NeuroData SIMPLEX Report: February 2017

The following report documents the progress made by the labs of PI Joshua T. Vogelstein and Co-PIs Randal Burns and Carey Priebe at Johns Hopkins University towards goals set by the DARPA SIMPLEX grant.

Contents

1 Data: What's in the Cloud

2 Statistical Theory and Methods

2.1 LOL @jovo

LOL took a backseat this month while jovo had a baby :)

2.2 meda @JesseLP

Updates in meda include the addition of plots that explore the clusters generated by hierarchical clustering. We are using mclust [1] in our hierarchical clustering function. At each level we use the Bayseian Information Criterion (BIC) to determine if the data should be split into two clusters or kept as one. The dendrogram (figure 1 left) shows the binary tree clustering structure with branch size denoting the size of the cluster. The stacked level means plot (figure 1 right) shows the means of features in each node of the tree.

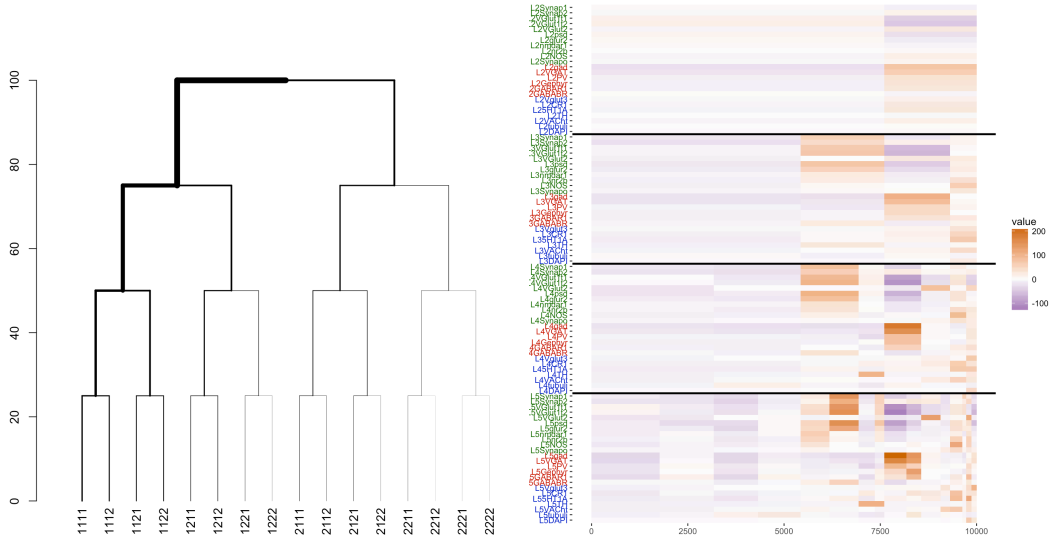

Figure 1: Left: A dendrogram showing the results of hierarchical mclust. The splits are constrained to be binary and branch sizes show relative cluster sizes. Right: A stacked level means plot showing for each node in the dendrogram the feature means.

2.3 Randomer Forest (RerF)

RerF is now working in pure R code. The running time of this implementation is roughly the same as the Matlab implementation for various sizes of input. We are re-implementing portions of the code in both C and FlashR in an attempt to reduce the running time of the algorithm.

Previously, we did not have any simulation experiments in which we know for a fact that RF is the "right" thing to do. We conducted such experiments to see how much RerF and RR-RF lose by allowing oblique split directions. The simulated datasets were constructed as follows. Data was sampled in p dimensions over a unit hypercube centered at the origin. Datapoints all falling into the same orthant were assigned the same class label. Therefore, for p dimensions, there are 2^p unique class labels. The true decision boundary separating the classes is purely axis-aligned. In such a case, RF is the best classifier among the class of all ensemble tree classifiers.

