NeuroData SIMPLEX Report: January 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 Bibliography

Manuscripts

[1] D. Mhembere, D. Zheng, C. E. Priebe, J. T. Vogelstein, and R. Burns, "knor: A NUMA-optimized In-memory, Distributed and Semi-external-memory k-means Library," 2017.

Invited Talks

[1] D. Zheng, "FlashR: Parallelize and Scale R Machine Learning Libraries with Extreme Efficiency for Big Data," rstudio conference 2017, Jan 2017, Invited talk.

2 Data

2.1 New Datasets to the Cloud

We have now pushed several of our canonical datasets to the cloud, including 8 whole brain CLARITY specimens and two electron microscopy datasets: bock11, and kasthuri11:

Reference	Modality	Species	Bits	Proj	Ch	T	GV	Res	GB
Bhatla[1]	EM	C. elegans	8	3	3	1	437	6	248
Bock[2]	EM	M. musculus	8	1	1	1	20,249	11	13,312
Harris[3]	EM	R. rattus	8	3	3	1	19	4	9
Kasthuri[4]	EM	M. musculus	8	1	1	1	1,063	8	577
Lee[5]	EM	M. musculus	8	1	1	1	22,334	8	11,264
Ohyama[6]	EM	D. melanogaster	8	1	1	1	2,609	7	2,458
Takemura[7]	EM	D. melanogaster	8	1	1	1	190	5	203
Bloss[8]	AT	M. musculus	8	1	3	1	363	4	215
Collman[9]	AT	M. musculus	8	1	14	1	13	4	2
Unpublished	AT	M. musculus	16	1	24	1	29	3	23
Weiler[10]	AT	M. musculus	16	12	288	1	215	3	141
Vladimirov[11]	Ophys	D. rerio	16	1	1	100	9	4	9
Dyer[12]	XCT	M. musculus	8	1	1	1	3	3	3
Randlett[13]	LM	D. rerio	16	1	28	1	4	2	4
Kutten[14]	CL	M. musculus	16	1	23	1	7,191	6	6,727
Grabner[15]	MR	H. sapiens	16	1	3	1	<1	1	< 1
Totals	–	–	–	29	349	–	47,508	–	28,441

3 Tools

3.1 meda

Matrix Exploratory Data Analysis (meda) is a package being developed to allow for easy generation of modern summary statistics effective for high-dimensional data analysis.

- Source code: https://github.com/neurodata/meda
- Example output generated from Fisher's Iris data is here: http://docs.neurodata.io/meda

The goal of this package is to realize the following checklist: Given a new set of n samples of vectors in $\mathbb{R}^d$

1. histogram of feature types (binary, integer, non-negative, character, string etc.)
2. # NaNs per row? Per column? Infs per row? Per column? "Zero" variance rows? columns?
3. Heat map of raw data that fits on screen (k-means++ to select 1000 samples, CUR to select 100 dimensions)
4. 1st moment statistics
   (a) mean (line plot + heatmap)
   (b) median (line plot + heatmap)
5. 2nd moment statistics
   (a) correlation matrix (heatmap)
   (b) matrix of energy distances (heatmap)
6. density estimate
   (a) 1D marginals (Violin + jittered scatter plot of each dimension, if n > 1000 or d>10, density heatmaps)
   (b) 2D marginals (Pairs plots for top 8 dimensions, if n*d>8000, 2D heatmaps)
7. Outlier plot
8. cluster analysis (IDT++)
   (a) BIC curves
   (b) mean line plot
   (c) covariance matrix heatmaps
9. spectral analysis
   (a) cumulative variance (with elbows) of data matrix
   (b) eigenvectors (pairs plot + heatmap)

- To rescale the data in case of differently scaled features, we will implement the following options:
  - raw
  - linear options
    * linear squash between 0 & 1
    * mean subtract and standard deviation divide
    * median subtract and median absolute deviation divide
    * make unit norm
  - nonlinear
    * rank
    * sigmoid squash

- To robustify in the face of outliers, we will utilize Geometric median and robust estimation in Banach spaces
- if features have categories
    1. sort by category
    2. color code labels by category
- if points have categories: label points in scatter plots by symbol

For point 6 (a) in the above checklist we have developed functionality in `meda` to plot 1-dimensional heatmaps. The 1D heatmap is a different representation of a histogram, using color to denote count instead of bin height, see figure 1.

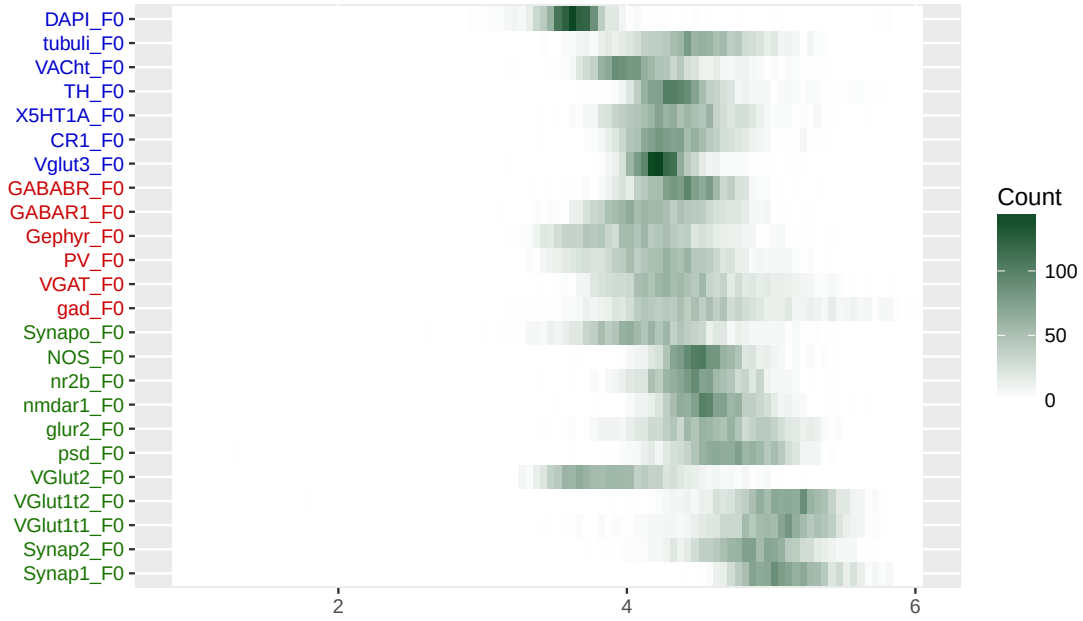


Figure 1: A one dimensional heatmap for each of the $\log_{10}$ transformed feature columns of the Kristina15 synaptome dataset. Colors correspond to the count of data points falling in to each bin. Scott's binning streategy is used which, in this case, yields 105 bins of equal width $w = 0.05$. Label colors and groups starting from the bottom: Synap1_F0 - Synapo_F0 (green, excitatory), gad_F0 - GABABR_F0 (red, inhibatory), Vglut3_F0 - DAPI_F0 (blue, other).

3.2 Randomer Forest (RerF)

Most recently, we compared classification performances of RF, RerF, RR-RF, and XGBoost on 119 benchmark datasets. RR-RF is identical to RF except that the data is randomly rotated prior to building each tree. XGBoost is a computationally efficient implementation of gradient boosted trees and has been the winner of many recent Kaggle competitions. For each dataset, for each algorithm, error was subtracted by that of RF and normalized by the chance probability of error. These normalized relative errors were then binned and the counts in each bin were computed. The y-axis represents the bins. Color indicates how many times the normalized relative error of an algorithm fell into a particular bin. For instance, the figure shows that RerF had a normalized relative error 0.05 to 0.10 less than that of RF on approximately 15 datasets. The "0 to 0" bin indicates the number of times the normalized relative error was exactly 0. Overall the figure indicates that RerF rarely loses to RF by much and frequently does substantially better. RR-RF and XGBoost, on the other hand, frequently perform worse than RF by a large margin.

Additionally, we have made progress towards a more scalable implementation of RerF in R. The previous port of RerF to R rotated input data at the tree level instead of at the more desired node level. The new port of RerF now rotates at the node level. The change to R will allow a fast RerF implementation by integrating the algorithm into FlashR.

