SOP for Adaptive Graphical Lasso:

“dataNeededforSim\_MEG.mat” must have the following variables for code to work:

* Lead field L (single sources – not dipoles): channels x sources
* SC: (sources+subcortROIs) x sources (sources+subcortROIs)
* Labels: sources x 1 (values of the sources in SC) – this was originally the vertices of the Brain where each vertex is marked by the source value, then this could be used to identify which were the correct ROIs to select from the SC to avoid the subcortical ROIs.
* subcortROIs: num subCortROIs x 1 (locations of the subCortROIs in SC)

Section 1:

1. Start from the “MEG\_SC\_sim.m”:  
   This script will generate the precision matrices needed and the test the adaptive graphical lasso.
2. This will first run “allInverses\_MEG\_dip.m”. This script will take the leadField and apply the correct weighting structure across the dipole to reduce them to a single source at each ROI.
   1. The first section of “allInverses\_MEG\_dip.m” (from lines 6-22) is unnecessary if you don’t have a dipole fit. Edit this part out or use the script: runWeightedL2norm.m
   2. This script is useful because it establishes a few necessary variables for the following script and also this only needs to be run once. Currently the simulation re-runs the inverse solution creation but this is unnecessary.
3. Run “genCov\_CMVN\_SC.m”. This function will use the SC to create a new precision and covariance. This will get re-run within each run of the loop of the simulation however it is run before just to establish some variables.
   1. Note that the SC required by this function currently assumed subcortical ROIs are also present. If they are not then this function needs to be changed so that the expectation of subcortical ROIs is removed. This means the useAreas variable becomes unnecessary and it needs to be removed from places it is used or it needs to be altered appropriately (for e.g. if 443 sources present it could be set to 1:443).

At the end of the section three critical variables are set:

* GforFit: This is the network used to test which edges are truly within the SC – for when correlations are calculated. This currently directly uses the SC, if you cut some edges in the SC and want to use the original SC for the metrics used to test AGL results then the creation of GforFit will need to be altered.
* allLambdas: The set of penalization values that is passed through by the AGL for the grid search of cross-validation. This interacts strongly with the value of the off-diagonal value used to scale it for the penalization (which is set on line 60 of estBestPenalizationQUIC.m)
* allLambdasOut: The set of penalizations applied outside the connectome.

Section 2:

Establish several necessary variables:

* How many samples (samps) are being drawn? If there are 1000 edges then at least 2000 samples tends to be needed for full recovery.
* What is the SNR of noise? Currently set to 25.
* What is the penalization used for the weighted L2 norm inverse?

1. For loop will run over the allInverses\_MEG\_dip script though this is not necessary for every run. Can be removed.
2. The “genCov\_CMVN\_SC.m” is necessary for every run as it establishes a new precision matrix based on the SC.
3. The “mvnrnd” function will sample from the “genCov\_CMVN\_SC.m” as real values.
4. Currently noise is added at the source level. Alternatively, it could be set up to be added at the channels. I found the performance was worse in that case but is worth testing.
5. estBestPenalizationQUIC.m runs the adaptive graphical lasso algorithm. The SC that is passed here should be what is intended to structure the penalization. IF the intent is to penalize based on the damaged connectome then that will need to be passed instead of full SC.