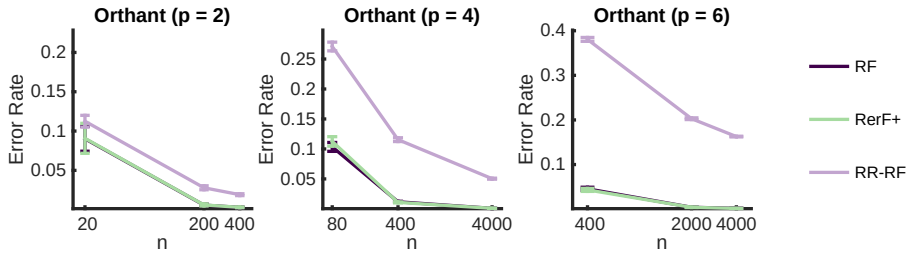

Figure 2: Error rate of RF, RerF, and RR-RF on the "Orthant" dataset as a function of n, the number of training samples, for three values of p. The results indicate that there is no significant difference in performance between RerF and RF, while RR-RF performs significantly worse across all settings.

2.4 Non-Parametric Shape Clustering

We have been mostly focused on developing non-parametric clustering methods. To this purpose, we are exploring ideas from energy statistics, which is non-parametric, robust, and rotational invariant, thus it incorporates the main ingredients that we are looking for. The main difficulty is to formulate an algorithm based on this, i.e. to identify the correct test statistic, or to formulate it as a feasible optimization problem. Consider K-Means clustering problem which is $\min_{\{\mathcal{C}_k\}} \sum_{k=1}^{K} \sum_{x \in \mathcal{C}_k} \|x - \mu_k\|^2$, where $\mathcal{C}_k$ is the kth cluster and μ_k the mean of its points. We showed that this problem is equivalent to

$$\max_{G} \operatorname{Tr}\left(G^T K G\right) \qquad \text{s.t.} \quad G \geq 0, G^T G = I, G G^T e_1 = e_1. \tag{1}$$

where $e_1 = (1, 1, \ldots, 1)^T$. This is a Quadratically Constrained Quadratic Problem (QCQP), which is usually NP-hard. Analogously, consider the energy function $\mathcal{E}(F, G) = 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|$ between $X, X' \sim F$ and $Y, Y' \sim G$. We showed that this can be written as $\mathcal{E}(A, B) = e_1^T \Delta e_1$, where Δ is a dissimilarity matrix between the two sets of data points $A \overset{iid}{\sim} F$ and $B \overset{iid}{\sim} G$. Consequently, a simple two-class clustering problem would be

$$\max_{x, z \in \mathbb{R}^N} x^T \Delta z \qquad \text{s.t. } x_i^2 = 1, x + z = 0, \tag{2}$$

which is also a QCQP problem. We are currently investigating this problem and trying to generalize it correctly for more classes. A simple check of the energy function as a test statistic is shown in Fig. 3. Under the null $F = G$, T converges to a quadratic form of normally distributed random variables. This seems to be the case in the first (blue) histogram, while it is definitely not the case in the other (red and green) histograms. For the blue histogram a single test gives $T \approx 0.32$ (small), for the red histogram $T \approx 4000$ (large), and for the green histogram $T \approx 105$ (large), with only a few points. Thus, energy statistics based approach is able to distinguish between different distributions, even when the clusters have the same mean, which is a property that K-Means cannot resolve.

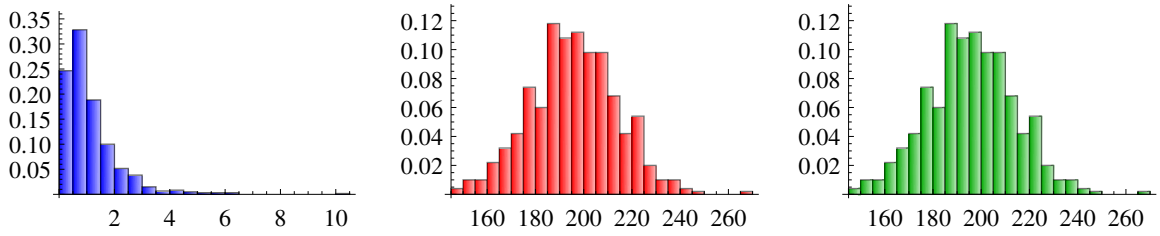

Figure 3: Distribution of test statistic $T \equiv \frac{nm}{n+m}\mathcal{E}(A, B)$ for an ensamble obtained from two distributions: $A \overset{iid}{\sim} \mathcal{N}(\mu_A, \sigma_A^2)$ and $B \overset{iid}{\sim} \mathcal{N}(\mu_B, \sigma_B^2)$, where $|A| = n$ and $|B| = m$. Blue histogram: $\mu_A = \mu_B = 0$ and $\sigma_A = \sigma_B = 1$; Red histogram: $\mu_A = -\mu_B = 1$ and $\sigma_A = \sigma_B = 1$; Green histogram: $\mu_A = \mu_B = 0$, $\sigma_A = 1$ and $\sigma_B = 1.5$.

