SIMPLEX JHU
Q4 2016 Report

Neurodata Team

Contents

1 Tools

One core goal contained within SIMPLEX is the creation of tools for processing, storing, and analyzing large scale neuroscience data. Here, we describe progress made for which new tools or manuscripts exist, each performing either some novel task on our data, or performing an existing task in a novel/scalable way in a space where a feasible solution did not exist. This ranges from backend development on our spatial database, image analysis pipelines, and statistical methodologies for performing inferences upon our data.

1.1 ndstore

We continue to enable authentication for ndstore and converting all RESTful calls to HTTPS from HTTP. We have two different ways to authenticate to ndstore for different use-cases. First, a single persistent token for ndio users who should authenticate for every RESTful call. This token can be downloaded from the management console as a secret token and saved locally by the ndio user. Second, ephemeral session tokens for third-party applications which can authenticate users via a login screen. Third-party applications forward these credentials to the authentication server for session based tokens. The use case for this are applications are *ndviz*, *ndtilecache*, *CATMAID*, etc. This feature is under active development and is undergoing integration and testing. We also added continuous integration for automated testing with Travis–CI. This enables running our tests automatically for every commit.

1.1.1 Filtered Metadata Tiles

A new tile interface was added to support filtered metadata tiles. Previously, when the user wanted to visualize an annotation project, all annotations were rendered as different colors based on ID. The filtered metadata tiles service allows the user to specify which annotation IDs to include when the tile is rendered. Using the service, combined with the RAMON Metadata services described in ndramondb below, a user can dynamically filter for specific IDs (e.g. all synapses), resulting in a cleaner and simpler view.

1.1.2 ndingest

Ingest is required to get data into our infrastructure. Our existing ingest capabilities are limited by our hardware, which lacks in arbitrary scalability, both for a single user and multiple simultaneous users. We therefore are building *ndingest*, a submodule which supports interactions with the AWS cloud services specific to neurodata. This will be primarily used by the auto-ingest service to enable parallel ingest to AWS S3 buckets. The auto-ingest service has now been redesigned to use AWS lambda in conjunction with SQS, S3 and DynamoDB.

1.2 ndramondb

Several new RAMON metadata queries were added to the ndstore RESTful interfaces; these include get bounding box, query by key, and top keys. The get bounding box query takes a RAMON metadata object ID and a base resolution as arguments, and retrieves the three-dimensional bounding box in voxel space for the specified object. Users can retrieve the extent of a RAMON backed annotation and easily locate the object in 3D space.

Query by key allows a user to filter available RAMON objects by specifying a key-value pair. For example, all RAMON objects of a specific type can be queried by specifying "ann_type" as the key and an integer identifying the annotation type as the value. The query returns a list of RAMON objects, which can be further queried for more specific information (e.g. using the bounding box query above).

The RAMON metadata standard is designed for arbitrary key/value combinations. The top keys query allows a user to get the top K keys in a database, where K is a user supplied parameter. The results from the top key query can be used to inform a call to query by key, allowing a user to explore both the RAMON metadata in a database and the available information encapsulated within each RAMON object.

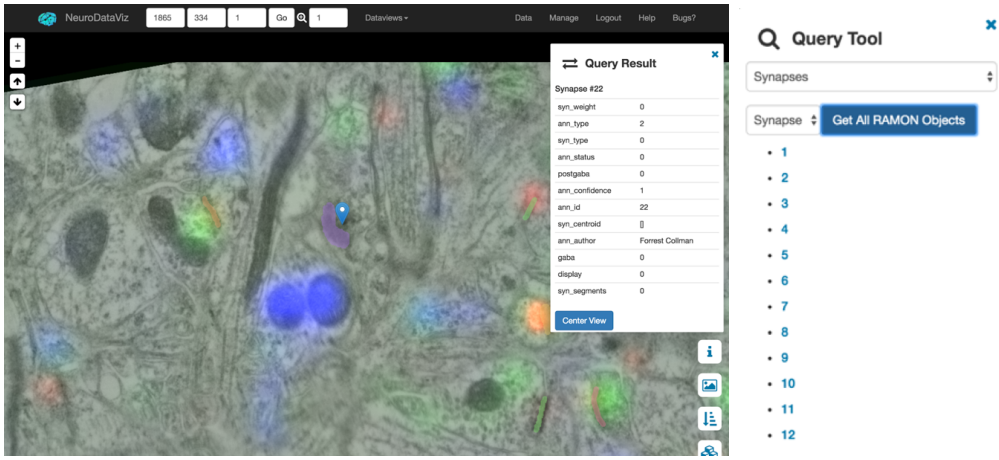
1.3 ndviz

Several small ndviz interface adjustments were introduced. First and foremost, a scale bar was added to the viewer screen. Additional visual tweaks and bug fixes were made, and a new beta release was published[1].

The ndviz backend system was re-engineered to incorporate the React javascript library. React speeds up dynamic javascript client code by only making necessary adjustments to the Document Object Model (DOM). Updating the DOM is a slow process. To speed updating, React builds a virtual DOM. With each change, React checkpoints the virtual DOM, builds a new virtual DOM, and compares the two. Based on the comparison, changes are applied to the real DOM. Although this process is more complex, building the virtual DOM is much faster than making wide-ranging updates to the real DOM.

React also decouples the ndivz viewer state from form fields and user controls. For example, the state of the opacity sliders is now set by backend javascript code, and the sliders are updated on each open, or on user interaction. By handling application in backend code, the user facing state is always consistent. And, the groundwork is now in place to save the complete application state for each user session, allowing a user to set a number of image processing parameters and resuming their session with all parameters restored at a later time.

Finally, both the bounding box and the query by key ndramondb queries were added to ndviz. Ndviz can now center on an arbitrary annotation, as well as list all annotations by type for a given project (see Figure 1).



(a) Array Tomography with conjugate Electron Microscopy data in ndviz.

(b) Query Tool controls box.

Figure 1: ndviz displaying an annotation project with Synapse 22 for this particular dataset (centered blue marker). Clicking an ID will open the Query Tool controls box, shown at right, and allow the user to center the current view on the selected annotation. The figure was obtained by loading the dataset[2], selecting Synapse 22 in the Query Tool box (magnifying glass, picture below), and clicking the blue "Center View" button.

[1] http://viz.neurodata.io

[1] `http://synaptomes.neurodata.io/ndv/project/collman15_annotations/xy/1/1590/1069/0/#`

1.4 Graph Explorer

The graph explorer, shown in Figure 2 now has new ability to view statistics conditional on the attributes in the graph. The conditional distributions illustrate the similarity and distinction among the different vertices in the network data. Difference among these distributions are potentially linked to their different functions in physiology.

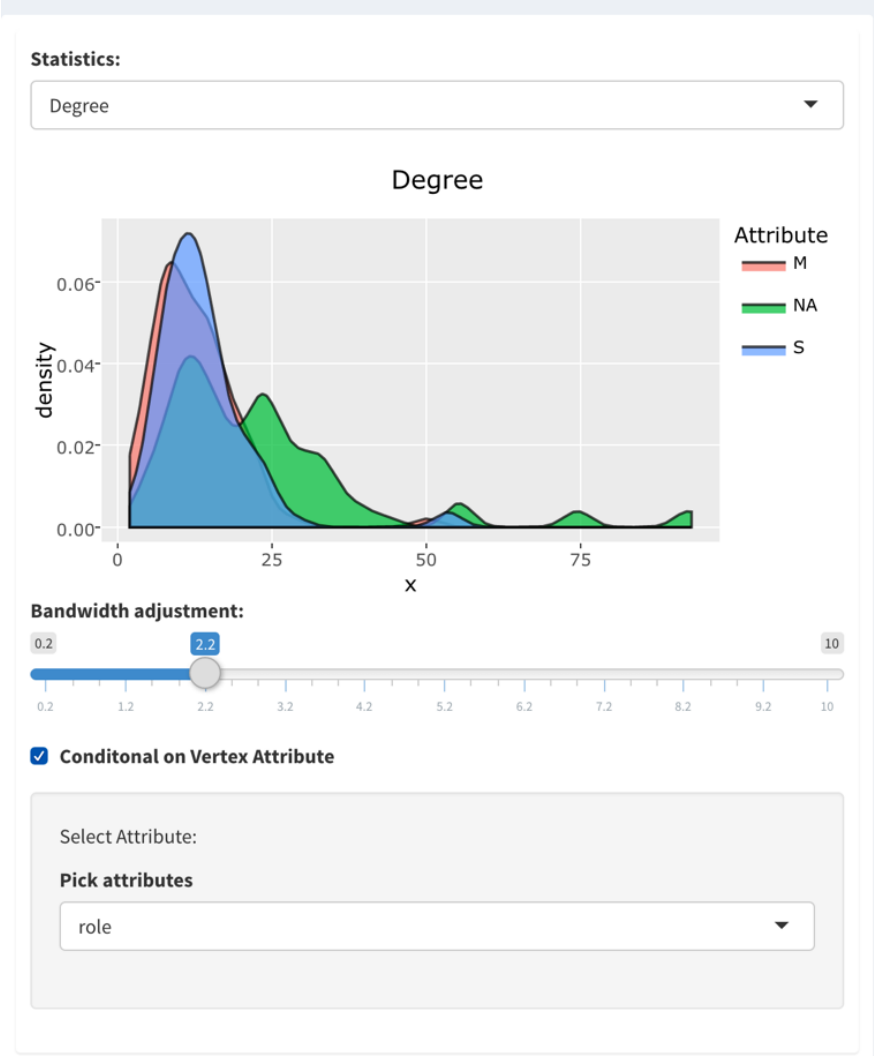


Figure 2: Graph explorer view illustrating different degree distributions for graphs with different attributes

1.5 ndreg

The Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm is an image registration method used to compute a smooth invertible transform ϕ_{10} which aligns template image I_0 to target image I_1. Traditionally LDDMM searches for a transform ϕ_{10} which minimizes the Mean Square Error (MSE) (sum of the differences) between the transformed template $I_0(\phi_{10})$ and target images. This approach works well when I_1 and I_0 have the same intensity profile (dark areas match dark areas) but fail when the images have very different intensity profiles (dark areas match light areas). Figure 3 depicts an image pair with different intensity profiles.