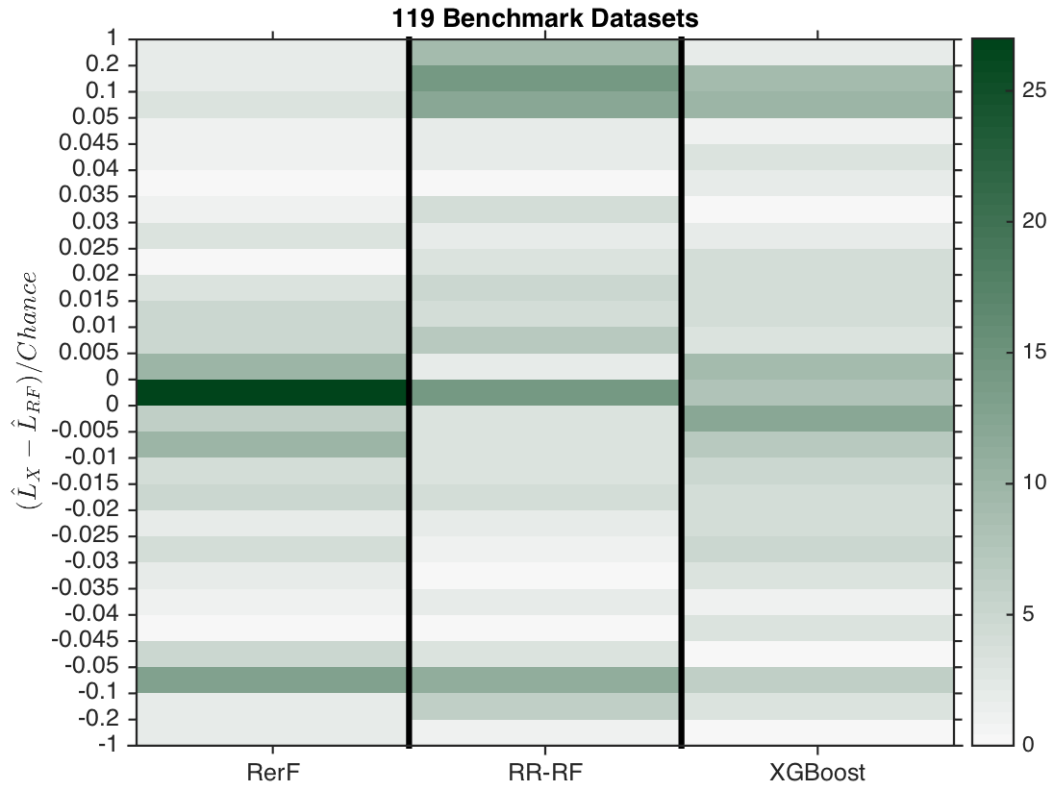


Figure 2: Classification performances of RF, RerF, RR-RF, and XGBoost on 119 benchmark datasets.

3.3 ndstore

We continue to now migrate all our annotation datasets to the cloud. The annotation projects will now be hosted in MySQL backed with AWS EBS storage and will be converted to AWS DynamoDB soon. There will be more than 30 odd annotation datasets avaliable in the cloud for public use. The datasets can be accessed at http://neurodata.io.
We added support for new resource and authentication web-servics in our python wrapper called ndio. This will allow our users to continue to use ndio for future interactions with the web back-end. They can now utilize the new web-services we have added and create resources using this which is more easier then using the RESTful web-services directly.
We are also in the process of deploying a status page for our web-services. The status page will act as a dashboard for all our users and a single point of contact for them to check if our services are online or suffering from an outage. It will also allow us to inform all our users who are subsribed to the status page of future planned outages. This will streamline our services and is standard industry practice for other companies running web-services. The status page is located here https://neurodata.statuspage.io/.
The MRI ingest service is now active and being used by some users in the lab to ingest data into ndstore. All of this service was already deployed in the cloud and the ingested data will be avaliable at http://mri.neurodata.io.

3.3.1 ndingest

The access policies for the ndingest is now complete. We are currrently testing our the new service and will soon release it in beta mode to some our close collaborators. This service will allow us to speedup our data ingest rates manifold. We plan to benchmark this service once we are done deploying it.
The ingest client developed witj JHU-APL, has now been converted so that it can use multiple threads in python. This capability will allow us to upload multiple slices of the data simultaneously and allow us to upload data to the cloud at a much faster rate.

3.4 ndreg

We received three new CLARITY image volumes from our colleagues at Stanford University. Each dataset contained two channels of a single mouse brain hemisphere at a 585 μm × 585 μm × 5000 μm resolution. The images were ingested and propagated to lower resolutions using NeuroData infrastructure. NeuroData's registration module (ndreg) was then used to register each image to the Allen institute's mouse Reference Atlas (ARA).

First each image was reoriented to the ARA, the background was subtracted and a mask was generated to eliminate bright regions. After affine alignment, each Stanford image was deformably aligned to the ARA through Large Deformation Diffeomorphic Metric Mapping (LDDMM). Since the ARA and CLARITY images differed greatly in intensity profile, mutual information matching was adopted during this step. Alignment was done in a 3-step multi-resolution approach, with registration at coarser scales initializing the alignment at subsequent finer scales. It was clear from the ARA-CLARITY checkerboard composite images that registration proceeded successfully (Figure 3).

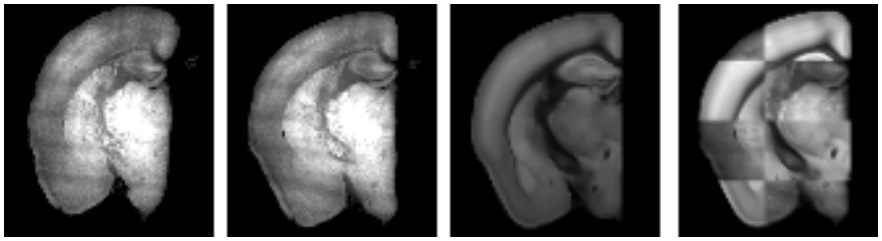


Figure 3: Coronal slices from image volumes. From left to right: CLARITY before LDDMM, CLARITY after LDDMM, ARA, checkerboard composite of ARA and CLARITY after LDDMM.

3.5 FlashX

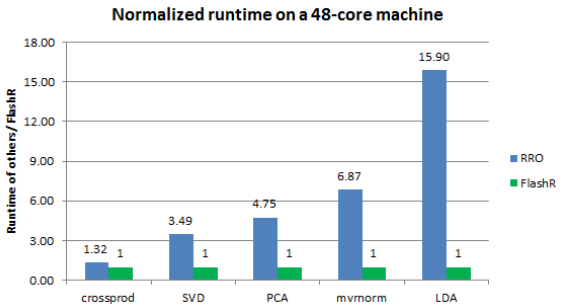


Figure 4: Normalized runtime of FlashR vs. Revolution R when executing R implementations of machine learning primitives on a dataset with one million data points and 1000 features on a large parallel machine with 48 CPU cores. FlashR outperforms Revolution R in all computations. When the computation gets more complex, the speed advantage of FlashR over Revolution R gets larger.

After having efficient matrix operations in the past months, including sparse matrix multiplication and various dense matrix operations, we integrate all matrix operations in a single computation framework called FlashR, which provides both high compatibility with R and efficiency. FlashR now overrides about 70 R matrix functions in the R base package. As such, we can run existing R code with little modification or no modification at all. For example, we ported a few R implementations of machine learning algorithms in the MASS package with little modification. We compare the speed of FlashR against Revolution R, which is also designed to parallelize and accelerate R code, on a large parallel machine with 48 CPU cores (Figure 4). Even for the simple matrix operation such as crossprod, FlashR outperforms Revolution R. As the computation gets more complex, the advantage of FlashR over Revolution R gets larger. When executing the LDA implementation (Linear Discriminant Analysis) in the MASS package, FlashR outperforms Revolution R by over an order of magnitude.

3.6 ndmg

In an effort to further verify that derivatives produced by the ndmg pipeline are high-quality and execution of a given dataset within the pipeline was successful, we have expanded upon a automatically generated set of quality control figures. In particular, we have developed the first, to our knowledge, connectome-specific graph quality control plot. This figure, shown as the "Degree" panel in Figure 5, considers the intra- and inter- hemispheric connectivity of the graph, and plots the degree of each node for both same and across hemispheric connectivity. Many real-world graphs contain node and edge attributes, but often location is not among them (or it is used to define the edges); in connectomics, location is an important feature of each node and plays a role in its connectivity patterns, as in whether a connection will exist within or across hemispheres of the brain. We also are computing the mean connectome for the given dataset in this summary figure.

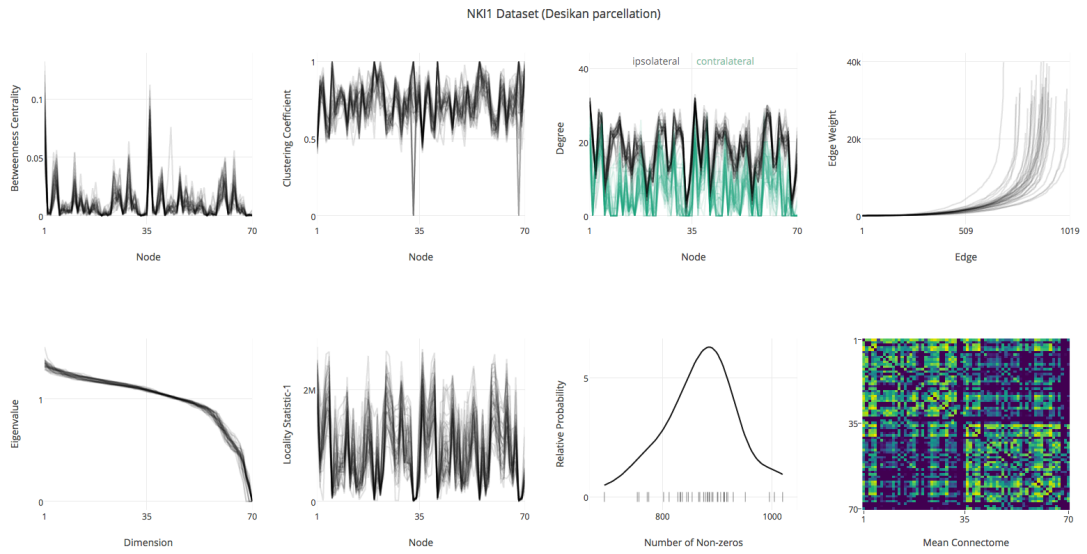


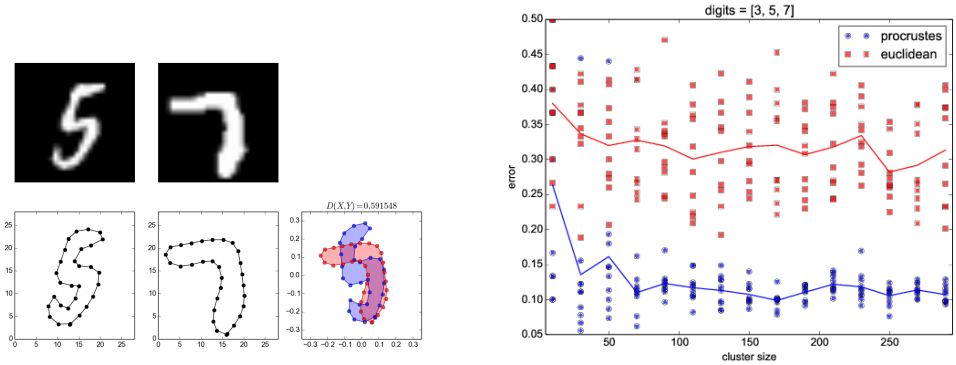
Figure 5: Quality Control of Connectomes