2.5 Joint Embedding

We developed a method to jointly embed multiple graphs/networks. Previous spectral embedding techniques work on each graph separately. Our joint embedding approach generalizes Adjacency Spectral Embedding to multiple graphs. Specifically, the joint embedding method identifies a linear subspace spanned by rank one symmetric matrices and projects adjacency matrices of graphs into this subspace. Given m graphs $\{G_i\}_{i=1}^m$ with $\mathbf{A}_i$ being the corresponding adjacency matrix, the d-dimensional joint embedding of graphs $\{G_i\}_{i=1}^m$ is given by

$$(\hat{\mathbf{H}}, \hat{\mathbf{D}}_1, ..., \hat{\mathbf{D}}_m) = \underset{\mathbf{D}_i, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \mathbf{H}\mathbf{D}_i\mathbf{H}^T\|^2$$
$$\text{subject to} \quad \mathbf{D}_i \text{ being diagonal.}$$

Here, h_k is the kth column of matrix $\mathbf{H}$. The $\hat{\mathbf{H}}$ are estimated latent positions for vertices, and the diagonal of $\hat{\mathbf{D}}_i$ can be treat as the feature of graph i. We performed theoretical and numerical analysis of the joint embedding. The code and paper can be found here.

We study predicting individual composite creativity index (CCI) through brain connectomes obtained by Multimodal Magnetic Resonance Imaging. In total, 113 healthy, young adult subjects were scanned and their CCI is assessed by independent judges using the Consensual Assessment Technique. First, we jointly embed brain graphs of all subjects. Figure 4 shows a typical graph and $\hat{h}_6\hat{h}_6^T$ estimated by the joint embedding. Next, we construct a linear regression model by treating the diagonal of $\hat{\mathbf{D}}_i$ as explanatory variables and CCI as the response variable.

Overall, the regression model for predicting CCI is significant at level 0.05 compared to the null model. We found CCI positively correlated to overall connectivity of the brain. Furthermore, we found CCI significantly negatively related to $\hat{h}_6\hat{h}_6^T$. This implies that compared to within hemisphere connectivity across hemisphere connectivity has a larger positive impact on human creativity.

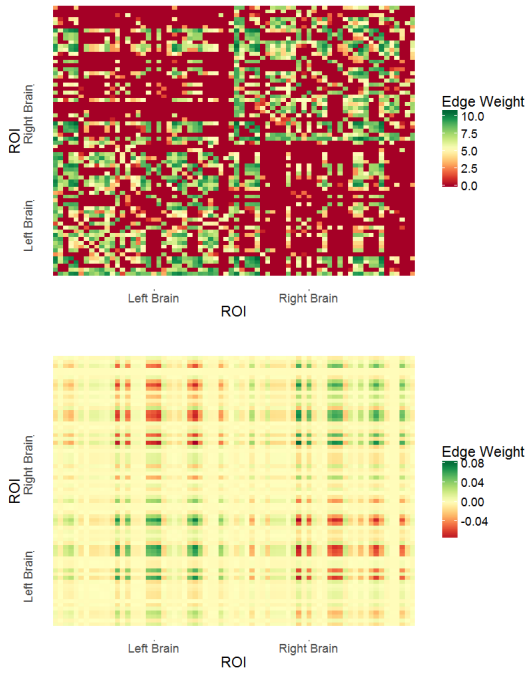

Figure 4: The top panel shows the graph derived from a typical subject. There is much more neural connectivity within each hemisphere. The bottom panel shows the rank one matrix $\hat{h}_6^T \hat{h}_6$, which has positive connectivity within each hemisphere, but negative connectivity across hemispheres.

