Network Test Refactor
Release Notes

I. Motivation
   1. Remove accumulation calculation for the end result.
   2. Along with this, preserve each permutation calculation result.
   3. Calculate p-value ranking by traditional method.
   4. Consolidate results into single model.
   5. Move plotting out of results model(s) into separate model(s)
   6. Rename variables to human-readable names

II. Effected Users
   1. Everyone that uses Network Level Analyzer.
   2. Even script users

III. Changes
   1. Eliminate many result models

Previously, results for network-level tests were broken down in a large hierarchical structure. There was a general permutation result, then broken down by input type, then a Cohen's D only, then individual tests that branched off the input tests. While there were common fields between these models (perm_rank, perm_prob, etc.) there were also fields that were not (t, z, chi2, etc). This made editing code that did not directly affect the individual results difficult. Trying to write code that could be used by all of them was extremely difficult.

Example, trying to modify TestPool to keep all the permutation results necessitated creating many different result models and case checks to determine which model had which fields.

   2. New result model

The new result model is the same for each different network test regardless of the type of input it receives or what type of test is run. This allows for objects and code that does not care about the type of test (like the TestPool or plotting) was run. Some detail about the new result:

- Fields:
    - test_name – name of the test run
    - test_name_display – display name for GUI
    - test_options – options for the test
        - Full_connectome, within network pair?
        - Plot specs
        - Previously known as "input_struct"
    - full_connectome – final full connectome results

- within_network_pair – final within network pair results
- no_permutations – final no permutation result (the "observed")
- permutation_results – results for each individual permutation

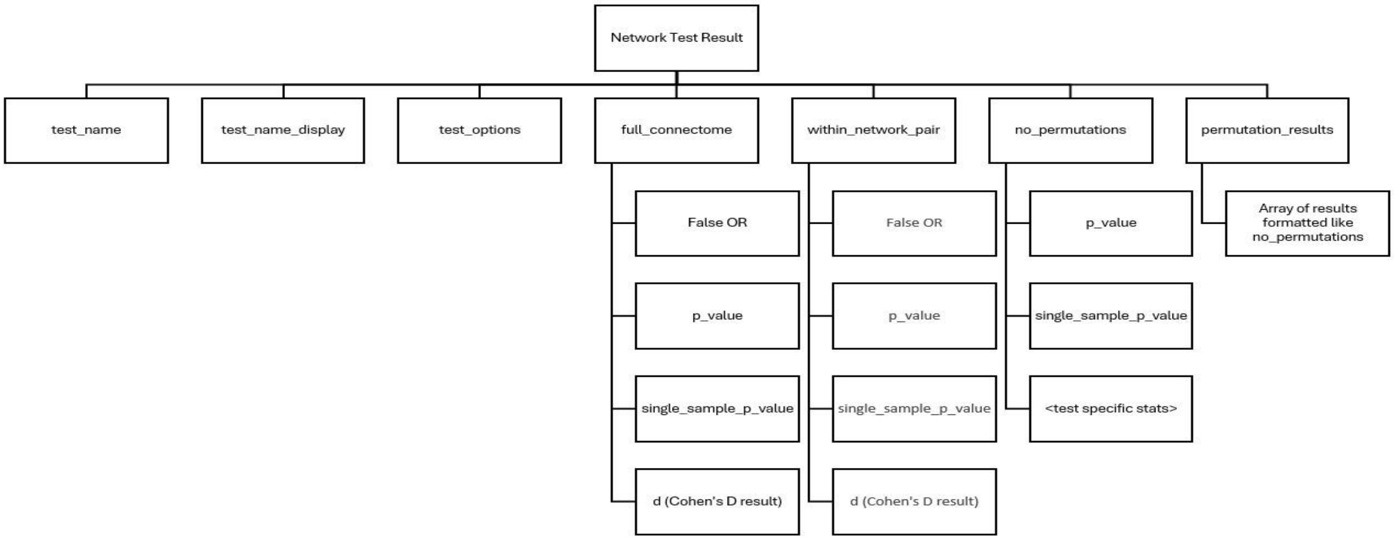
The name and display name are self-explanatory and tell what type of test was run.

The test options describe all the specifications for the tests and if there were special options or changes

The permutation results field is an array of TriMatrix. One for each permutation. These are what are used to calculate the final results (p-values). Holds statistics specific to each test along with p-value for each permutation.

The three final results (no_permutations, full_connectome, within_network_pair):

- Initially set to "false"
- If a test was not run, it will remain "false".
- Matlab is much better and much easier at testing for false than testing for blanks or empty data types.
- Each will have "p_value" and "single_sample_p_value" and a Cohen's D result



3. Ranking

Ranking used to be done on each iteration. This was done by adding or subtracting depending on the value of the observed and the permutation result. This is not the standard way of calculating the p-value for permutation testing. The change has been made to return to the standard method. We create a distribution of values for the test. Then we just count the number of results that are greater than our observed value (sometimes we count the less than results, too). This can all be done by sorting the the results and finding the index of the observed value.

Since this is now dependent on all the results, and keeping with object-oriented principles, ranking/calculating the result is a separate object and call.

The object is ResultRank(). It needs to be initialized with:

- nonpermuted_network_result (single test)
- permuted_network_result (single test)
- number_of_network_pairs

Then call the rank() method. All of this can be seen in the TestPool.

4. Plotting

This is for the GUI users. The large plots that are created by the GUI have a good amount of calculations. These have been moved to their own models.

The parameters for plotting is done in NetworkResultPlotParameter.

These parameters are then used with NoPermutationPlotter, FullConnectomePlotter, and WithinNetworkPairPlotter.

IV. New file names and locations

- +nla/+net/+result/NetworkTestResult
- +nla/+net/+result/NetworkTestResultPlotParameter
- +nla/+net/+result/+plot/NoPermutationPlotter
- +nla/+net/+result/+plot/FullConnectomePlotter
- +nla/+net/+result/+plot/WithinNetworkPairPlotter
- +nla/+net/+result/+chord/ChordPlotter
- +nla/+net/ResultRank
- +nla/+net/CohenD (this is moved out of tests)
- +nla/+net/+test/ChiSquaredTest
- +nla/+net/+test/HyperGeometricTest
- +nla/+net/+test/KolmogorovSmirnovTest
- +nla/+net/+test/StudentTTest
- +nla/+net/+test/WelchTTest

- +nla/+net/+test/WilcoxonTest