3.7 Non-Parametric Shape Clustering

We developed a prototype algorithm for 2-dimensional shape clustering, which is invariant under affine transformations. This employs the procrustes distance between two objects, which requires feature extraction to obtain landmark points; see Fig. 6 for an example. We intend to apply a variant of this method to analyze neural synapses, since we have such datasets from our collaborators. However, to this particular dataset, the preprocessing techniques may be highly sophisticated, and a new metric to compare different objects may be necessary. Moreover, these shapes are 3-dimensional. We are currently working on this project with our collaborators, extending our existing techniques to this particular dataset.
We also started working on a related, and more general, project. We intend to develop non-parametric clustering algorithms with statistical guarantees. We will use an energy-statistics based approach. Given two datasets X and Y, there is an energy function $\mathcal{E}(X, Y)$ test statistic which allows us to infer if X and Y have the same distribution. Our results thus far suggest that this can be written as a quadratic optimization problem with quadratic constraints:

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

where Δ is a dissimilarity data matrix. There is not enough literature on this interesting problem, so this will very likely lead to new methods which can have interesting applications, in particular to neuroscience datasets.



(a) MNIST digits with extracted landmarks and alignment.

(b) Classification error against the size of each cluster (the three classes have the same number of points) is shown in blue. The red line is standard K-means with Euclidean distance for comparison.

Figure 6: MNIST handwritten digits and classification error results.

3.8 Batch effect removal in dimension reduction of multiway array data

Batch effects are unwanted random variations caused by different data sources and experimental conditions. Generalized linear random effects model is effective to mitigate these confounders in traditional low dimensional data; however, there is a lack of such tool for high dimensional and multiway array data. While tensor factorization is routinely used for dimension reduction, due to the sharing of factors among all batches, the batch effects quickly populate the low dimensional core and confound the signal. In this research, we propose a different strategy by letting factor matrices vary over batches, while leaving the remaining variation in the core. This allows capturing sophisticated batch effects, while retaining the low rank structure for describing signal. To allow estimation with flexible factors, we utilize a hierarchical random effects model to borrow information among the batches. An efficient closed-form expectation conditional maxmization strategy is developed for rapid estimation. We focus the application on the joint diagonalization of brain connectivity data obtained from different sources.
The model we propose is:

$$
\begin{aligned}
A_{ji,kl} &= A_{ji,lk} \\
A_{ji,kl} &\overset{indep}{\sim} \text{Bern}(\text{logit}(\psi_{ji,kl})) \\
\psi_{ji,kl} &= \sum_{r=1}^{d} c_{ji,r} f_{j,kr} f_{j,lr} \\
f_{j,kr} &\overset{indep}{\sim} \text{N}(f_{0,kr}, \sigma^2) \\
f_{0,kr} &\overset{iid}{\sim} \text{N}(0,1)
\end{aligned}
\tag{2}
$$

with $k = 1 \dots l$ and $l = 2 \dots n$.
The batch effect adjusted connectome is then $A_{ji,kl} = \psi_{ji,kl} = \sum_{r=1}^{d} c_{ji,r} f_{0,kr} f_{0,lr}$

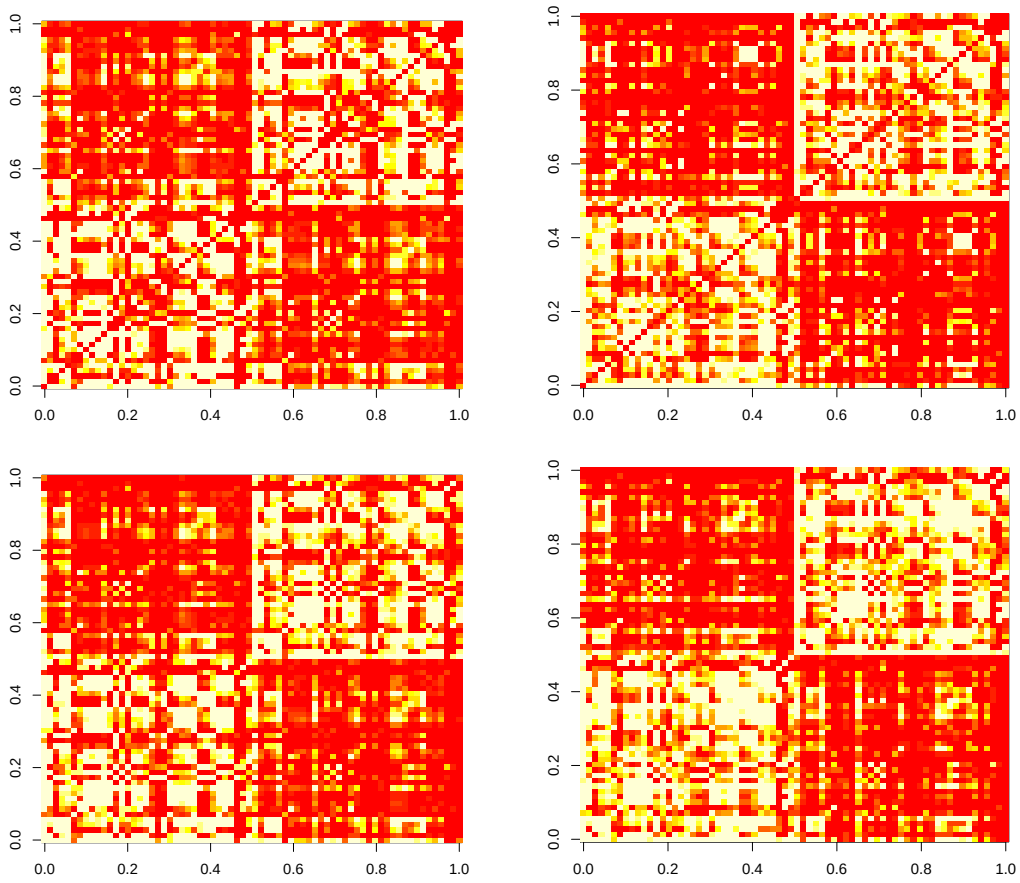


Figure 7: The first row shows the average connectome of two groups, that some difference can be obsevered. The second row shows the batch effect adjusted average connectome of the two groups, that they become similar.

3.9 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 the t[th] trial of subject i and $\Delta(,)$ be the metric, the (population) 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.
During the last month, we have been trying to finalize the paper. Specifically, we formalize three algorithms and write the pseudocode for the algorithms. The three algorithms are computing sample discriminability from test and retest data set 1, testing whether the data is better than random 2, and testing whether one processing pipeline is better than another 3. The pseudocode is provided below. For the R code and functions, please see neurodata/discriminability.
Furthermore, we carry out the Algorithm 3 on 13 fMRI data sets. Previously, we process 13 fMRI data sets with 64 pipelines, and show some pipelines yields data sets with high sample discriminability. However, we do not have a valid test to compare different pipelines other than rank them by mean discriminability. We use the algorithm 3 to carry out a two sample test to compare different pipelines for each data set. Specifically, all pipelines are compared to pipeline CFXSG. Then, a single p-values is calculated by applying Fisher's method to combine p-values from 13 sets. We group the pipelines by p-values and within each group the pipelines are ordered by mean discriminability. The result of rank graphs is shown in Figure 8. The result of raw graphs can also be found here. The new two sample test provide a better ordering of pipelines compared to the previous ordering by Wilcoxon signed-rank test.

Discriminability Algorithms

Algorithm 1 Compute discriminability estimate $\hat{D}$.