2.6 Law of Large Graphs

We showed that our estimates are better in terms of the MSE for the real data. To further illustrate the differences between the two estimators, now we examine the actual estimates to see how they look like. In order to emphasize the connectome smoothing effect, here we consider the same random sample of size $M = 1$ based on the Desikan atlas in Fig. 5 and plot the estimate $\hat{P}$ in the right panel. Note that compared to the sample mean $\bar{A}$, $\hat{P}$ has a finer gradient of values which in this case leads to a more accurate estimate of the true probability matrix P, especially for edges between the two hemispheres, in the upper right and corresponding lower left block.

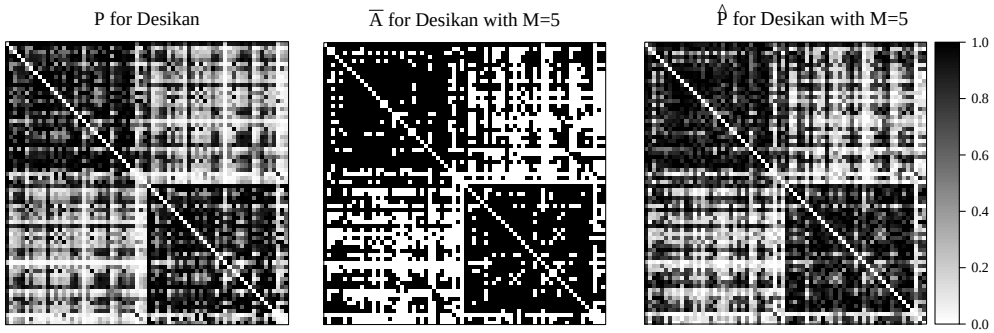

Figure 5: **Heat maps of the population mean, the sample mean, and the estimator $\hat{P}$.** These heat maps indicate the population mean for the 454 graphs (left), sample mean for the 1 sampled graph (center), and $\hat{P}$ for the 1 sampled graph with dimension $d = 12$ selected using the Zhu and Ghodsi method (right). Darker pixels indicate a higher probability of an edge between the given vertices. Note that $\hat{P}$ appears to better estimate the true probability matrix P, especially for edges between the two hemispheres, in the upper right and corresponding lower left block.

2.7 Robust Law of Large Graphs

We had theorems for the MLqE under the exponential distribution. Actually the results can be generalized to a broader class of distribution families, and even a different entry-wise robust estimator other than MLqE with the following conditions:

1. Let $A_{ij} \overset{ind}{\sim} (1-\epsilon)f_{P_{ij}} + \epsilon f_{C_{ij}}$, then $E[(A_{ij} - E[\hat{P}_{ij}^{(1)}])^k] \leq \text{const} \cdot k!$, where $\hat{P}^{(1)}$ is the entry-wise MLE as defined before; This is to ensure the that observations will not deviate from the expectation too far away, such that the concentration inequality can apply.
2. There exists $C_0(P_{ij}, \epsilon) > 0$ such that under the contaminated model with $C > C_0(P_{ij}, \epsilon)$,

$$\lim_{m\to\infty} \left| E[\hat{P}_{ij}] - P_{ij} \right| < \lim_{m\to\infty} \left| E[\hat{P}_{ij}^{(1)}] - P_{ij} \right|;$$

   It requires the contamination of the model to be large enough (a restriction on the distribution) and $\hat{P}$ to be robust enough with respect to the contamination (a condition on the estimator).
3. $\hat{P}_{ij} \leq \text{const} \cdot \hat{P}_{ij}^{(1)}$; (This might be generalized to with high probability later)
   Since we use the results of $\hat{P}^{(1)}$ to bound $\hat{P}^{(q)}$, the proof can apply directly with this condition for an arbitrary $\hat{P}$.
4. $\text{Var}(\hat{P}_{ij}) = O(m^{-1})$, where m is the number of observations.
   We will get exactly the same results under this condition. However, even if the variance of the new estimator is not of order $O(m^{-1})$, we will get similar results with a different term related to m.