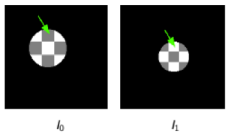


Figure 3: Example image density profiles.

To align images with differing intensity profiles, LDDMM code which matched images using histogram Mutual Information (MI) instead of MSE was developed. Figure 4 below compares the registration results using the MSE matching (1st and 2nd columns) to MI matching (3rd and 4th columns). In the MSE experiments (1st and 2nd columns) registering images of differing intensity profiles failed as indicated by the unnaturally deformed mapping ϕ_{10}. Furthermore the Power Spectra of the displacements $P(\phi_{10} - id)$ shown in the Fourier Domain have significant high frequency components. These components indicated by the large values far from the origin show that ϕ_{10} is not smooth. The MI results (3rd and 4th columns) were much better. Mappings were smooth and were primarily composed of low frequency components.

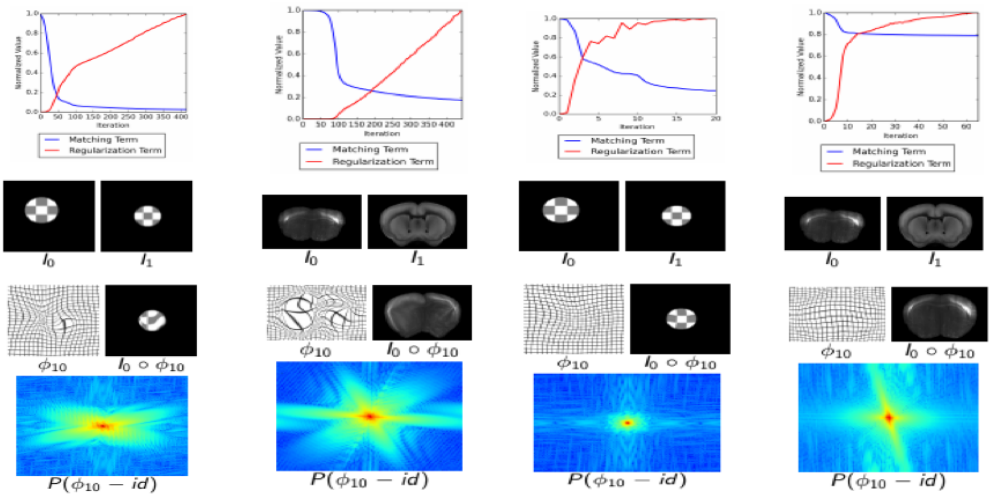


Figure 4: Example fieldmaps for aligning template images as well as a sample of the CLARITY brains aligned to the ARA.

1.6 FlashX

Sparse matrix multiplication in FlashX uses its own format (SCSR) for sparse matrices to improve performance. We investigated the overhead of using customized format to thoroughly evaluate the performance of sparse matrix multiplication in FlashX. Thus, it is essential to accelerate the format conversion from a standard format such as CSR to our customized format SCSR. Figure 5, below, shows the benefit of using the SCSR format including the overhead of format conversion. When an application requires more than 4 SpMVs, converting the format of a sparse matrix improves the performance of the application. Thus, majority of the applications benefit from the format conversion.

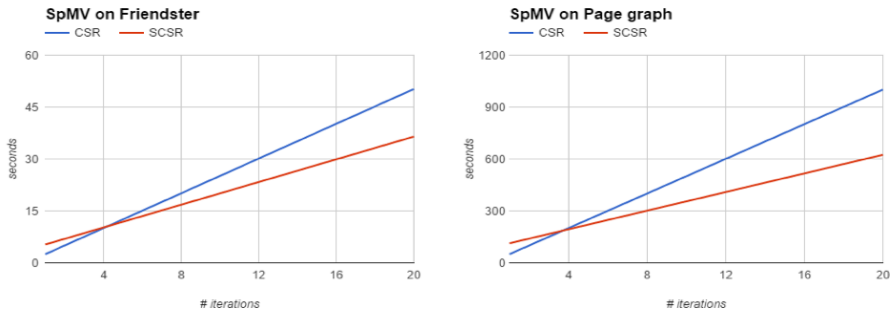


Figure 5: Format conversion to FlashX format is beneficial for large graphs.

The performance ratio of the in-memory and semi-external memory sparse matrix multiplication in FlashX is affected by multiple factors. Shown in Figure 6, we demonstrate some of the factors with SBM graphs with the same number of vertices and edges. We vary the number of clusters and the number of edges inside clusters. We measure the performance of SpMV on both clustered and unclustered graphs. When vertices are ordered based on the cluster structure, more clusters and more edges inside clusters increase CPU cache hits, which leads to less computation overhead and larger performance gap between in-memory and semi-external memory executions. However, if vertices are ordered randomly, these two factors have less obvious impact on performance.

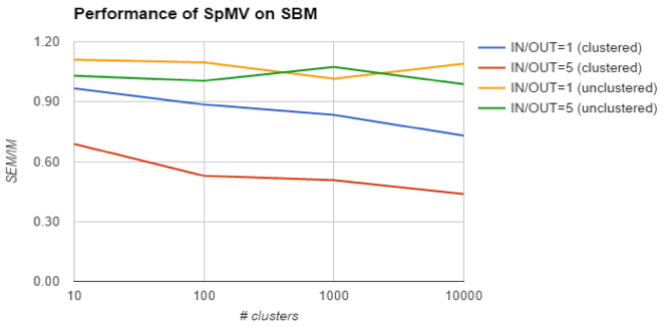


Figure 6: Example use-case of graph operations using SpMV on clustered and unclustered graphs.

1.7 FlashY

FlashY is created to encapsulate the R packages created for multiple-graph processing and clustering.

When multiple graphs are involved, the dimensionality problem becomes more challenging – since each graph is a matrix, stacking multiple graphs leads to a 3-dimensional array referred as tensor. There has been little research in handling this type of data in the network data domain. A dimension reduction toolbox, to be used in R, has been created to convert each adjacency matrix into a small vector, while retaining most of the variability across subjects. We tested, as show in Figure 7, the classification performance using sex as outcome labels and brain network as input, the low dimensional vector produced indistinguishably good performance as the large network data.

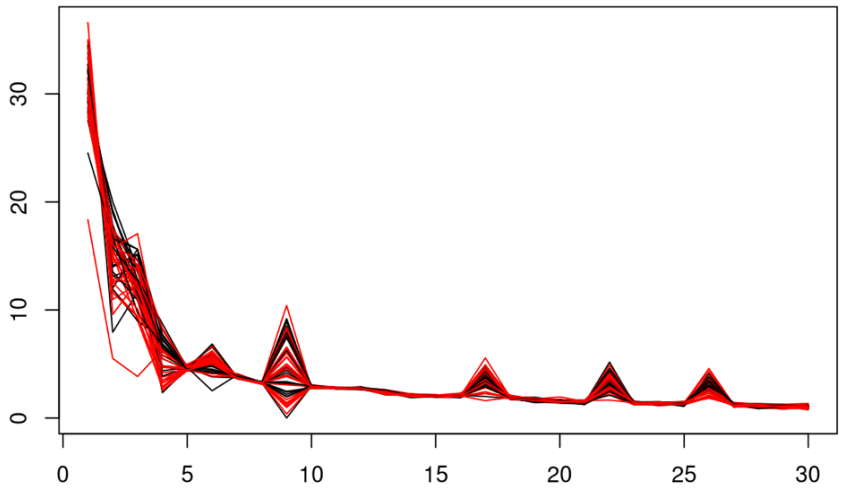


Figure 7: View of the low-dimensional vectors from 46 subjects, dimension is reduced from 4,900 to only 30 for each subject.