Require: Test-retest samples $\{\boldsymbol{x}_{i,t}\}$.
Ensure: Sample discriminability $\hat{D}$.
function ComputeDiscriminability
for i, t, i', t' **do** ▷ compute pairwise distances
$PD[i, t, i', t'] = \delta(\boldsymbol{x}_{i,t}, \boldsymbol{x}_{i',t'})$
Dsum = 0;
Count = 0;
for $i = 1, ..., n$ **do**
for $t = 1, ..., s$ **do**
$d =$ across subject distances to $\boldsymbol{x}_{i,t}$
for $t' = 1, ..., s$ and $t' \neq t$ **do**
$\hat{D}_{i,t,t'} = \sum_j \mathbf{I}\{PD[i, t, i, t'] < d[j]\}/Length(d)$ ▷ compare within and across distances
$Dsum = Dsum + \hat{D}_{i,t,t'}$
$Count = Count + 1$
$\hat{D} = Dsum/Count$ ▷ Sample Discrminability

Algorithm 2 The function returns a p-value for testing the null hypothesis that $D = 0.5$.

Require: Test-retest samples $\{\boldsymbol{x}_{i,t}\}$, the number of permutations np.
Ensure: The p-value $p \in [0, 1]$ for testing the hypothesis that $D = 0.5$.
function OneSampleTest
$\hat{D} = ComputeDiscriminability(\{\boldsymbol{x}_{i,t}\})$ ▷ compute true sample discriminability
for $j = 1, ..., np$ **do**
$\{\boldsymbol{x}_{i,t}^{(j)}\} = Permute(\{\boldsymbol{x}_{i,t}\})$ ▷ permute the subject labels
$d[j] = ComputeDiscriminability(\{\boldsymbol{x}_{i,t}^{(j)}\})$ ▷ compute discriminability of permuted samples
$p = \sum_j \mathbf{I}\{\hat{D} < d[j]\}/np$ ▷ compute p-value

Algorithm 3 The function returns a p-value for testing the null hypothesis that $D(\boldsymbol{\psi}_1) = D(\boldsymbol{\psi}_2)$.

Require: Raw data $\{f_\phi(\boldsymbol{v}_i)\}$, pipeline $\boldsymbol{\psi}_1$, pipeline $\boldsymbol{\psi}_2$, the number of bootstrapped samples nb.
Ensure: The p-value $p \in [0, 1]$ for testing the hypothesis that $D(\boldsymbol{\psi}_1) = D(\boldsymbol{\psi}_2)$.
function TwoSampleTest
$\{^1\boldsymbol{x}_{i,t}\} = g_{\boldsymbol{\psi}_1}(\{f_\phi(\boldsymbol{v}_i)\})$
$\{^2\boldsymbol{x}_{i,t}\} = g_{\boldsymbol{\psi}_1}(\{f_\phi(\boldsymbol{v}_i)\})$ ▷ process the raw data with two pipelines
$\hat{D}(\boldsymbol{\psi}_1) = ComputeDiscriminability(\{^1\boldsymbol{x}_{i,t}\})$
$\hat{D}(\boldsymbol{\psi}_2) = ComputeDiscriminability(\{^1\boldsymbol{x}_{i,t}\})$ ▷ compute sample discriminability estimates
for $j = 1, ..., nb$ **do**
for $i = 1, ..., n$ **do**
$i_1, i_2 = Sample(n)$ ▷ randomly select two subjects
$\lambda = SampleUniform$
for t = 1,...,s **do**
$^1\boldsymbol{x}_{i,t}^{(j)} = {}^1\boldsymbol{x}_{i_1,t}\lambda + {}^1\boldsymbol{x}_{i_2,t}(1-\lambda)$ ▷ Linearly combine two observations
$\hat{D}^{(j)}(\boldsymbol{\psi}_1) = ComputeDiscriminability(\{^1\boldsymbol{x}_{i,t}^{(j)}\})$
$\hat{D}^{(j)}(\boldsymbol{\psi}_2) = ComputeDiscriminability(\{^2\boldsymbol{x}_{i,t}^{(j)}\})$
$ind = 0$
for $j = 1, ..., nb$ **do** ▷ generate the null distribution
for $j' = i + 1, ..., nb$ **do**
$d[ind] = \hat{D}^{(j)}(\boldsymbol{\psi}_1) - \hat{D}^{(j')}(\boldsymbol{\psi}_1)$
$ind = ind + 1$
$d[ind] = \hat{D}^{(j)}(\boldsymbol{\psi}_2) - \hat{D}^{(j')}(\boldsymbol{\psi}_2)$
$ind = ind + 1$
$p = \sum_j \mathbf{I}\{\hat{D}(\boldsymbol{\psi}_1) - \hat{D}(\boldsymbol{\psi}_2) < d[j]\}/Length(d)$ ▷ compute p-value

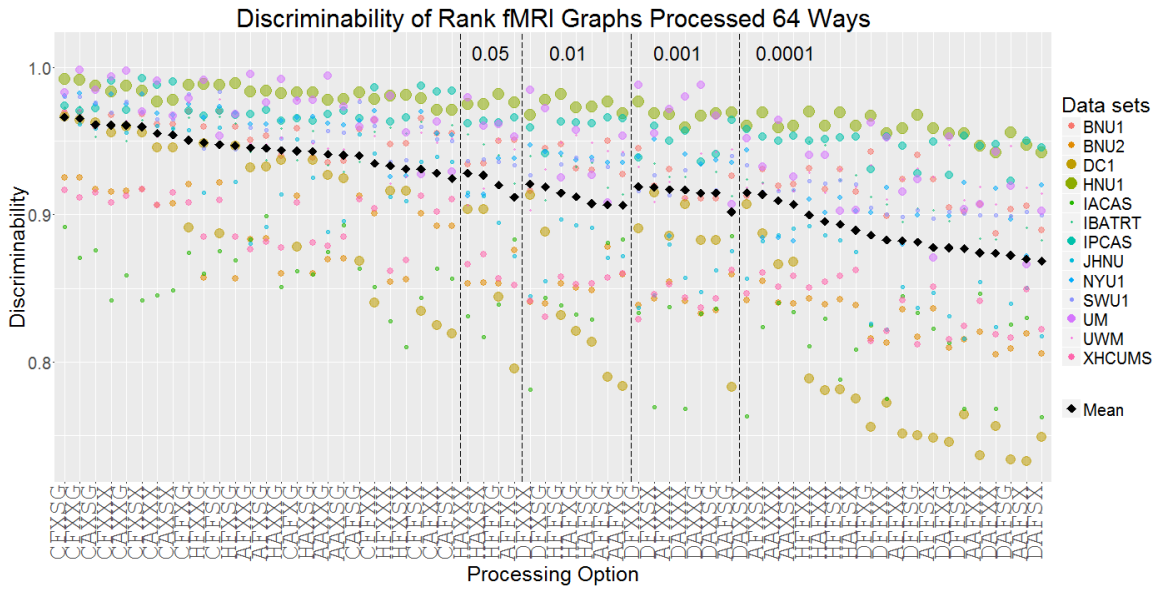


Figure 8: **Discriminability of rank fmri graphs from 13 data sets processed 64 ways.** Discriminability of BNU1, BNU2, DC1, HNU1, IACAS, IBATRT, IPCAS, JHNU, NYU1, SWU1, UM, UWM and XHCUMS pre-processed by 64 pipelines are shown in the plot. Color of each dot indicates data set and size indicates the number of measurements in data set. The black square indicates the weighted mean discriminability across 13 data sets. For each data set, all pipelines are compared to the pipeline CFXSG using the two sample test, and a single p-value is calculated by Fisher's method. The pipelines are grouped by p-values. The number at the top indicates the range of the p-values. Within each group, the pipelines are ordered by the mean discriminability. CFXSG pipeline has the best mean discriminability across data sets.

3.10 Law of Large Graphs

3.10.1 Low-rank Assumption Discussion

As we noted above, if the graphs are distributed according to an SBM or an RDPG, the relative efficiency is approximately invariant to the number of graphs M when N is large. If on the other hand, the graphs are generated according to a full-rank independent edge model, then the relative efficiency can change more dramatically as M changes. The reason for this is because for larger M, more of the eigenvectors of $\bar{A}$ will begin to concentrate around the eigenvectors of the mean graph. This leads to the fact that the optimal embedding dimension for estimating the mean will increase, making $\bar{A}$ and the low-rank approximation at the optimal dimension closer together. As a result, $\text{RE}(\bar{A}, \hat{P})$ will increase as M increases for full-rank models. Indeed, for large M we could have $\text{RE}(\bar{A}, \hat{P}) \geq 1$ since we cannot guarantee that $\hat{P}$ will choose the optimal dimension. The lack of gaps in the eigenvalues of the mean graph makes dimension reduction quite dangerous. In an extreme case, the low-rank assumption will be mostly violated when all eigenvalues of the mean graph are almost equal. This leads to a certain type of structure, which is close to a constant times the identity matrix. However we don't see such structure in connectomoics. Now we check this before applying our estimator to the CoRR dataset.

We first check if the dataset has the low-rank property. In Fig. 9, we plot the eigenvalues of the mean graph of all 454 graphs (with diagonal augmentation) in decreasing algebraic order for three different atlases. For all three atlases, the eigenvalues first decrease dramatically and then stay around 0. In addition, we also plot the histograms in Fig. 10. From the figures we can see many eigenvalues are around zero, with a few large eigenvalues. So the information is mostly contained in the first few dimensions. Such quasi low-rank property could be used by $\hat{P}$ to improve $\bar{A}$.

