NKI+Controllability Code Explanations

Michael D Hess
Claremont McKenna College

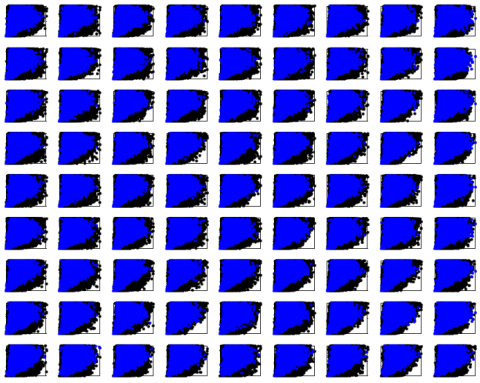
fcn

optimalControlContinuous.m

The entire project hinges around the optimalControlContinuous function, which was not a piece of code that I created, but one that Dr. Betzel gave to me as a starting point for this project. The code takes in a structural connectivity matrix, an initial state, a target state, a rho value (which we set to 100), and a length of time for the transition to occur. The function returns the trajectories and input energies for each node over the specified length of time.

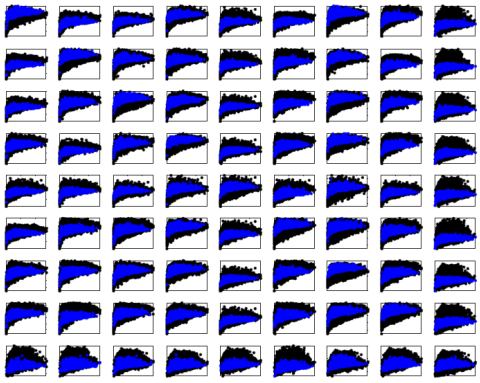
figs

degree_figures.png



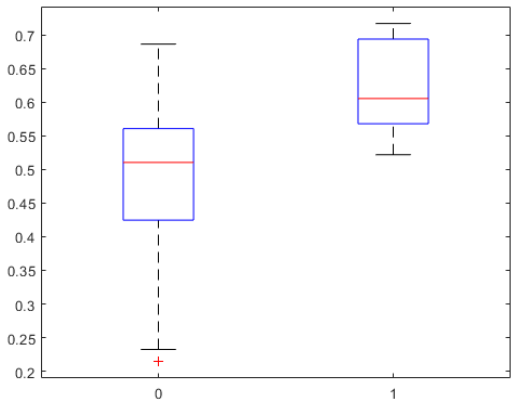
Here, we are looking at the relationship between average energy and degree, where the blue is the scrambled, but degree preserving nets and the black is the real nets.

degree_figures_not_ranked.png



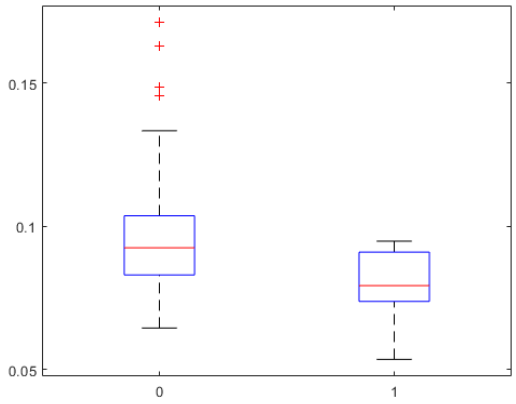
This is the same as the above figure, but instead of using the ranked energies, we use the log-transformed energies.