1.8 m2g

1.8.1 Processed Data

Using the ndmg and CPAC pipelines, both structural and functional connectomes have been estimated from all known publicly re-distributable datasets. These graphs have been generated across multiple brain parcellations/anatomical atlases[3], ranging in scale from approximately 50 nodes, up through a voxelwise parcellation of nearly 2 million nodes. These graphs can be found through our website[4], and downloaded for analysis. Table 1 summarizes all of the datasets processed, while Figure 8 summarizes the scales upon which each the graphs are generated.

Dataset	Subjects	Scans Per Subject	Total Scans Processed
KKI2009	21	2	42
NKI-ENH	198	1	198
NKITRT	24	2	40
MRN114	114	1	114
MRN1313	1313	1	1307
JUNG2015	255	1	254
SWU4	235	2	454
BNU1	57	2	114
BNU3	48	1	48
HNU1	30	10	300
Total	**2295**		**2871**

Table 1: Publicly available and redistributable fMIR+DWI+MPRAGE datasets.

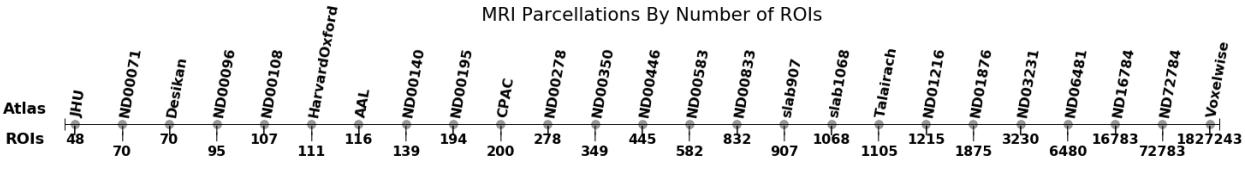


Figure 8: Atlas scales

1.8.2 Batch Effects in MRI

The large database of processed graphs from MR subjects enables us the opportunity to explore mega-analyses in MRI data unlike that which has been done previously. The exemplar task we chose to illustrate this capability was exploring the significance of batch effects in data collected in different studies. The following plot, Figure 9, shows histograms of B0 image volume intensities from various datasets. We can see slight differences in raw data

[3]http://docs.neurodata.io/nddocs/mrgraphs/atlases.html
[4]http://docs.neurodata.io/nddocs/mrgraphs/processed_data.html

through this metric, including intensity shifts either left or right by a dataset, or a dataset containing multiple populations.

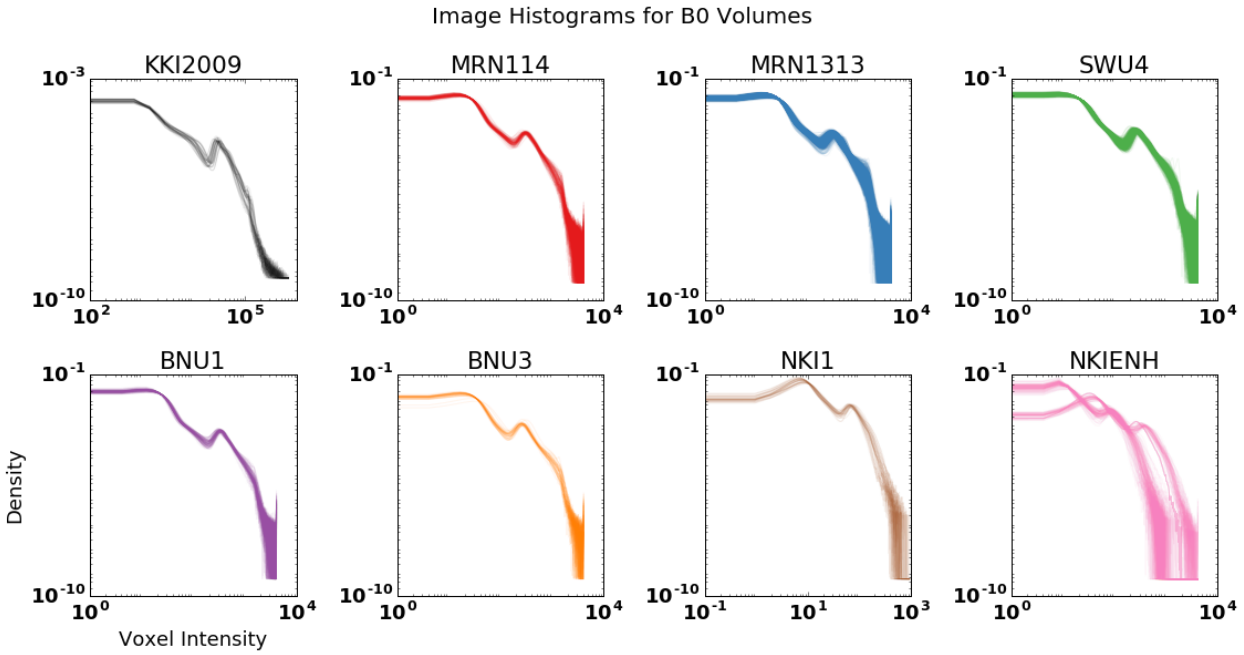


Figure 9: B0 image intensities

The data summarized here have been run through the ndmg pipeline. We then summarized the graphs through several metrics, two of which are shown here in Figures 10 and 11: degree distribution and number of non-zero edges. We see here that the subtle differences observed - as well as differences we didnt notice from histograms have a pronounced effect in our graphs.

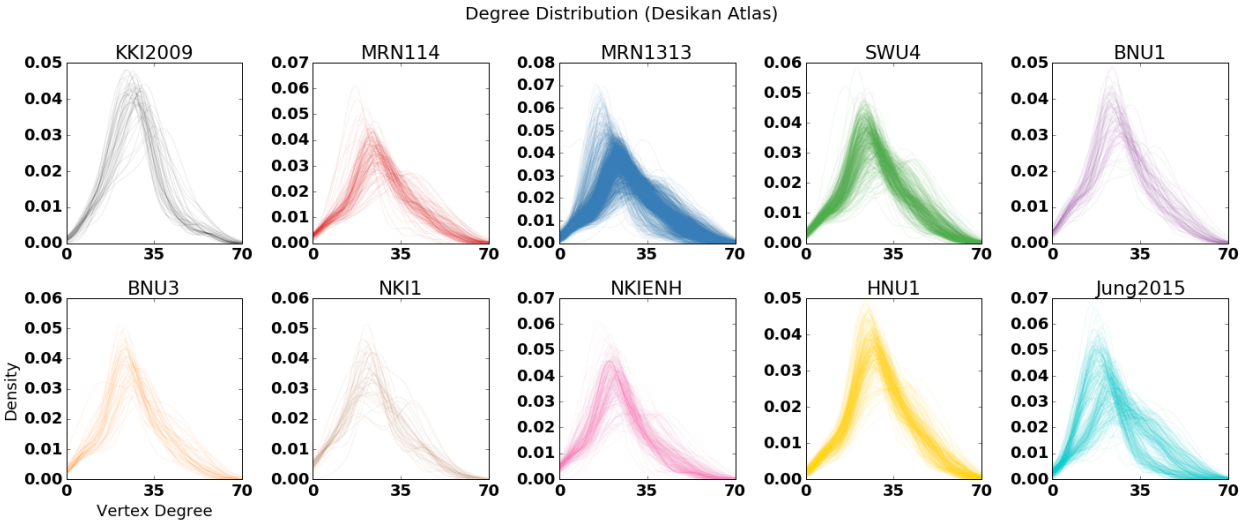


Figure 10: Degree distribution

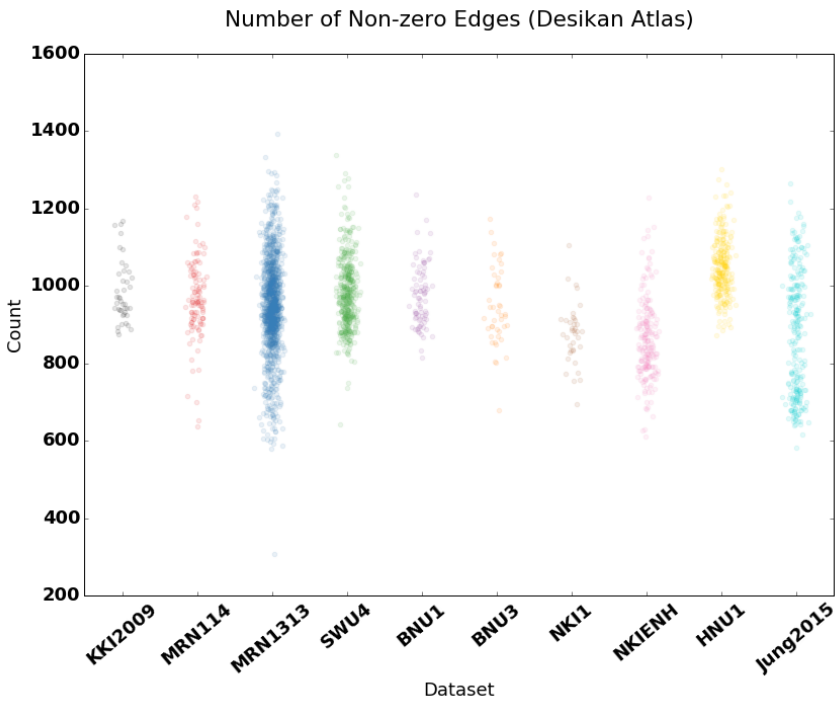


Figure 11: Number of non-zero edges

In order to assess the significance of these batch effects, we design a classification experiment with the graphs in which we attempt to predict subject sex from their connectome. If there were no batch effects, training with LOO-dataset or LOO-subject techniques would achieve approximately the same results (with the one factor contributing to differences being the slight difference in training sample size when we have a larger leave-out set). However, when we attempt to classify with leave out one dataset we notice that we achieve no better prediction than chance, whereas leave out one subject achieves 70% accuracy using the simplest classifier. This result is shown in Figure 12.