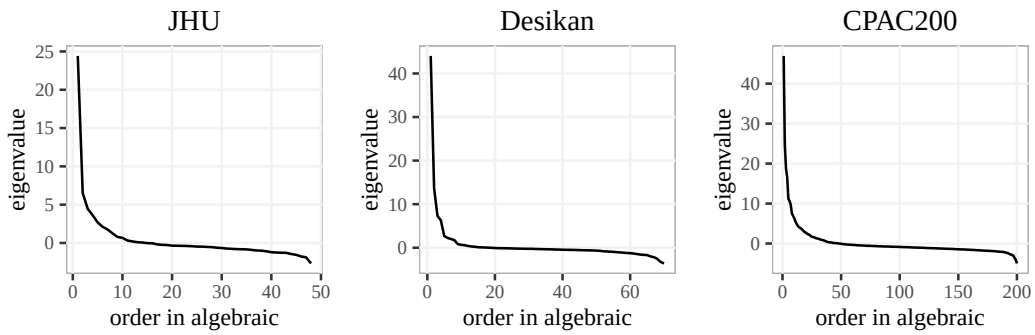


Figure 9: **Screeplot of the population mean.** These screeplots show the eigenvalues of the mean graph of all 454 graphs with diagonal augmentation in decreasing algebraic order for three atlases. Many eigenvalues are around zero, which lead to a quasi low-rank structure.

3.10.2 Interpretability of the Latent Positions

We also note that low-rank methods can often be more easily interpreted. By representing a low-rank matrix in terms of the latent position, where each vertex is represented as a vector

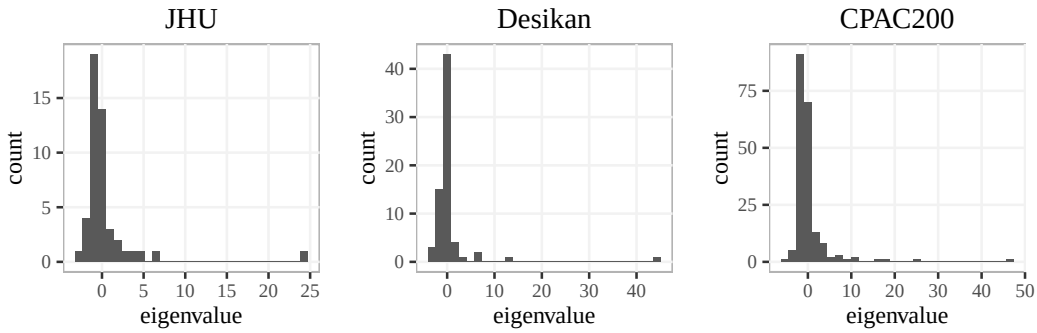


Figure 10: **Histogram of the population mean.** These figures show the histograms of the eigenvalues of the mean graph of all 454 graphs with diagonal augmentation. Many eigenvalues are around zero, which lead to a quasi low-rank structure.

in $\Re^d$ and the entries of the matrix are given by the inner products of these vectors, one can analyze and visualize the geometry of these vectors in order to interpret how each vertex is behaving in the context of the larger graph. Now we take the CoRR dataset experiment as an example and consider the same sample of size $M = 5$ based on the Desikan atlas. Our estimator $\hat{P}$ is based on the estimated latent positions $\hat{X} \in \mathbb{R}^{N \times d}$, where $N = 70$ is the number of vertices and $d = 11$ is the dimension selected by the Zhu and Ghodsi's method. Now we plot the heat map of $\hat{X}$ in Fig. 11, with each row to be the estimated latent position for the corresponding vertex. From the second column, we can see a clear distinction of the left and right hemisphere as conveyed in the second dimension. To have a more direct understanding of how they distinguish the two hemispheres, we color the brain using the 2nd dimension of $\hat{X}$ as in Fig. 12. Additionally, such a representation allows the use of techniques from multivariate analysis to further study the estimated population mean.

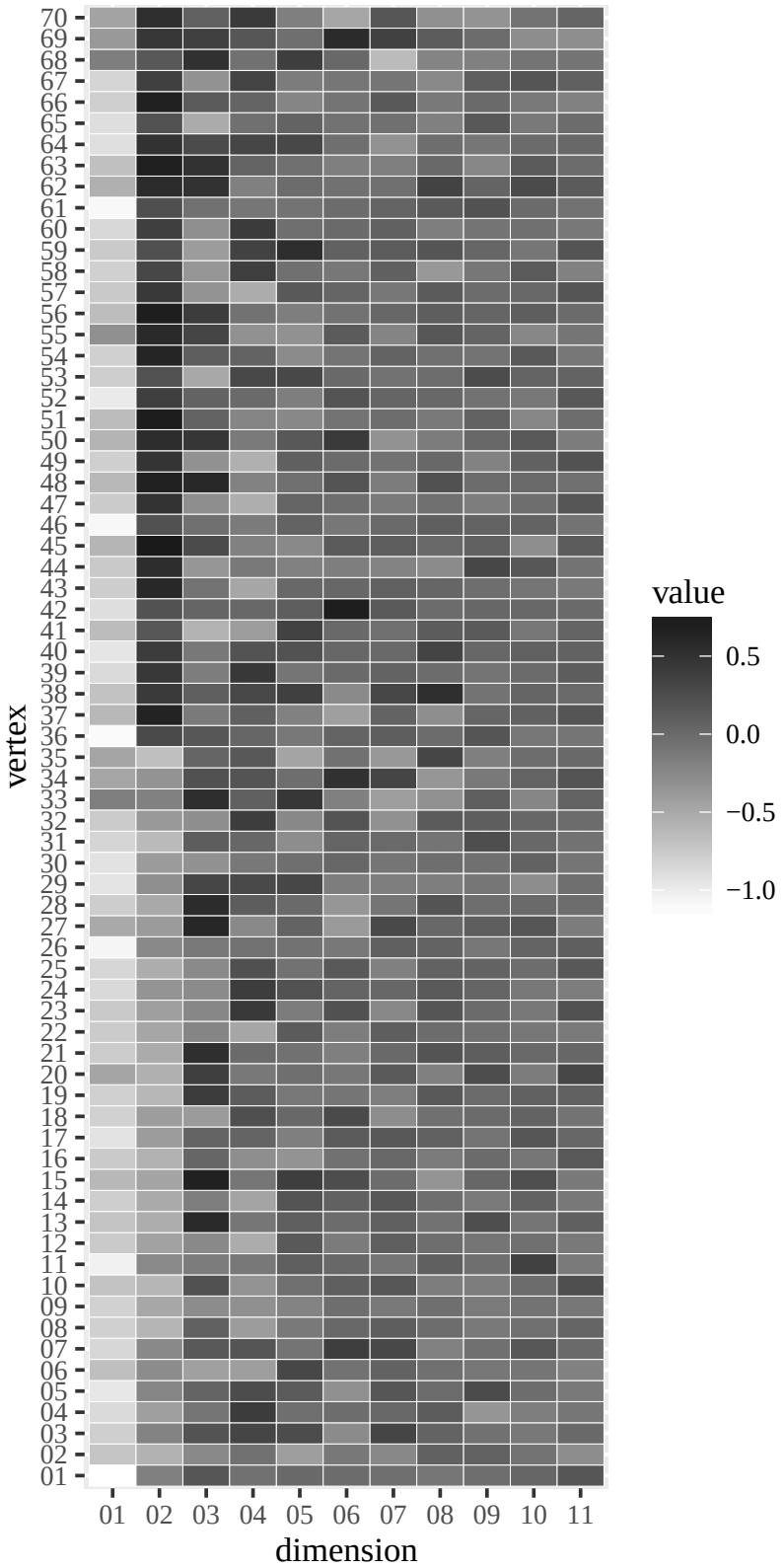


Figure 11: **Heat map of $\hat{X}$.** Heat map of $\hat{X}$ with each row to be the estimated latent position for the corresponding vertex. From the second column, we can see a clear distinction of the left and right hemisphere as conveyed in the second dimension.

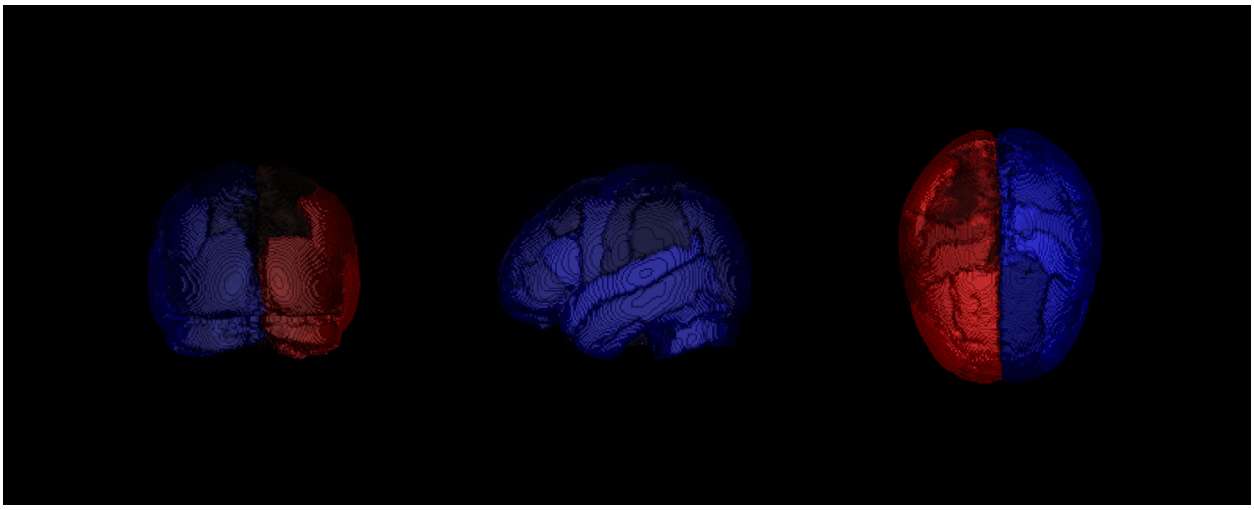


Figure 12: **Brain colored by the 2nd dimension of $\hat{X}$.** We plot the brain using the 2nd dimension of $\hat{X}$. From the figure, we can see a clear distinction of the left and right hemisphere as conveyed in the second dimension.