rich_club_avg_energy.png



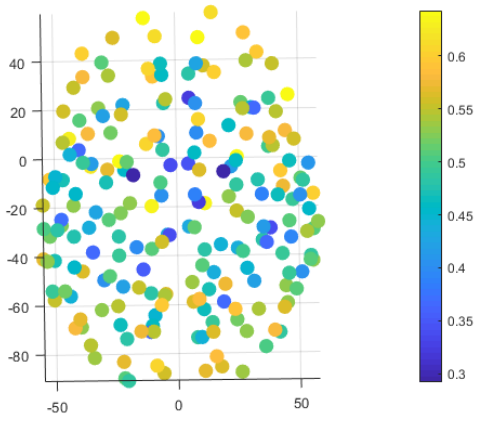
Here we are looking at the average energies of nodes that are in the rich club (high degree, 1 in this plot) versus those that are not in the rich club (0 in this plot).

rich_club_std.png



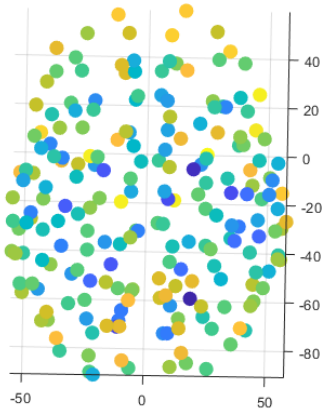
A comparison of the average standard deviation of energies for rich club and non-rich club nodes.

scrambled_nets_avg_energy_on_brain.png



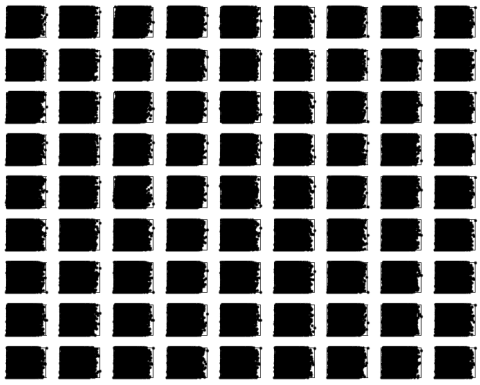
This is the average over subjects of the nodeAvgGenerated.mat file (which is a node-level average of energies averaged across all subjects) plotted in 3-space. (This is a top down view of the brain the nodes form!!)

real_nets_avg_energy_on_brain.png



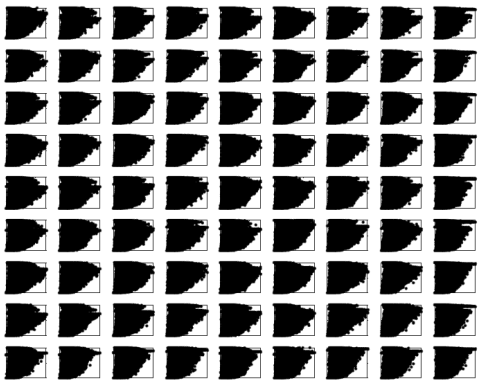
The same as the above figure, but with the real networks' energies, not the scramble nets' energies

degree_figures_all_transitions_SD.png



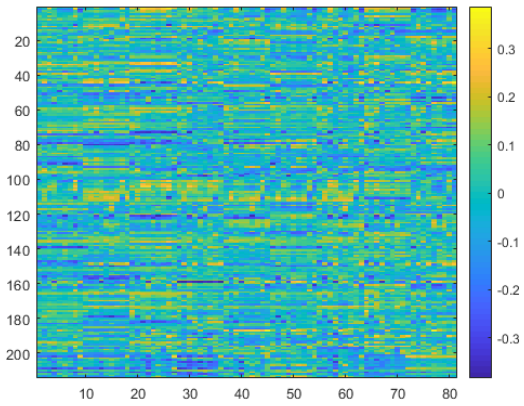
The relationship between degree and energy when we use the selected drivers as nodes.