1.8.3 Science in the Cloud

The burden of merely reproducing another scientist's results, much less extending it further, is often unbearably high, even on public datasets. A solution to this challenge is something we call "scientific in the cloud (SIC), a term expanded from the computer science term "software container". *SIC* contains all of the software one needs to 1-click reproduce all the scientific claims of a result, while pulling the data from publicly accessible data repositories.

We have taken the first step we have taken towards achieving the goals of reproducibility, reliability, extensibility, and accessibility of scientific products. Specifically, as illustrated in Figure 13 we introduce a working example of SIC, one of the first to our knowledge. Our container performs an analysis of a cohort of structural and diffusion magnetic resonance imaging scans by (i) downloading the required data from a public repository over the internet, (ii) fully processing each subject's data to estimate a connectome for each and associated graph statistics using the ndmg pipeline, and (iii) optionally, plot quality control figures on

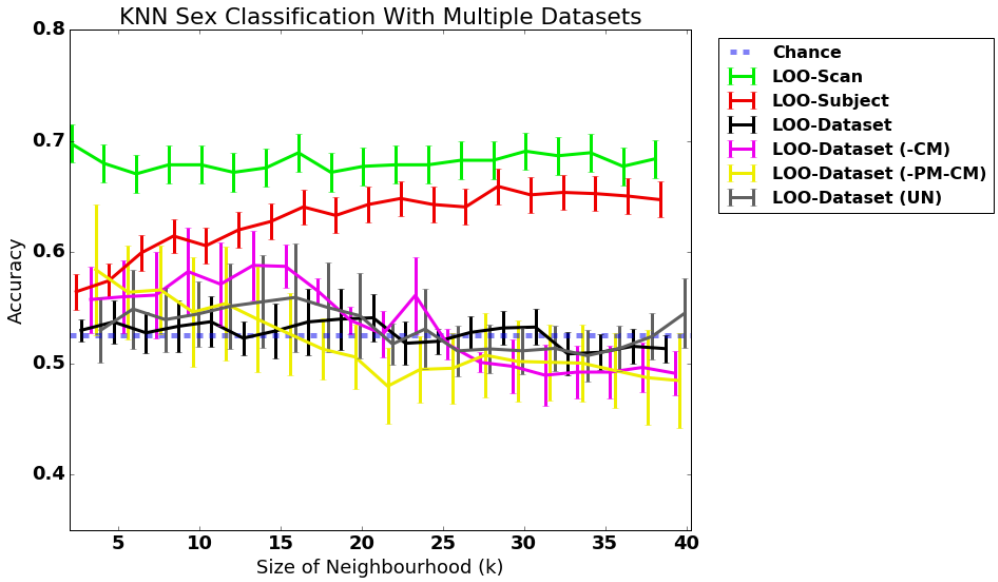


Figure 12: Classifier performance using KNN across datasets

various graph statistics. We hope this demonstration is a useful example of reproducible, and more importantly, extensible science.

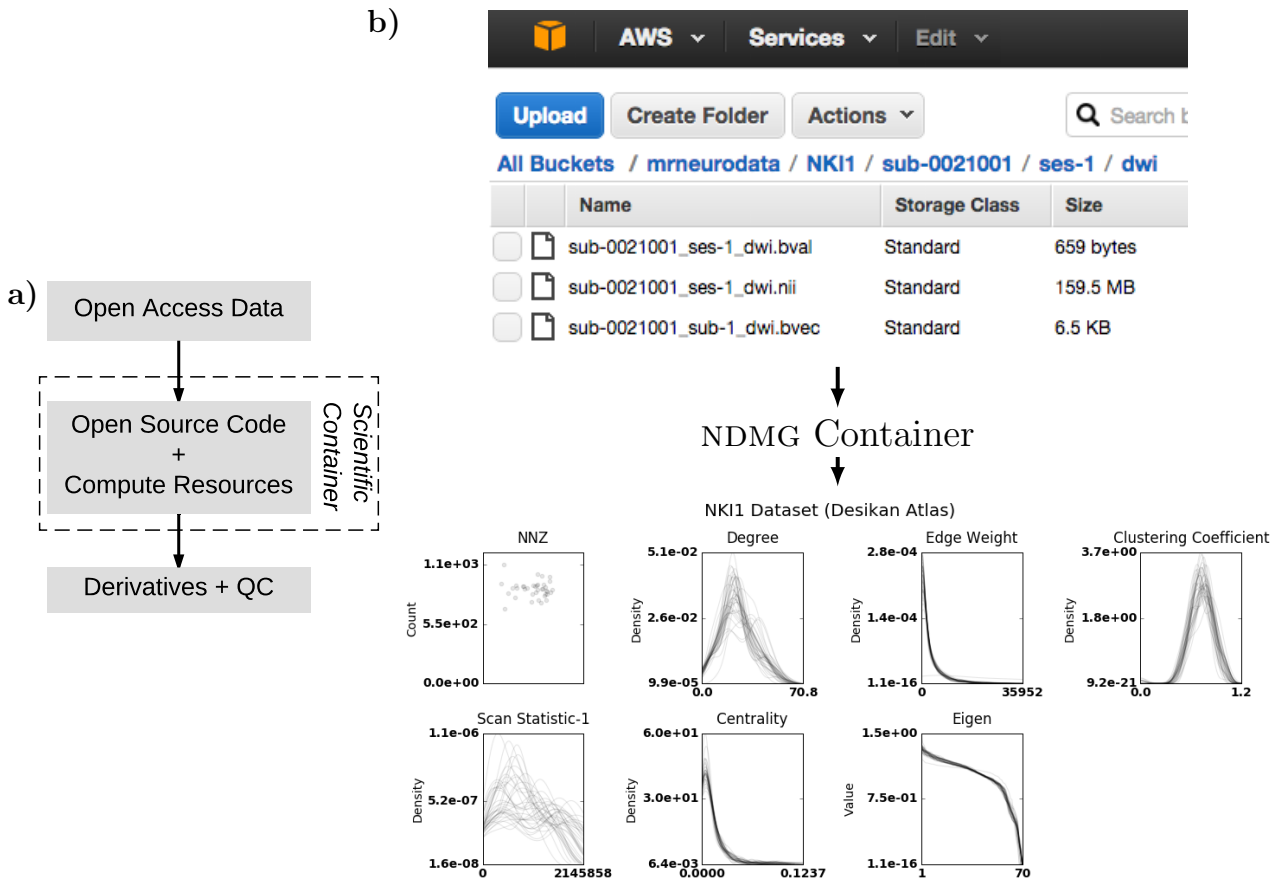


Figure 13: *SIC* is the partnership of open-source code and a reproducible computing environment, which can be used to process open-access data and produce data derivatives and quality control metrics on them

1.9 Randomer Forest (RerF)

Random Forest (RF) remains one of the most widely used general purpose classification methods, due to its tendency to perform well in a variety of settings. One of its main limitations, however, is that it is restricted to only axis-aligned recursive partitions of the feature space. Consequently, RF is particularly sensitive to the orientation of the data. Several studies have proposed oblique decision forest methods to address this limitation. However, the ways in which these methods address this issue compromise many of the nice properties that RF possesses. In particular, unlike RF, these methods either dont deal with incommensurate predictors, arent well-adapted to problems in which the number of irrelevant features are overwhelming, have a time and space complexity significantly greater than RF, or require additional hyperparameters to be tuned, rendering training of the classifier more difficult. Our proposed method, which we call RerF, seeks to address these limitations.

Previously, RerF, Random Forest (RF), Forest-RC (F-RC), and Random Rotation Random Forest (RR-RF) were evaluated by comparing out-of-bag errors on simulated and benchmark datasets. However, out-of-bag errors are potentially biased estimates of the true generalization error. We have since repeated all analyses, as illustrated in Figure 14 instead comparing error rates on separate held-out datasets. Additionally, previously we evaluated a method called RerF(rank), which rank transforms the data prior to training the forest. The hope in rank transforming the data is to achieve scale and unit invariance as well as robustness to outliers. However, we did not evaluate the other methods (RF, F-RC, and RR-RF) after rank transforming the data. Furthermore, other ways of achieving scale invariance were not compared (i.e. rescaling each feature to $[0, 1]$ and z-scoring). Over the past weeks we have begun evaluating all classifier methods trying all rescaling methods (i.e. ranking, $[0, 1]$ normalization, z-scoring). Preliminary results suggest that rank transforming does not help with outliers to our surprise, and z-scoring is just as effective at achieving scale invariance.