3.11 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. Previously we illustrated our theory and methods by Monte Carlo simulation. And now we focus on the real data experiment.

The real data we consider is a structural connectomic data. The graphs are based on diffusion tensor MR images. It contains 114 different brain scans, each of which was processed to yield an undirected, weighted graph with no self-loops, using the m2g/ndmg pipelines. The vertices of the graphs represent different regions in the brain defined according to an atlas. We used the desikan atlas with 70 vertices. The weight of an edge between two vertices represents the number of white-matter tract connecting the corresponding two regions of the brain. As we know, ndmg is a better pipeline compared to m2g, which means that the mean graph derived from ndmg should be a more accurate estimate to actual population mean graph. In order to evaluate the performance of the four estimators, we build estimates based on the samples from m2g, while using the sample mean graph from ndmg as an estimate of the probability matrix P. Specifically, each Monte Carlo replicate corresponds to sampling m graphs out of the 114 from the m2g dataset and computing the four estimates based on the m sampled graphs. We then compared these estimates to the sample mean for the 114 graphs from the ndmg dataset. We ran 100 simulations for the sample sizes $m = 2, 5, 10$. We also considered all possible dimensions for adjacency spectral embedding by ranging d from 1 to 70 in order to investigate the impact of the dimension selection procedures. We plot the result in figure 13

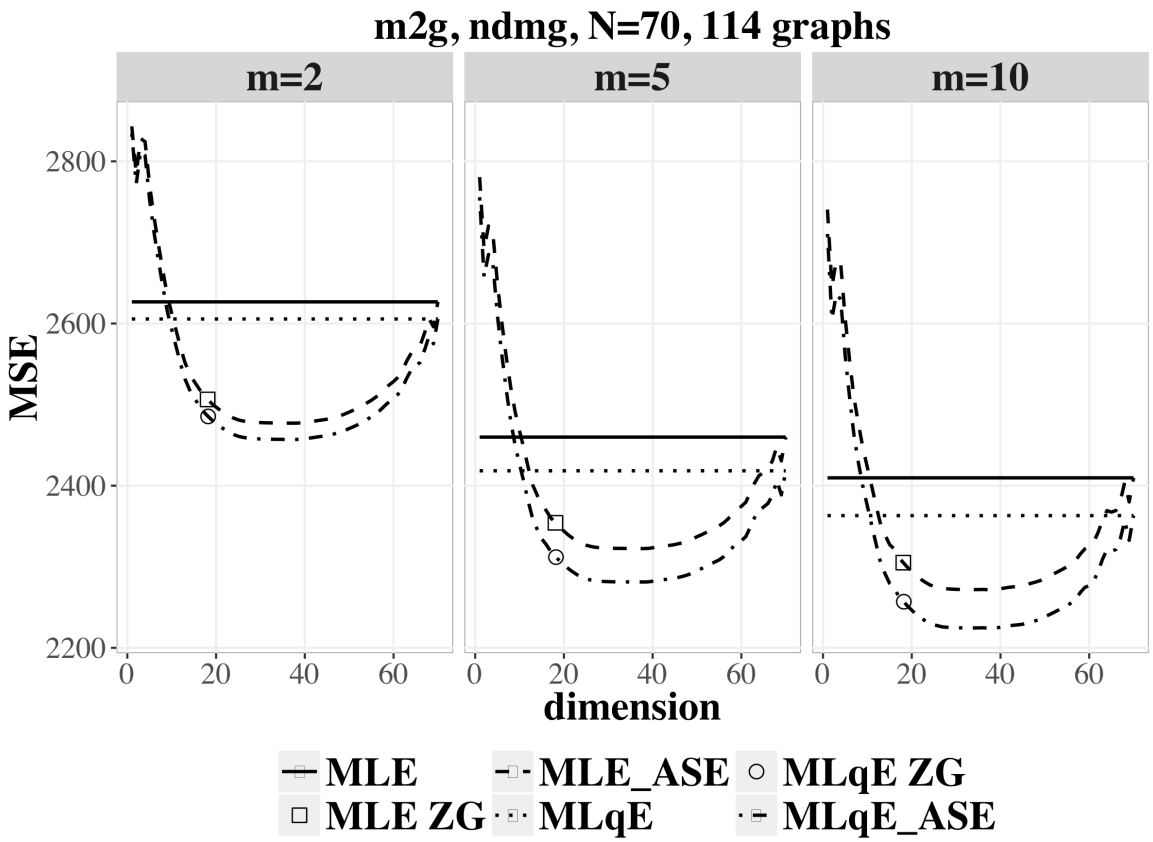


Figure 13: **Comparison of MSE of the four estimators for the desikan atlases at three sample sizes based on m2g and ndmg pipelines. 1. MLE (horizontal solid line) vs MLqE (horizontal dotted line):** MLqE outperforms MLE since robust estimators are always preferred in practice; **2. MLE (horizontal solid line) vs MLE_ASE (dashed line):** MLE_ASE wins the bias-variance tradeoff when embedded into a proper dimension; **3. MLqE (horizontal dotted line) vs MLqE_ASE (dashed dotted line):** MLqE_ASE wins the bias-variance tradeoff when embedded into a proper dimension; **4. MLqE_ASE (dashed dotted line) vs MLE_ASE (dashed line):** MLqE_ASE is better, since it inherits the robustness from MLqE. And the square and circle represent the dimensions selected by the Zhu and Ghodsi method. We can see it does a pretty good job. But more importantly, a wide range of dimensions could lead to an improvement.

3.12 LOL

As you may recall from last month, we successfully proved that LOL outperforms PCA essentially always when performing a linear dimensionality reduction prior to classification. This month we generate simulation results supporting those theoretical claims (see Figure 14).

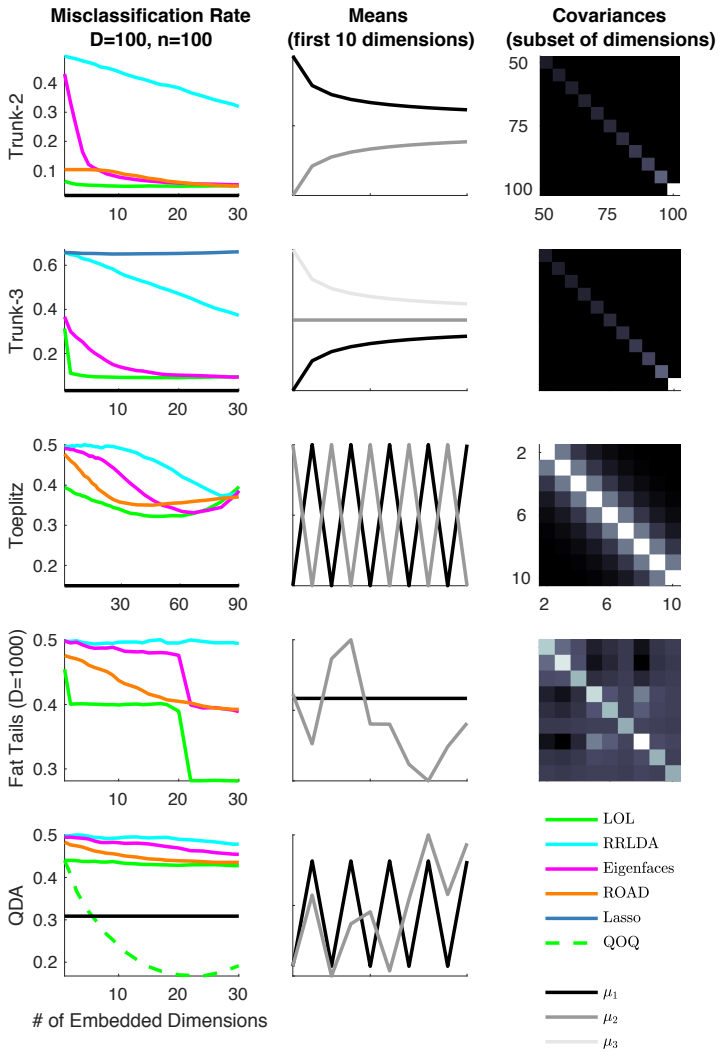


Figure 14: LOL outperforms other linear classifiers in a wide variety of settings, including those for which we have proven LOL should (rows 1 and 3), and those beyond our current theoretical grasp (rows 2, 4, and 5). This is true regardless of the dimensionality into which we embed.