2.8 Graph-testing

In neuroimaging connectomics studies, it is often desired to determine whether the observed network properties are statistically significant or not. In order to correctly achieve this, we need to define a null distribution. In order to generate the null distribution, the common technique is to generate an average of 1000 samples of graphs with the same degree sequence of the observed graphs. However, this technique does not yield a uniform sample from the null distribution, resulting in ill-conditioned tests. Here, we investigate statistically accurate methods in graph testing.

One strategy that we have explored is generating the null distribution by sampling from graphs with the same degree-sequence of the observed graph. While the samples are not generated uniformly, we know how to rescale the samples such that we can estimate the mean of the uniform distribution of the graphs. We are currently working on to extend this algorithm to validly estimate the 95 percentile as well.

Another technique that we have explored is using parametric bootstrap to obtain the critical region for any significance level. Namely, we fit a stochastic block model to the data and use Generalized Likelihood Ratio Test to determine the number of blocks that best fit the data. In the initial experiments, as expected, we see that the power of the test increases as the graph size increase. As the next step, after finding the model that best first the data, we can sample from that distribution many times, compute the test statistic, and get the critical region for any significance level. In this strategy we are not conditioning the graphs on their degree sequence, which is an advantage as the graphs tend to include noise so we cannot use the observed degree sequence as the ground truth.

3 Scalable Algorithm Implementations

3.1 FlashX

We use FlashR to process the billion-scale datasets to demonstrate its scalability (Table 1). We use three datasets here: (i) the Criteo dataset has over four billion data points with binary labels (click vs. no-click), used for advertisement click prediction; (ii) PageGraph is the adjacency matrix of a graph, which has 3.5 billion vertices and 128 billion edges; (iii) PageGraph-32ev are 32 singular vectors that we computed on the largest connected component of Pagegraph with the tools we built previously. In these experiments, we run the iterative algorithms (Logistic regression, k-means and PageRank) on the datasets until they converge.

Table 1: The runtime and memory consumption of FlashR on the billion-scale datasets on the 48 CPU core machine. The runtime of iterative algorithms is measured when the algorithms converge. We run PageRank on the PageGraph dataset, run k-means on PageGraph-32ev and the remaining algorithms on Criteo.

	Runtime (s)	Memory (GB)
Correlation	91.23	1.5
PCA	136.71	1.5
NaiveBayes	76.55	3
LDA	2280	8
Logistic regression	4154.40	26
k-means	1110.82	28
PageRank	3900	135

Even though we process the billion-scale datasets in a single machine (with 48 CPU cores), none of the algorithms are prohibitively expensive. Simple algorithms, such as Naive Bayes and PCA, require one or two passes over the datasets and take only one or two minutes to complete. Logistic regression and k-means take about $10-20$ iterations to converge. Because the PageRank implementation uses the power method, it takes 100 iterations to converge. Nevertheless, all of the iterative algorithms take about one hour or less.

FlashR scales to datasets with billions of data points easily when running out of core. Most of the algorithms have negligible memory consumption. PageRank consumes more memory because the sparse matrix multiplication in PageRank keeps vectors in memory for semi-external memory computation. The scalability of FlashR is mainly bound by the capacity of SSDs. The functional programming interface generates a new matrix in each matrix operation, which potentially leads to high memory consumption. Thanks to lazy evaluation and virtual matrices, FlashR only needs to materialize the small matrices to effectively reduce memory consumption.

3.2 ndviz

The vast majority of current high resolution imaging techniques in neuroscience are either 3-dimensional or contain a 3-dimensional component (e.g. 3-d data over time). Visualization of data produced by said techniques has traditionally been confined to the three canonical planes; xy, yz, and xz. However, advancements in Web graphics rendering have made dynamic, 3-dimensional visualization possible in modern Web browsers. Using WebGL, code running in a users Web browser can access the end users Graphics Processing Unit (GPU), taking advantage of specialized graphics hardware present in most modern desktops, laptops, and even smart phones.

Using WebGL for dynamic rendering improves both the performance and capability of a graphical Web application. To that end, we have integrated Neuroglancer, a Web visualization tool built for 3-dimensional data, into NeuroDataViz. Neuroglancer has provided us with a baseline 3-dimensional rendering tool, which we can use for specialized features (e.g. dynamic false coloring). By building on a common framework, we can contribute features back to the neuroscience community as well as take advantage of new features developed by our collaborators (or even other neuroscience users).