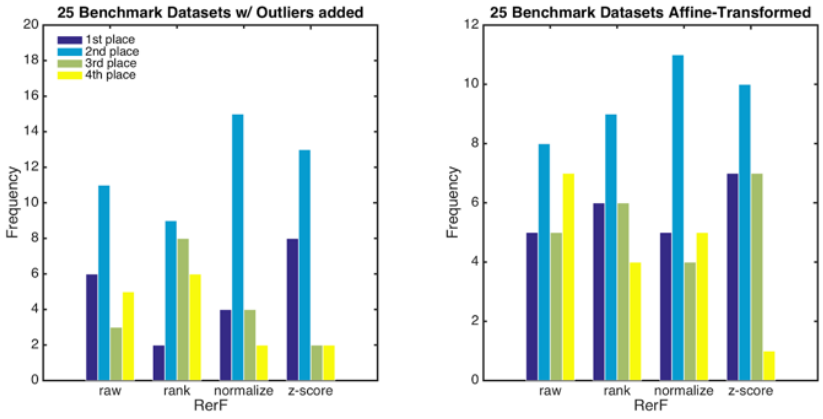


Figure 14: Distributions of "placement" for each method of RF when evaluated over various sets of datasets.

1.10 Discriminability

We develop a measure of discriminability (or reliability). It is intuitive to understand and easy to implement. Discriminability is defined to be the probability of within subject distances being smaller than the cross subject distances. If we let $x_{i,t}$ denote thet^{th} trial of subject i and $\delta(,)$ be the metric, the discriminability D is

$$D = P(\delta(x_{i,t}, x_{i,t}) \leq \delta(x_{i,t}, x_{i,t}))$$

We want to search for the optimal processing pipeline which has the maximal discriminability. The next figure summarizes the steps to compute discriminability. This is summarized in Figure 15

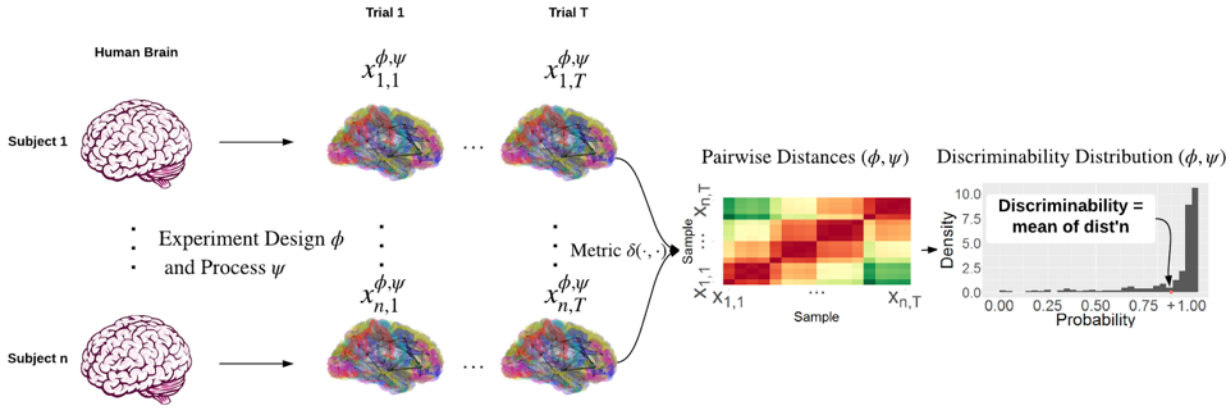


Figure 15: Workflow of discriminability analysis.

fMRI processing is discussed in previous reports. We further apply our methodology of discriminability to DTI processing. In particular, we considered 5 test-retest DTI data sets (BNU1, HNU1, NKI1, KKI and SWU4) collected under different experiment design options. We registered them with 15 atlases, and estimate graphs using raw, log and rank edge weights. The discriminability estimates of processed data sets is then computed. The results are shown in the next figure. The top left panel shows the discriminability of SWU4 data set. Log edge weights seems to be the best and the discriminability is stable across atlases with the number of ROI ranging from 48 to 1875. The bottom left panel shows the discriminability difference between four DTI and fMRI data sets. DTI data sets have comparable discriminability as fMRI data sets. Actually, DTI measurements are better for 3 out of 4 data sets. The top right and bottom right panels comparing two DTI experiment design options. Currently, we cannot conclude how number of directions and b-value affect discriminability.

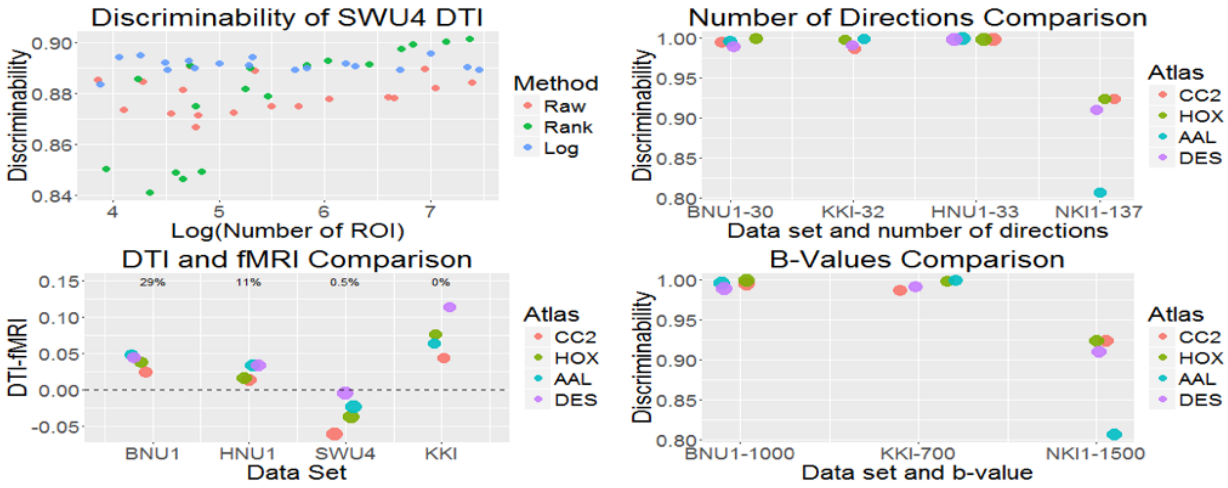


Figure 16: Discriminability results for several datasets with both fMRI and DTI derived graphs.

1.11 Synaptome Statistics

1.11.1 Molecular Synapse Types

Exploratory statistical analysis is being conducted on Synaptome data sets using the R language and environment for statistical computing and graphics. The goal is to discover synapse taxonomy for synapse level neuroscience. A pairwise regression against GABABR, one example analysis, is shown in Figure 17.