3.13 Nonparametric Network Dependence Test

Deciphering the association between network structures and corresponding nodal attributes of interest is a core problem in network science. We propose a new nonparametric procedure for testing dependence between network topology and nodal attributes, via diffusion maps and `MGC`. Specifically, under an exchangeable graph, we verify that the diffusion maps provide a set of conditionally independent multivariate coordinates for the nodes, which can be combined with `MGC` (or in general, any distance-based correlation measures) to yield consistent statistic for network dependence testing. In simulation, the new approach achieves superior testing performance under a variety of common network models than existing benchmarks. The diffusion maps provides a robust metric compared to adjacency matrix or geodesic distance, while `MGC` can better capture nonlinear dependencies, with their combined advantages shown in Figure 15.

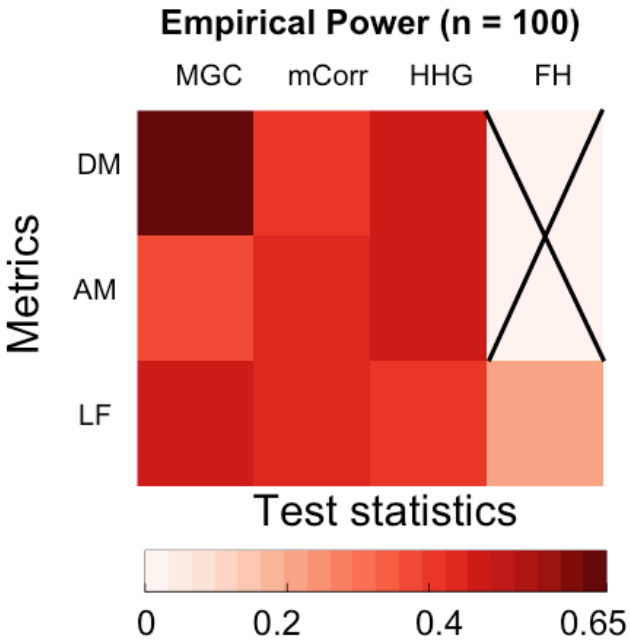


Figure 15: Power comparison for all possible combinations of metrics and correlation measure, under the stochastic block model with three blocks. `MGC` with the diffusion maps (DM) yields the best power, comparing to using other metrics like adjacency matrix (AM), latent factors (LF), and other test statistics like distance correlation (mcorr), Heller-Heller-Gorfine (HHG) test, or Fosdick and Hoff (FH) method.

This month we made significant progress in writing the manuscript and improving the exposition. The current draft was submitted to ASA Nonparametric Statistics Section Student Paper Awards, and we are notified as finalists for awards and special presentation section in the Joint Statistical Meeting this year.

3.14 Multiscale Generalized Correlation (MGC)

We developed the Multiscale Generalized Correlation method to better detect associations between two datasets X and Y. 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. A flowchart to illustrate the advantage of MGC is shown in Figure 16. The current draft is already uploaded to arXiv, and will be submitted soon. We collected valuable feedbacks from biologists this month, which resulted in improved and clarified texts and better flowchart. As the next step, we are investigating the required sample size for MGC to achieve perfect power, as an effort to better understand its mechanism and better apply the methodology to real data testing.

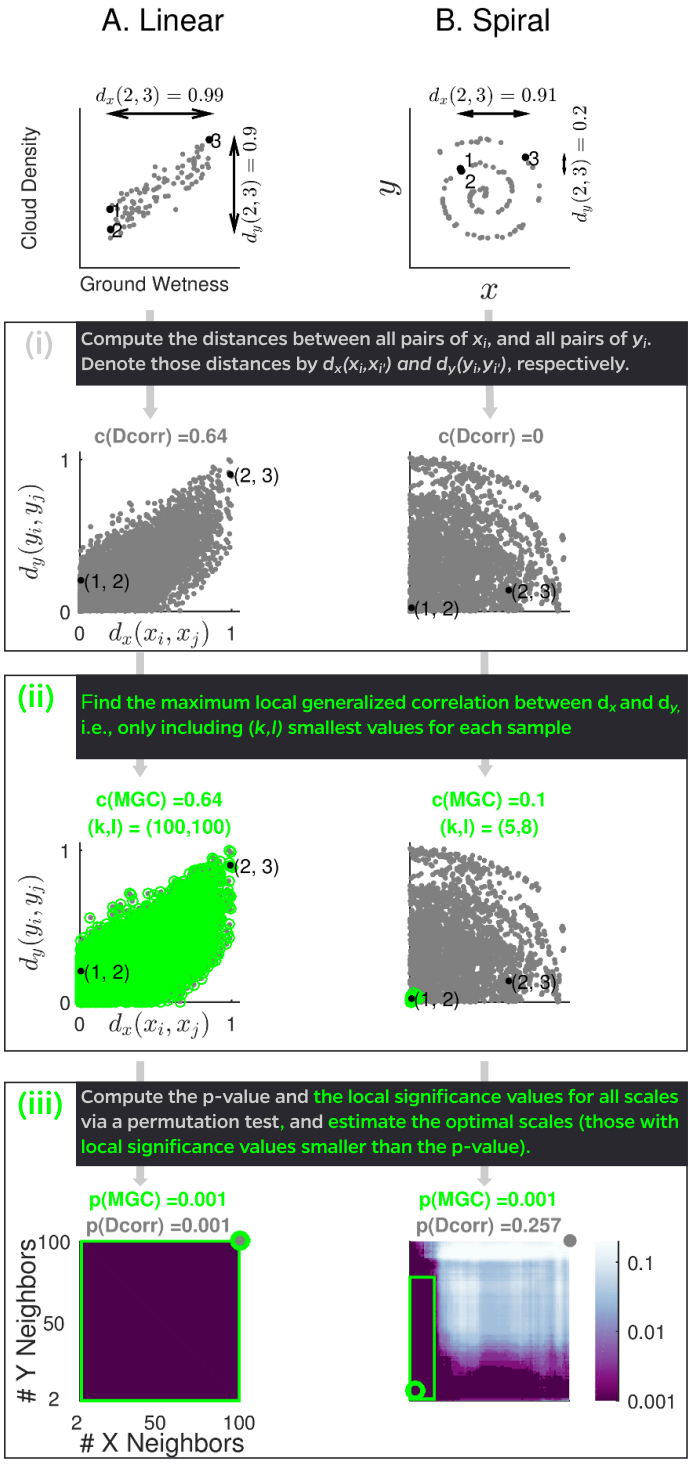


Figure 16: A flowchart to illustrate the advantages of MGC.

3.15 knor: K-means NUMA Optimized Routines

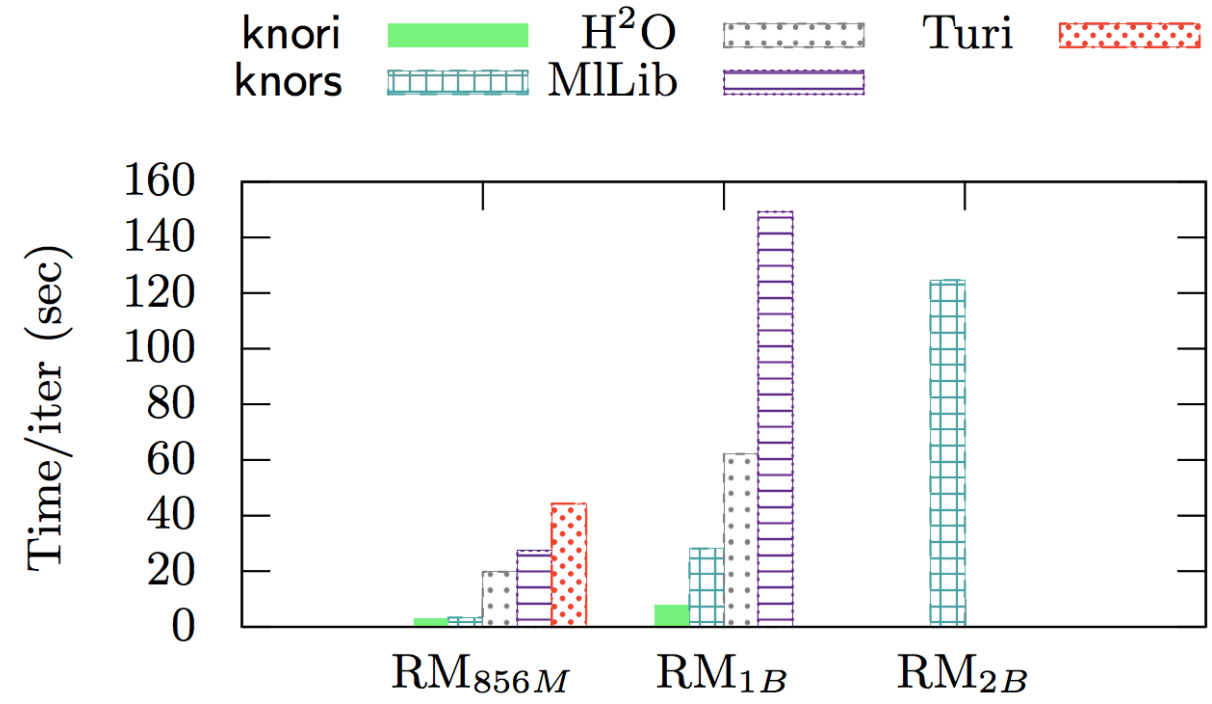
The **knor** library is a set of tools enabling users to perform the popular k-means algorithm at scale at speeds of 10x-100x time of that of popular frameworks in use today like Spark's MLlib, H^2O and GraphLab(Turi, Dato). We provide highly optimized routines for the following cases:

- Big-data that fits into the main memory (RAM) of a single machine, use **knori**.
- Big-data that fits into the aggregate main memory (RAM) of a cluster of distributed machines in the cloud, use **knord**.
- Big-data that cannot fit into the main memory (RAM) of a single machine, but can be placed out-of-corek, on disk, use **knors**.