degree_figures_all_transitions_AD.png



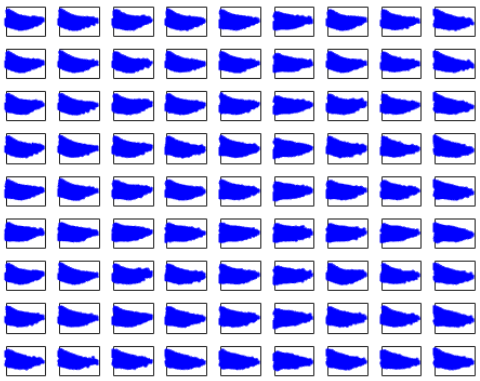
The relationship between degree and energy when we use the all drivers as nodes.

correlation_residuals_and_age.png



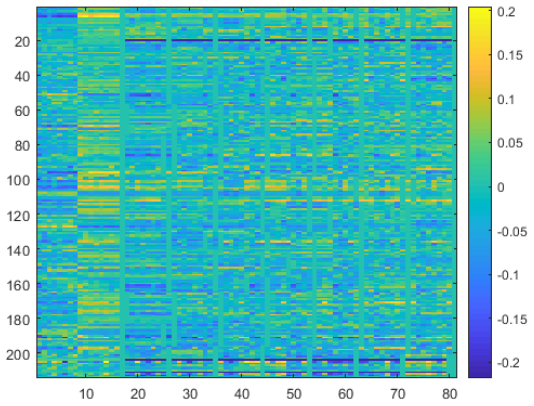
A 214x81 correlation heatmap showing the correlation of each residual node energy (using polynomial fitting and correction) for each type of transition with age

degree_vs_resisuals.png



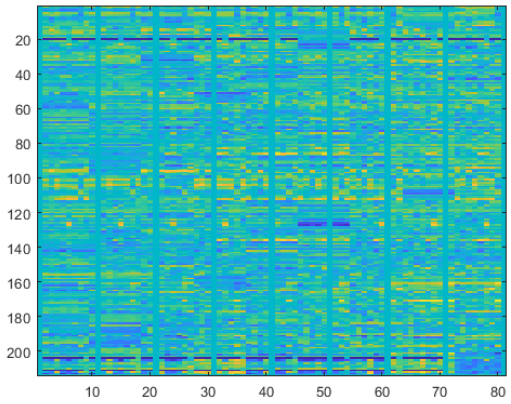
The relationship between degree and residual control energies!

sorted_correlations_fsiq_energy.png



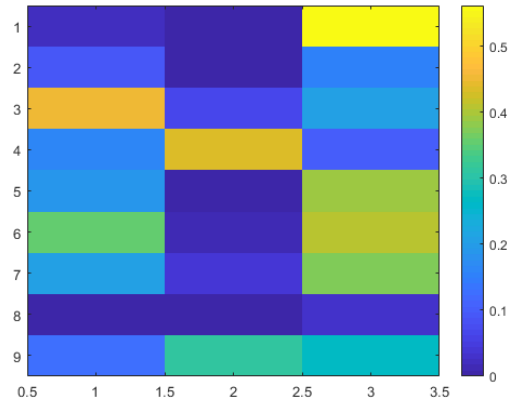
I am unclear about what the sorting does, but this figure and the following are correlation heatmaps of the corrected node energies with intelligence scores (wasi full scale IQ)

unsorted_correlations_fsiq_energy.png



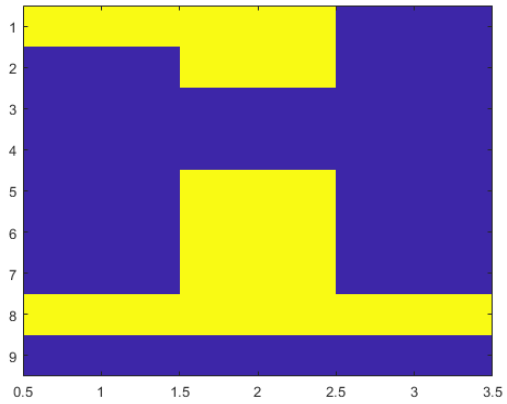
I am unclear about what the sorting does, but this figure and the preor are correlation heatmaps of the corrected node energies with intelligence scores (wasi full scale IQ)

p_values_for_correlations.png



This is a heatmap showing the “p-values” that we calculated using bootstrapping of random nodes’ energies and their correlations with intelligence versus our true correlations.