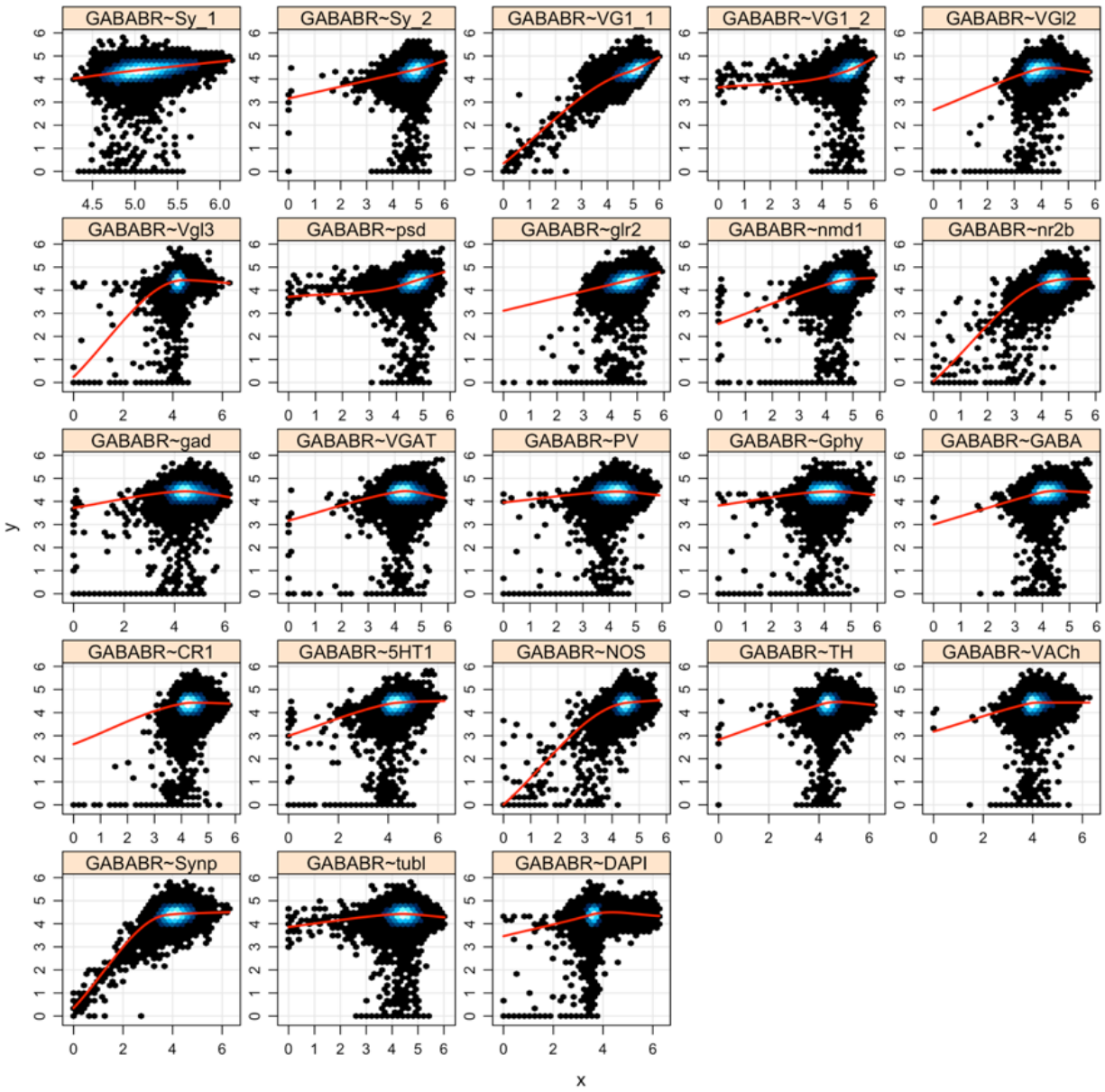
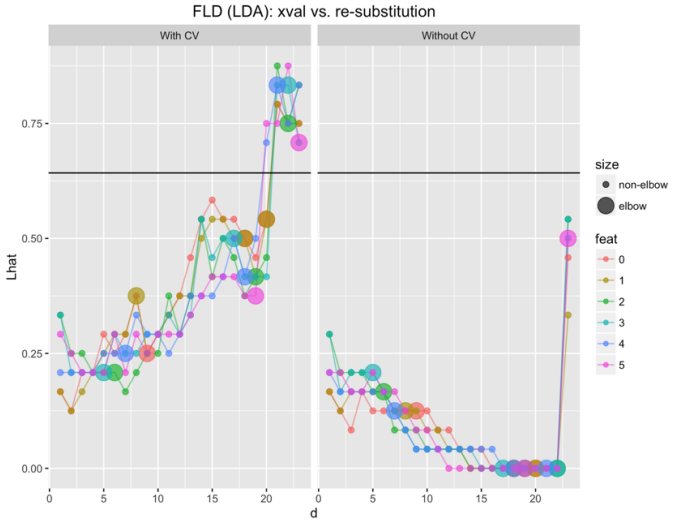


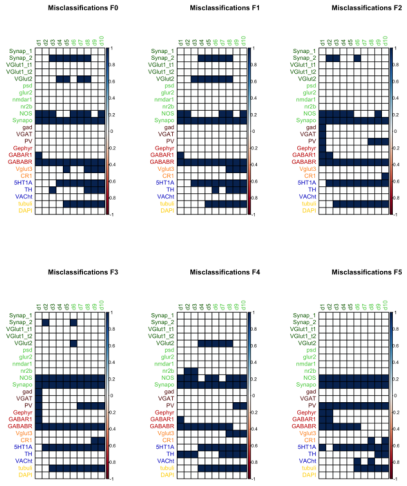
Figure 17: Pair-wise regressions involving GABABR.

1.11.2 PCA clustering

Continuing with discovery of properties of synapses using the Kristina15 dataset from the Allen Institute for Brain Science; the correlation matrix of the synapse markers was computed for each feature and passed through principal components analysis composed with linear discriminant analysis for embedding dimensions 1 through 23 with and without cross validation. The "true classes were set as the standard believed functions of each marker, with the exception of those that are believed to have multiple. The misclassifications resulting from Linear Discriminant Analysis (LDA) with cross validation are given below. The rows of the figure on the right correspond to markers, the columns correspond to the embedding dimension. A misclassified point is denoted by a colored square. This will be used to identify markers that could possibly be considered as acting differently from their believed class.



(a) Error rate for LDA with and without CV



(b) Misclassifications

1.12 Law of Large Graphs

We proposed an algorithm to estimate the mean of a collection of graphs. Our methodology is motivated by the asymptotical distribution of the adjacency spectral embedding of random dot product graphs. To take advantage of the low-rank structure of the graphs, adjacency spectral embedding, a rank-reduction procedure, is applied to the element-wise MLE. We then give a closed form for asymptotic relative efficiency between our estimator and the element-wise MLE, which theoretically proves that our estimator has smaller variance with sufficiently large number of vertices while keeping to be asymptotically unbiased. These results are demonstrated by various simulations. Moreover, our estimator also outperforms element-wise MLE for the CoRR brain graphs, which shows our estimator is valid even when the data does not perfectly follow a SBM.

An example SBM is illustrated in Figure 19. We consider a 5-block SBM and plot the corresponding probability matrix and one adjacency matrix generated from it with 200 vertices. From the top two panels of the figure, we can clearly see the structure of 25 blocks in both the probability matrix and the adjacency matrix as a result of 5 different blocks among vertices.

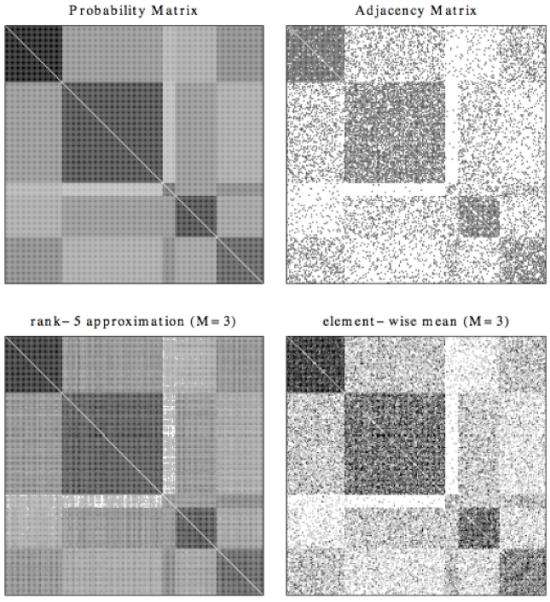


Figure 19: The top left figure shows the mean graph P with $K = 5$ blocks and $N = 200$ vertices and the top right figure shows an adjacency matrix A sampled according to the probabilities from P. While A is a noisy version of P, much of the structure of P is preserved in A, a property we will exploit in our estimation procedure. Based on three graphs sampled independently and identically according to the probability matrix P, we construct the element-wise mean $\bar{A}$, shown in the lower right panel. Finally, by taking a rank-5 approximation of $\bar{A}$ and thresholding the values to be between 0 and 1, we construct our proposed estimate $\hat{P}$, shown in the lower left panel. By visual inspection, it is clear that the low-rank estimate $\hat{P}$ more closely approximates the probability matrix P as compared to $\bar{A}$.

While the theory we have developed is based on the assumption that the mean graph is low rank, $\hat{P}$ often performs well even when this assumption is false. To further illuminate this point, we performed a synthetic data analysis under a full-rank independent edge model where we used the sample mean of the 454 graphs in the Desikan dataset as the probability matrix P.

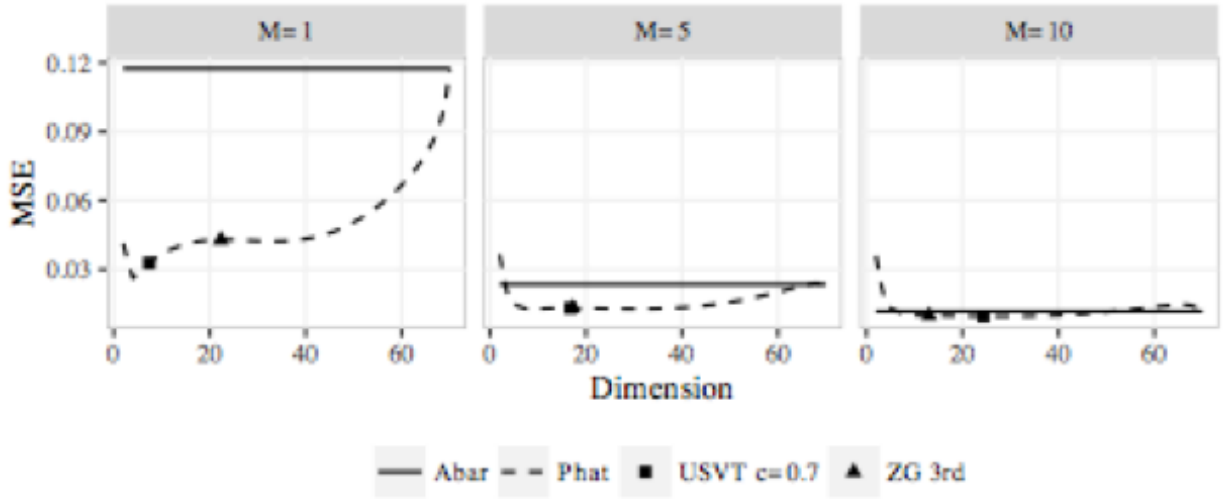


Figure 20: This figure shows $\hat{\text{MSE}}$ for $\bar{A}$ (solid line) and $\hat{P}$ (dashed line) for simulated data with different sample sizes M based on the sample mean for the Desikan dataset. Again, the average of dimensions selected by the USVT method (square) and the ZG method (triangle) tend to nearly approximate the optimal dimension. Overall, we see that the structure of these plots well approximates the structure for the real data indicating that performance for the independent edge model will tend to translate in structure to non-independent edge scenarios. On the other hand, the relative efficiency $\hat{\text{RE}}(\bar{A}, \hat{P})$ is lower for this synthetic data analysis than for the CoRR data.