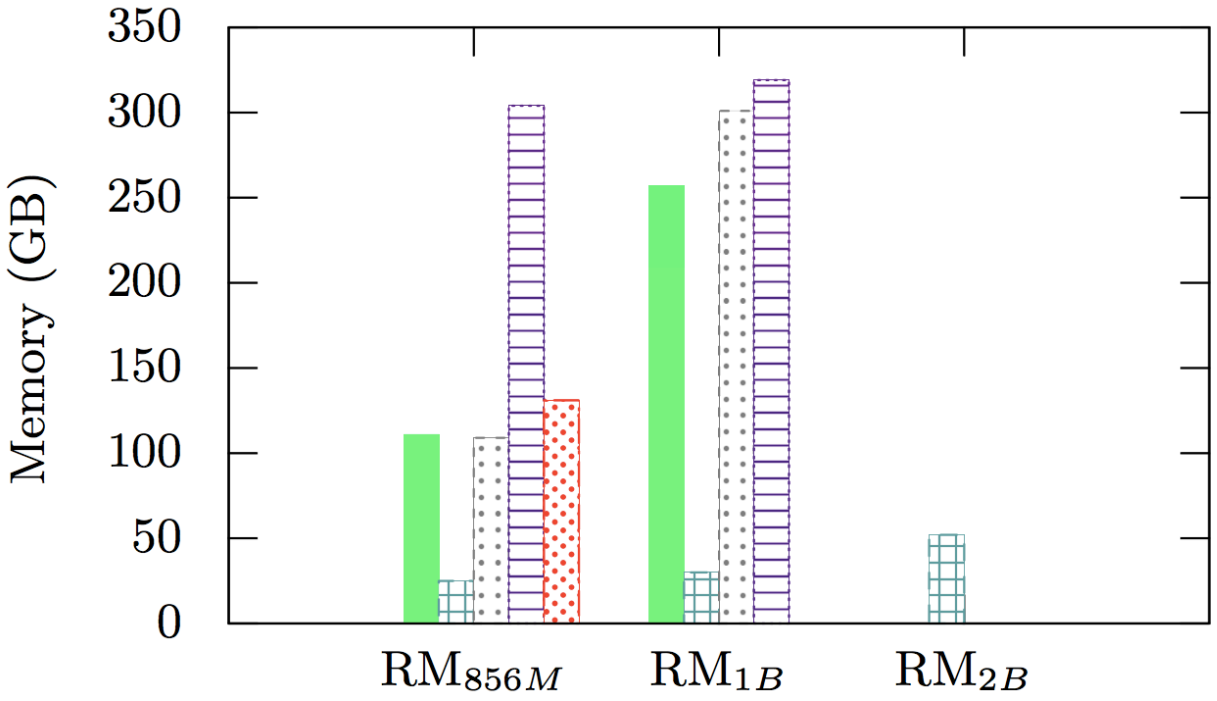
As part of efforts to make **knor** as user-friendly as possible we now support it on any platform via Docker containerization. We provide a one line installation procedure found at The repo's landing page. We further provide detailed instructions on how to use and extract data within the **knor** framework.

We recently submitted the **knor** paper to High-Performance Parallel and Distributed Computing (HPDC 2017) and publicly released it to arxiv.

Figure 17 compares the performance of our in-memory (**knori**) and our semi-external memory (**knors**) routines to our competitors. Distributed results (**knord**) can be found within the paper.



(a) Per iteration time elapsed of each routine.



(b) Memory consumption of each routine.

Figure 17: Speed and Memory comparison on randomly generated datasets (i) RM_{856M}, a 856 Million X 16 dataset of size 103GB and (ii) RM_{1B}, a 1 Billion X 32 dataset of size 251 GB and (iii) RM_{2B}, a 2 Billion X 64 dataset of size 1.1 TB. Turi is unable to run on RM_{1B} on our machine and only SEM routines are able to run on RU_{2B} on our machine with 48 Cores and 1 TB of RAM.

References

[1] N. Bhatla, R. Droste, S. R. Sando, A. Huang, and H. R. Horvitz, "Distinct neural circuits control rhythm inhibition and spitting by the myogenic pharynx of c. elegans," Current Biology, vol. 25, no. 16, pp. 2075 – 2089, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960982215007460

[2] D. D. Bock, W.-C. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid, "Network anatomy and in vivo physiology of visual cortical neurons," Nature, vol. 471, no. 7337, pp. 177–182, 03 2011.

[3] K. M. Harris, J. Spacek, M. E. Bell, P. H. Parker, L. F. Lindsey, A. D. Baden, J. T. Vogelstein, and R. Burns, "A resource from 3D electron microscopy of hippocampal neuropil for user training and tool development," Scientific Data, vol. 2, p. 150046, Aug 2015.

[4] N. Kasthuri, K. J. Hayworth, D. R. Berger, R. L. Schalek, J. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, T. R. Jones, M. Roberts, J. L. Morgan, J. C. Tapia, H. S. Seung, W. G. Roncal, J. T. Vogelstein, R. Burns, D. L. Sussman, C. E. Priebe, H. Pfister, and J. W. Lichtman, "Saturated reconstruction of a volume of neocortex," Cell, vol. 162, pp. 648–661, 05 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cell.2015.06.054

[5] W.-c. A. Lee, V. Bonin, M. Reed, B. J. Graham, G. Hood, K. Glattfelder, and R. C. Reid, "the visual cortex," Nature, vol. 532, no. 7599, pp. 370–374, 2016. [Online]. Available: http://dx.doi.org/10.1038/nature17192

[6] T. Ohyama, C. M. Schneider-Mizell, R. D. Fetter, J. V. Aleman, R. Franconville, M. Rivera-Alba, B. D. Mensh, K. M. Branson, J. H. Simpson, J. W. Truman, A. Cardona, and M. Zlatic, "A multilevel multimodal circuit enhances action selection in drosophila," Nature, vol. 520, no. 7549, pp. 633–639, 04 2015.

[7] S.-y. Takemura, A. Bharioke, Z. Lu, A. Nern, S. Vitaladevuni, P. K. Rivlin, W. T. Katz, D. J. Olbris, S. M. Plaza, P. Winston, T. Zhao, J. A. Horne, R. D. Fetter, S. Takemura, K. Blazek, L.-A. Chang, O. Ogundeyi, M. A. Saunders, V. Shapiro, C. Sigmund, G. M. Rubin, L. K. Scheffer, I. A. Meinertzhagen, and D. B. Chklovskii, "A visual motion detection circuit suggested by drosophila connectomics," Nature, vol. 500, no. 7461, pp. 175–181, 08 2013. [Online]. Available: http://dx.doi.org/10.1038/nature12450

[8] E. B. Bloss, M. S. Cembrowski, B. Karsh, J. Colonell, R. D. Fetter, and N. Spruston, "Structured dendritic inhibition supports branch-selective integration in ca1 pyramidal cells," Neuron, vol. 89, no. 5, pp. 1016 – 1030, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0896627316000544

[9] F. Collman, J. Buchanan, K. D. Phend, K. D. Micheva, R. J. Weinberg, and S. J. Smith, "Mapping synapses by conjugate light-electron array tomography," The Journal of Neuroscience, vol. 35, no. 14, pp. 5792–5807, 2015.

[10] N. C. Weiler, F. Collman, J. T. Vogelstein, R. Burns, and S. J. Smith, "Molecular architecture of barrel column synapses following experience-dependent plasticity." Nature Scientific Data, 2014.

[11] J. Freeman, N. Vladimirov, T. Kawashima, Y. Mu, N. J. Sofroniew, D. V. Bennett, J. Rosen, C.-T. Yang, L. L. Looger, and M. B. Ahrens, "Mapping brain activity at scale with cluster computing," Nature Methods, no. July, jul 2014. [Online]. Available: http://www.nature.com/doifinder/10.1038/nmeth.3041

[12] E. L. Dyer, W. Gray 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", "2016".

[13] O. Randlett, C. L. Wee, E. A. Naumann, O. Nnaemeka, D. Schoppik, J. E. Fitzgerald, R. Portugues, A. M. B. Lacoste, C. Riegler, F. Engert, and A. F. Schier, "Whole-brain activity mapping onto a zebrafish brain atlas," Nat Meth, vol. 12, no. 11, pp. 1039–1046, 11 2015.

[14] K. S. Kutten, J. T. Vogelstein, N. Charon, L. Ye, K. Deisseroth, and M. I. Miller, "Deformably Registering and Annotating Whole CLARITY Brains to an Atlas via Masked LDDMM," in Proceedings SPIE 9896, Optics, Photonics and Digital Technologies for Imaging Applications IV, P. Schelkens, T. Ebrahimi, G. Cristóbal, F. Truchetet, and P. Saarikko, Eds., 2016.

[15] G. Grabner, A. L. Janke, M. M. Budge, D. Smith, J. C. Pruessner, and D. L. Collins, "Symmetric atlasing and model based segmentation: An application to the hippocampus in older adults," in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2006, 2006, pp. 58–66.