p_values_for_correlation_significant.png



The same figure as above, but with the mask that yellow is for “p-values” less than or equal to 0.05!

m

Code and programs that Dr. Betzel and I have worked on through this project

control_attempt1.m

This code redefines the regions from 17 regions to just 9, then computes the optimal control energy using all nodes except target and initial nodes as drivers

net_scrambler.m

This program subsets a small group of the total participants (for this I chose age between 25 and 30 ffor simplicity, but a random sample would be better!) It uses these subjects’ structural connectivity matrix to randomly create 100 new networks that preserve degree for all nodes. Then, it computes the control energies for each node for each network for each subject

analysis_attempt1.m

In this code, we loop through all nodes, divide each node's log energies by the number of nodes used as drivers and then calculate the average energy for each node for each subject and store it in nodeAvg

analysis_generated_nets.m

This code works just like analysis_attempt1.m, but loops through all 100 randomly-scrambled, degree-preserving nets instead of the real nets.

degree_analysis.m

This code was just for us to compare the differences in energies between degree, age, and node!

degree_analysis_all_transitions.m

Basically works in the same way as analysis_attempt1.m, but calculates the average energy for each transition

degree_analysis_all_transitions_not_ranked.m

Does the same as above, but does not used ranked energies, but the energies themselves

mean_std_rich_club.m

This code finds the rich club nodes (those with the highest degree) and compares their average energy to all non-rich-club members.

rich_club_bu.m

This is a piece of code from the Brain connectivity toolbox

mean_std_rich_club_scrambled_nets.m

Does the same analysis as mean_std_rich_club.m but on the scrambled nets we generated!

control_selected_networks_and_all_networks.m

This code computes the control energies using selected drivers (regions 1,3,5) and it also computes the control energies using all drivers (including targets and initials)

analysis_SD_AD.m

This code extends our earlier nodeAvg analyses to the selected driver (SD) and all driver (AD) approaches

degree_analysis_all_transitions_SD.m

This extends our degree_analysis_all_transitions.m code to the selected drivers control energies

degree_analysis_all_transitions_AD.m

This extends our degree_analysis_all_transitions.m code to the all-drivers control energies

control_all_drivers_scrambled_nets.m

Using our scrambled nets with all nodes acting as drivers

analysis_AD_residual_removing.m

In this code, we loop through the subjects and transitions, and for each, we fit a polynomial model of degree to control energies, then subtract the model's values from our control energies to get residuals

correlation_analysis_res.m

Here, we are looking to see if there is a correlation between our node-level control energies and degree (to see if our correction worked) and age.

analysis_AD_residual_removing_mean_degree.m

In this code, we used a nonparametric approximation of energy as a function of degree to perform our correction, then perform our correlation analyses

strength_correction.m

This code attempts to perform a correction based on the strength (instead of degree) of the nodes.

comparison_of_degree_removing.m

Here, we perform a correction based on whether the node was a target, initial, or bulk node, then compare to our original way

class_correction.m

This code uses the class correction mentioned above, but compares across strength and degree corrections.

intelligence_age_energy_correlation.m

Computes a correlation matrix for each node and transition and intelligence on the wasi full scale IQ test, then does a random sampling of node energies, calculates a correlation, and reports 1 if the "random" correlation is greater than our observed correlation and 0 otherwise. Used to emulate a p-value for our correlations.

age_energy_correlation.m

This code does the same as the above, but instead of looking at correlations with energies and intelligence, it looks for correlations with age.

age_raw_energy_correlation.m

Does the same analysis as above, but uses the raw, uncorrected energy values instead.