1.13 Robust Law of Large Graphs

To estimate the mean of a collection of weighted graphs under a low rank random graph model (e.g. Stochastic Blockmodel) when observing contaminated graphs, we propose an estimator which not only inherits robustness from element-wise robust estimators but also has small variance due to application of a rank-reduction procedure. Under appropriate conditions, we prove that our estimator outperforms standard estimators via asymptotic relative efficiency. And we illustrate our theory and methods by Monte Carlo simulation as following.

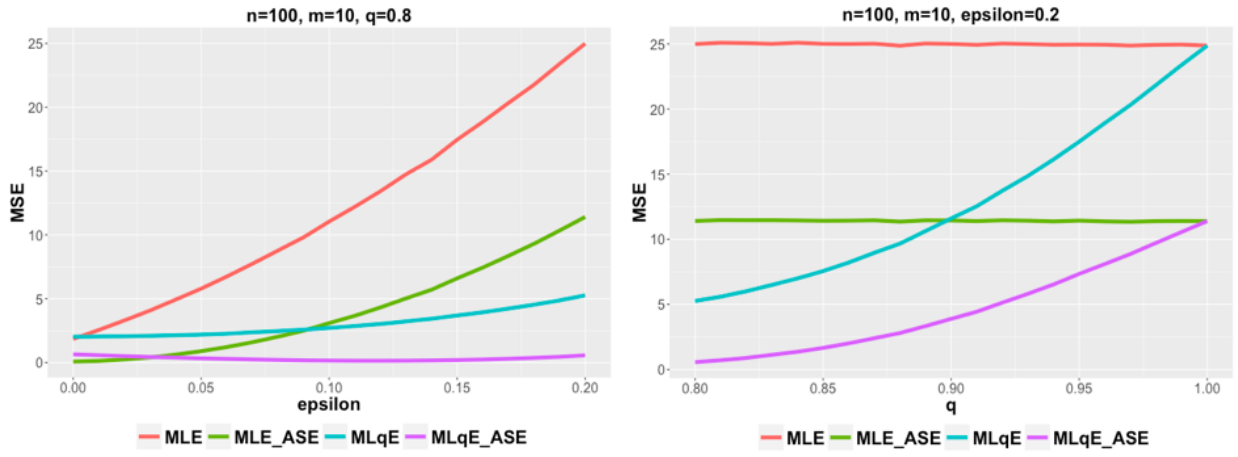


Figure 21: For the figure on the left, we vary the proportion of the contamination represented by ϵ and conclude: **1. MLE (red) vs MLqE (blue)**: MLE outperforms a little bit when there is no contamination, but it degrades dramatically when contamination increases; **2. MLE (red) vs MLE_ASE (green)**: MLE_ASE wins the bias-variance tradeoff; **3. MLqE (blue) vs MLqE_ASE (purple)**: MLqE_ASE wins the bias-variance tradeoff; **4. MLqE_ASE (purple) vs MLE_ASE (green)**: When contamination is large enough, MLqE_ASE is better, since it inherits the robustness from MLqE. For the right panel, we vary the parameter q in the MLqE. Note that when $q = 1$, MLqE becomes MLE so that the curves intersect. As q decreases, we intend to have more bias but still win the bias-variance tradeoff by reducing the variance.

In summary, MLE outperforms a little bit when there is no contamination, but it degrades dramatically when contamination increases while MLqE still does a good job. Applying ASE to either entry-wise MLE or entry-wise MLqE reduces the variance by bias towards the low rank approximation. Our estimator based on ASE of MLqE inherits the robustness from MLqE, and it has low variance. When contamination is large enough, our estimator performs the best among all the four.

1.14 LOL

We have proven the conditions under which LOL outperforms PCA. Specifically, PCA only outperforms LOL is we store enough eigenvectors such that the d^{th} LOL outperforms PCA whenever the difference of the means and the first d eigenvectors contain less information than the $(d+1)^{th}$ eigenvector. The proof utilizes Chernoff divergences. More specifically, we can compute the Chernoff Information that one distribution has about another. Thus, for a given low-dimensional projection, we can evaluate how far F_0 and F_1 are from one another, which determines the induced Bayes optimal error rate. The settings for which PCA outperforms LOL are quite pathological.

1.15 Multiscale Network Test

To guarantee validity and consistency of MGC applied to testing in network, we should find independent and identically distributed (i.i.d.) configuration of each vertex in a graph (network), of which metric well reflects the distance between vertices. We demonstrated that Euclidean distance of raw adjacency matrix does not satisfy i.i.d assumption generally; while diffusion maps at every time step are i.i.d under certain latent function, which is supported by Aldous-Hoover representation theorem and de Finettes theorem. On the other hand, under these theorem, graph is empty or dense. Fortunately, we have found that exchangeable graph can be generated more generally, even containing sparse graphs. We generate a simple simulation to check whether MGC works or not. Thus we are going to test independence between diffusion maps at each time point t and nodal attribute X.

1.16 Multiscale Generalized Correlation (MGC)

1.16.1 Using MGC for Human Brain Classification

We first investigate whether brain shape and disease status are dependent on one another. Previous investigations have linked major depressive disorder to the hippocampus shape, though global tests were unable to detect a statistically significant dependence structure at the = 0.05 level. This brain shape versus disease dataset consists of n = 114 subjects, for each we have an MRI scan as well as a discrete variable indicating whether the subject is clinically depressed (2), high-risk (1), or non-affected (0). From the MRI data, previous work extracted both the left and right hippocampi. For the brain shape view of the data, they computed the interpoint comparison matrices using a nonlinear landmark matching approach. For the discrete disorder variable, we use squared Euclidean distance, then add 1 to every non-diagonal entry (so only the diagonals are of distance 0).

The next experiment investigates whether brain networks are independent of creativity. Neural correlates of creativity have previously been investigated, though largely using structural MRI and cortical thickness. Here, we used data from a previously published result on graph similarity, which included for each of n = 109 subjects, we have both diffusion weighted (DW-) MRI data as well as the subjects creativity composite index (CCI). We processed the raw DW- MRI data via MIGRAINE, a pipeline for estimating brain networks from diffusion data. To compute the distance between graphs, we use the semiparameteric test statistic, developed specifically to compare pairs of graphs with labeled vertices. This test statistic was developed to reduce the noise due to the very high-dimensionality of adjacency matrices via employing adjacency spectral graph embedding to reduce the dimensionality into something much smaller; in this case, we chose to embed each graph into 2 dimensions for simplicity. We use simple squared error (Euclidean metric) to compare CCI values.

In the last experiment, MGC is applied to test independence between brain voxel activities and non- existent stimulus similar to a pair of studies led by Eklund et al., by using 26 resting state fMRI data sets from the 1000 functional connectomes project[5], consisting of a total of 1604 subjects. We used CPAC to estimate regional time-series, in particular, using the sequence of pre-processing decisions determined to optimize discriminability. The output for each scan is the resting state fMRI time-series data containing 200 regions of interest for 200 time-steps. We then also generate an independent stimulus by sampling from a standard normal at each time step. Of course, the brain activity data and the stimuli are independent by construction. For each brain region, we test: is activity of that brain region independent of the time-varying stimuli. We pool brain activity over all of the samples from the population. Any regions that are detected significant are false positives by definition. By testing reach brain region separately, we obtain a distribution of false positive rates. If our test is unbiased, that distribution should be centered around the significance level, which we set at 0.05 for this experiment. This is shown in Figure 22.