With this new version of NeuroDataViz, we can visualize all of our existing data in the three canonical planes mode. We are now working to build 3-dimensional meshes, both for annotated Electron Microscopy data and for thresholded Light Microscopy data. A sample of EM meshes is available in the figure below. We are now developing tools for automatically generating 3-d meshes (shapes) for both datatypes on-demand as the user makes a selection in the 3-d view.

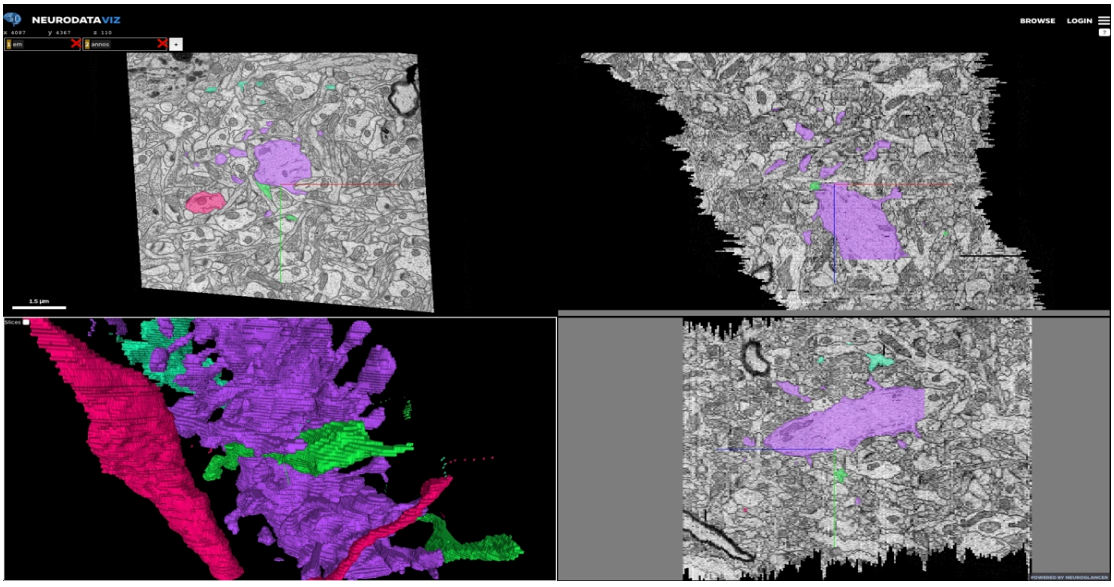

Figure 6: NeuroDataViz, powered by Neuroglancer with pre-computed meshes displayed in 3-d. Data from Harris et al. "A resource from 3D electron microscopy of hippocampal neuropil for user training and tool development," Nature Scientific Data 2015.

4 Scientific Pipelines: Infrastructure & Dataset Specific Progress

4.1 Science in the Cloud (SIC)

Through the use of Amazon's cluster-computing engine, AWS Batch, the sic use case now supports scalable deployment in the Amazon cloud natively. Additionally, the Science in the cloud (SIC) manuscript has been accepted for publication at GigaScience.

4.2 CLARITY

4.2.1 A low-latency pipeline for processing CLARITY data in the cloud

We are working on migrating our exisiting CLARITY pipeline to run entirely on virtualized infrastructure in the cloud. This workflow includes ingesting into ndstore, aligning to a reference atlas, and storing a registered stack back into ndstore. Currently, ndstore is working the cloud, with manual ingest of data. The next steps are to deploy LDDMM via Docker containers to run within the cloud environment.

5 Bibliography

Manuscripts

[1] N. Peeps, "Paper with a cool title," 2017.

Invited Talks

[1] N. Peeps, "Talk with a cool title," SIAM, JSM, NIPS, ???, Jan 2017.

Conferences

[1] N. Peeps, "Poster with a cool title," SIAM, JSM, NIPS, ???, Jan 2017.

References

[1] C. Fraley and A. E. Raftery, “Model-based clustering, discriminant analysis, and density estimation,” Journal of the American statistical Association, vol. 97, no. 458, pp. 611–631, 2002.