Frequency based methods need to be treated differently since the matrices that such methods return have an extra dimension.

Modify mln\_Result2file.m as well. You should add a 1 to is3dimemsion. Again, frequency based methods will have a 0 in this list, time based ones will have a 1. This is some what redundant since istimeM already exists, but that’s how MULAN is set up.

Finally, add your method to mln\_issymetricM if it is undirected.

**Adding Accuracy and PPV to MULAN**

We modified MULAN to save ACC and PPV values of methods when evaluating the methods. It saves these extra values in the AUC file, which was previously only used to save the AUC, FPR, and TPR values of methods. To accomplish this, we added a file called mln\_calcFalseRateextended to the Helper-scripts folder. We also modified mln\_MethodStructuresAUC to call mln\_calcFalseRateextended instead of mln\_calcFalseRate which is in the MULAN/Calculation folder. mln\_calcFalseRateextended is basically the same as mln\_calcFalseRate except it also returns ACC and PPV for a method. mln\_MethodStructuresAUC then saves all the metrics for all the methods to the AUC file.

**Analyzing Patient Data**

Once all the directories with MULAN results for the ADHD patient and nonpatient data are saved to the MULAN directory, the multisub\_analyze function can be run. In the command line, with the MULAN directory as the present working directory, to analyze all the ADHD patient and nonpatient data, type:

multisub\_analyze ADHD 1 70

multisub\_analyze CTRL 71 184

Note that the multisub\_analyze function needs to be run twice, once with ADHD and once with CTRL. The analysis files are all saved to a new Analysis directory in the MULAN directory. In the Analysis directory will be directories with the format ADHD\_Sub[# subject]\_analysis and CTRL\_Sub[# subject]\_analysis. Each individual subject’s directory will have four text files: meanMatrix, with an adjacency matrix of the mean value for each edge, minMatrix, with an adjacency matrix of the minimum value for each edge, maxMatrix, with an adjacency matrix of the maximum value for each edge, and AllMethodsAnalysis, which is a list of the mean, min, and max edge value for each method and for all methods. There will additionally be a directory called AllMatrices which has the adjacency matrices from each method. Finally, there will be a files in the Analysis directory called ADHD\_CombinedAnalysis and CTRL\_CombinedAnalysis, which will be mean, min, and max values for all methods from each ADHD subject or CTRL subject.

**Finding top methods**

We added a function called generateStatistics(path) in the Helper-scripts folder. This function goes through every results folder located at path and calls mln\_MethodStructuresAUC(dirname, prenom), which evaluates the methods using the gold standard network and generates the AUC file which includes the AUC values, the TPR and FPR values, as well as PPV and ACC values.

Thus, first call generateStatistics on the folder containing all of your simulations for ScaleFree or Random networks to ensure that ACC and PPV values are present in the AUC file.

Then, call topMethods(path), which is also located in the Helper-scripts folder. This function will look at all the simulations in the path folder and print to the console the top methods overall.

**Creating consensus networks**

To create consensus networks, as before, first call generateStatistics on the folder containing all the results folders you want to analyze. Then call consensusNetworks(path) on this folder. This function, which is present in Helper-scripts will create a new file in the ToutResults folder which includes all the consensus networks. Finally, call topMethodsExtended(path) which will print to the console the top methods overall, including the community-based methods.

**Analyzing Parameter Results**

To analyze the results of repeated runs of Genie and Tigress on 6 sets of parameters, call analyzeParameters(path, folderName, number). Path is where the results folders are located. The results folders of the different iterations should have the following format: folderName[iteration #]\_params[param set #]. Thus, the number argument for analyzeParameters is the maximum iteration number.

**Not running the Granger family of methods in MULAN**

\*Requires manually changing a script

Open the mln\_calEvaN.m file in the ClusterComputation folder. Delete ‘Granger’ from VGroupMethlog, which should be at line 19. Type it back in to put it back.

**Only analyzing patient data results from the top 5 methods**

\*Requires manually changing a script

Open the multisub\_analyze.m file. There should be a switch statement starting at line 35. At line 35, change “for i=1:41” to “for i=1:5”. Next, delete every case in that switch statement but cases 17, 19, 37, 38, and 39, which should correspond to BCohF, Genie, Tigress, BCorrD, and BCorrU respectively, the top 5 methods for scale-free networks. Rename those cases 1 to 5, so case 17 becomes 1, case 19 becomes 2, and so on. Finally, replace all instances of the directory name “Analysis” to “Analysis\_top5”. Replacing should be case-sensitive. Now, repeat the same steps in Analyzing Patient Data to get the analysis from only the top 5 methods, where all of those results will be saved into a folder called “Analysis\_top5”.