[5] http://fcon_1000.projects.nitrc.org/

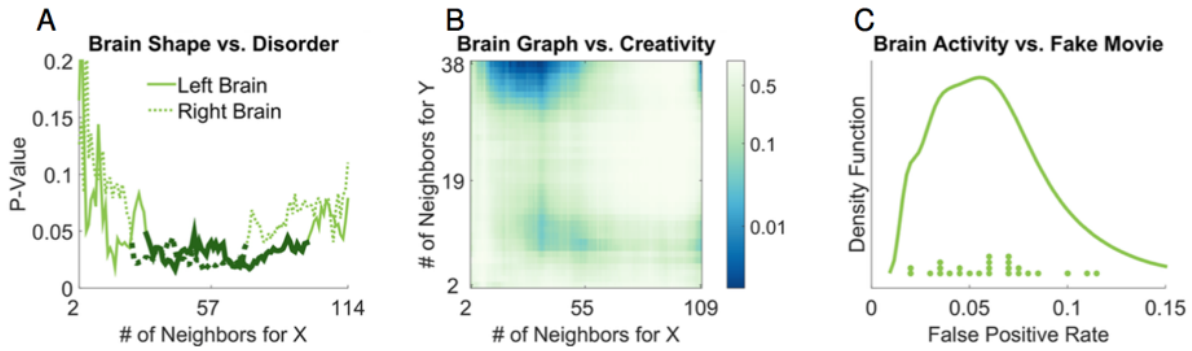


Figure 22: (A) Local correlation p-value curves with respect to k = 2, . . . , 114 at l = 4 for brain vs disease. l = 4 is the largest possible neighborhood size for the disease data, as it is categorical. Thick solid black lines correspond to the largest region of p-values ¡ 0.05. (B) Local correlation p-value heat map with respect to k = 2,...,109 and l = 2,...,38 for brain MIGRAINE vs CCI. l = 38 is the largest possible neighborhood size for the CCI data, due to repeated distance entries. (C) Density estimate for the false positive rates of MGC on the brain vs noise experiments, with the actual rate of each data shown as dots above the x-axis.

1.16.2 Optimal Local Correlation

MGC is the optimal local correlation between two datasets X and Y. For any given global correlation (Pearsons, rank, Mantel, distance correlation, etc.), their respective local correlations can be efficiently computed. By choosing the optimal local correlation based on maximizing testing powers, the Oracle MGC dominates the global correlation.

We demonstrate that Oracle MGC is a consistent test statistic (power converge to 1 as sample size increases) under standard regularity conditions, is equivalently to the global correlation under linear dependency (i.e., each observation X_i is a linear transformation of Y_i), and can be strictly better than the global correlation under common nonlinear dependencies. Thus Oracle MGC dominates the global correlation, and the sample MGC (i.e., choose the optimal scale by p-value map approximation, as the testing power are not available in the absence of the true model and training data) also empirically dominates the global correlation.

Numerically, we showed that both Oracle and sample MGC significantly improve over the global correlation and other existing state-of-the-art methods for the dependence test. Moreover, the optimal scale helps discovering the nature of the dependency, i.e., the global scale is close to optimal in the power / p-value map if and only if the underlying dependency is close to linear.

On real data, MGC helps identify useful relationships between brain activity vs personality, brain hippocampus vs major depressive disorder, which was confirmed by domain experts but not detected on raw data by existing statistical methods.

2 Data

The other core goal contained within SIMPLEX is the generation and management of data. Here, we describe progress made for which direct data derivatives now exist where they did not previously. This ranges from newly ingested datasets into our terrascale spatial database, aligned volumes, or new annotation datasets.

2.1 MRI in ndstore

The first aligned MRI dataset ingested into ndstore and made publicly available. Dataset consists of 21 subjects, each scanned twice, with Diffusion Weighted Imaging (DWI) MRI and structural MRI. The aligned DWI images, later used for connectome estimation, have been ingested in a project with a channel for each scan session, totalling 42. The images are all (182, 218, 182, 33) in size, where the first three dimensions indicate spatial dimensions, and the final dimension represents diffusion directions used during acquisition. In future MRI datasets, the first three dimensions will be consistent, though the fourth dimension will range from study to study.

2.2 Aligned CLARITY brains

The Allen Reference Atlas (ARA) was aligned to 10 CLARITY brains, an example shown in Figure 23, using Mutual Information based Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm. For registration the 20000 x 20000 x 1000 voxel images were downsampled to 200x200x200 voxels. After rigid, affine, and LDDMM registration by ndreg the registered ARA labels were uploaded to the ndstore spatial database. These labels then could then be overlaid on the CLARITY images using ndviz.

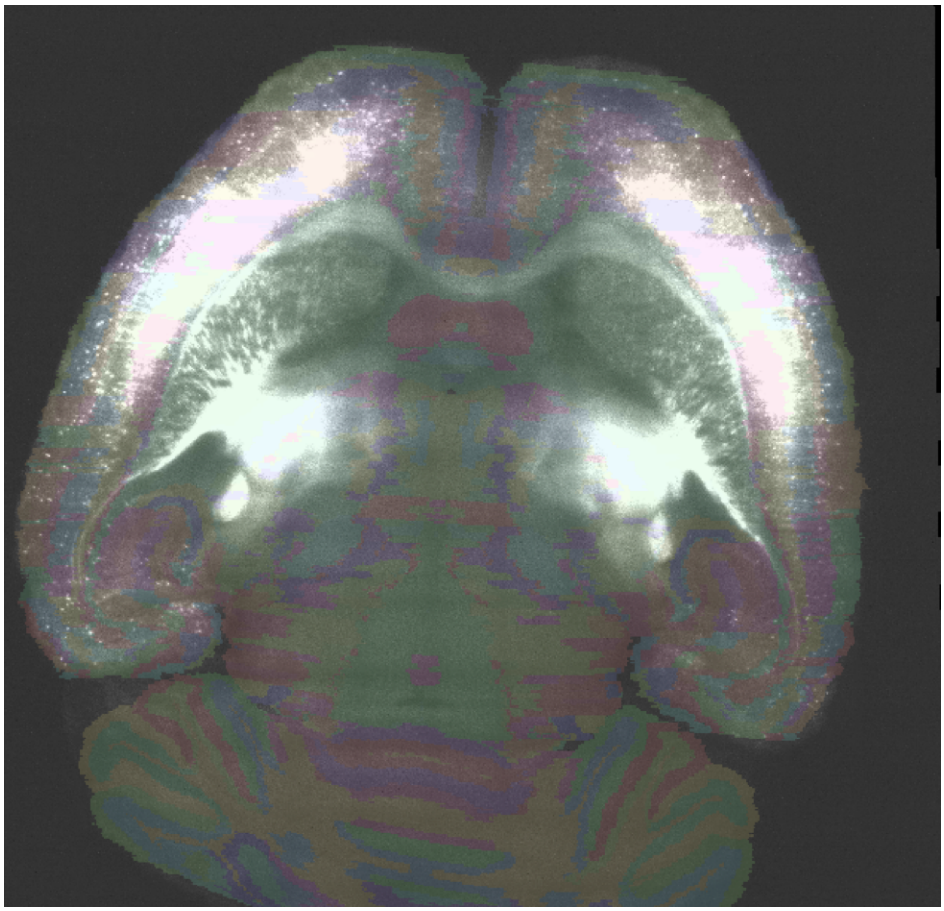


Figure 23: CLARITY brain aligned to ARA.

Below are publications, either accepted or pre-print, which have been created or updated during the reporting period.

References

[1] R. Tang, M. Ketcha, J. T. Vogelstein, C. E. Priebe, and D. L. Sussman, "Law of Large Graphs," 2016.

[2] D. Zheng, R. C. Burns, J. T. Vogelstein, C. E. Priebe, and A. S. Szalay, "An ssd-based eigensolver for spectral analysis on billion-node graphs," *CoRR*, vol. abs/1602.01421, 2016. [Online]. Available: http://arxiv.org/abs/1602.01421

[3] D. Zheng, D. Mhembere, V. Lyzinski, J. T. Vogelstein, C. E. Priebe, and R. C. Burns, "Semi-external memory sparse matrix multiplication on billion-node graphs in a multicore architecture," *CoRR*, vol. abs/1602.02864, 2016. [Online]. Available: http://arxiv.org/abs/1602.02864

[4] E. L. Dyer, W. G. Roncal, H. L. Fernandes, D. Gürsoy, X. Xiao, J. T. Vogelstein, C. Jacobsen, K. P. Körding, and N. Kasthuri, "Quantifying mesoscale neuroanatomy using x-ray microtomography," *arXiv preprint arXiv:1604.03629*, 2016.

[5] V. Lyzinski, K. Levin, D. E. Fishkind, and C. E. Priebe, "On the consistency of the likelihood maximization vertex nomination scheme: Bridging the gap between maximum likelihood estimation and graph matching," *arXiv preprint arXiv:1607.01369*, 2016.

[6] T. M. Tomita, M. Maggioni, and J. T. Vogelstein, "Randomer forests," *arXiv preprint arXiv:1506.03410*, 2015.

[7] A. Athreya, M. Tang, V. Lyzinski, Y. Park, B. Lewis, M. Kane, and C. Priebe, "Numerical tolerance for spectral decompositions of random dot product graphs," *arXiv preprint arXiv:1608.00451*, 2016.