mat

Contains the original data structures used in this project

age.mat

A 663x1 double containing the age of each of the subjects

nki_mats.mat

FC

A 214x214x663 double containing the 214x214 functional connectivity matrix for each of the 663 subjects

SC

A 214x214x663 double containing the 214x214 structural connectivity matrix for each of the 663 subjects

ci

A 214x1 double containing the region designation for each node. To determine which region this is, use the net variable stored in this file.

coor

A 214x3 double containing the 3d locations of each of the regions. This can be used in a scatter plot to plot 214x1 variables on the brain. The first, second, and third columns represent the X,Y,and Z coordiantes of nodes, respectively.

net

A 17x1 cell containing the names for each of the original 17 regions. The values from this variable are used in the ci variable stored in this file to assign each node to its region.

outputSubs

A 663x1 cell containing the subject name that is used in the phenoDf.csv for each of the 663 subjects used in this project. This can be used to obtain the various intelligence measures for each individual in the study.

phenoDf.csv

A csv containing the subject name, ID, age, sex, native language, diagnoses (if applicable), and intelligence scores for 954 subjects, only 663 of whom have been selected for this study.

Schaefer2018_200Parcels_17Networks_order_FSLMNI152_2mm.nii

A nifti file containing the parcels for each of the brain networks regions

output

Derivatives of the data in the mat folder and generated in this study

ciprime.mat

A 214x1 double that is a consolidation of similar groups from the original ci.mat file. Instead of having 17 distinct systems, similar systems were aggregated, resulting in 9 final regions.

energy.mat

The output from the control_attempt1.m file. This is a 214x9x9x663 double containing the control energy for each node over all transitions for every subject.

nodeAvg.mat

A 214x663 double containing the nanmean of the log-transformed control energies for each node over all transitions. Created in analysis_AD_SD.m.

scrambled_nets.mat

A 214x214x37x100 double containing 100 randomly-scrambled, degree-preserving structural connectivity matrices for 37 subjects. Created in net_scrambler.m. Since degree is such a large component of control energy, these nets are used to compare to the real networks.

energyGenerated.mat

A 214x9x9x37x100 double containing the control energies for each of the scrambled nets in scrambled_nets.mat for all transitions and the 37 subjects used for those nets.

nodeAvgGenerated.mat

A 214x37x100 double containing the average log-transoformed control energies for each node over all transitions.

energySD.mat

A 214x9x9x663 double containing the raw control energies for each node for each transition for each subject using selected nodes as drivers

energyAD.mat

A 214x9x9x663 double containing the raw control energies for each node for each transition for each subject using all nodes as drivers

nodeAvgSD_AD.mat

Contains the node-level average energies for the energyAD and enerySD structures, and then also contains the average across all subjects (NodeAvgAvg)

nodeAvgResAD.mat

A 214x663x9x9 double containing the residual energies after correction with a polynomial model

nodeAvgADCorrected.mat

Contains four doubles, each of size 214x663x9x9 containing the average energies for each subject, node, and transition of the degree corrected energies

PhenoDfStuff.mat

index

Used for converting between the index and the dataframe containing all of the subjects (not just those listed here) – these are the indices of the subjects we are looking at

names

Used for converting between the index and the dataframe containing all of the subjects (not just those listed here) – these are the names of the subjects we are looking at

wasi_fsiq_prcntile

A 663x1 double containing scores on the Wasi full scale IQ test

wasi_pri_prcntile

A 663x1 double containing scores on the Wasi PRI IQ test

wasi_vci_prcntile

A 663x1 double containing scores on the Wasi VCI IQ test

wiat_composite

A 663x1 double containing the composite score from the WIAT intelligence test

idx_class.mat

A 214x9x9 double containing the designation of which transition "class" each node belongs to for each transition. A class of 1 means that the node was initially active; a 2 means that the node was activated at the end of the control task; a 3 means that the node was neither initially activated nor ultimately activated and merely acted as a passage for the inputted control energy.