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

Invited Talks

Conferences

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Unpublished	AT	M. musculus	16	1	24	1	29	3	23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.

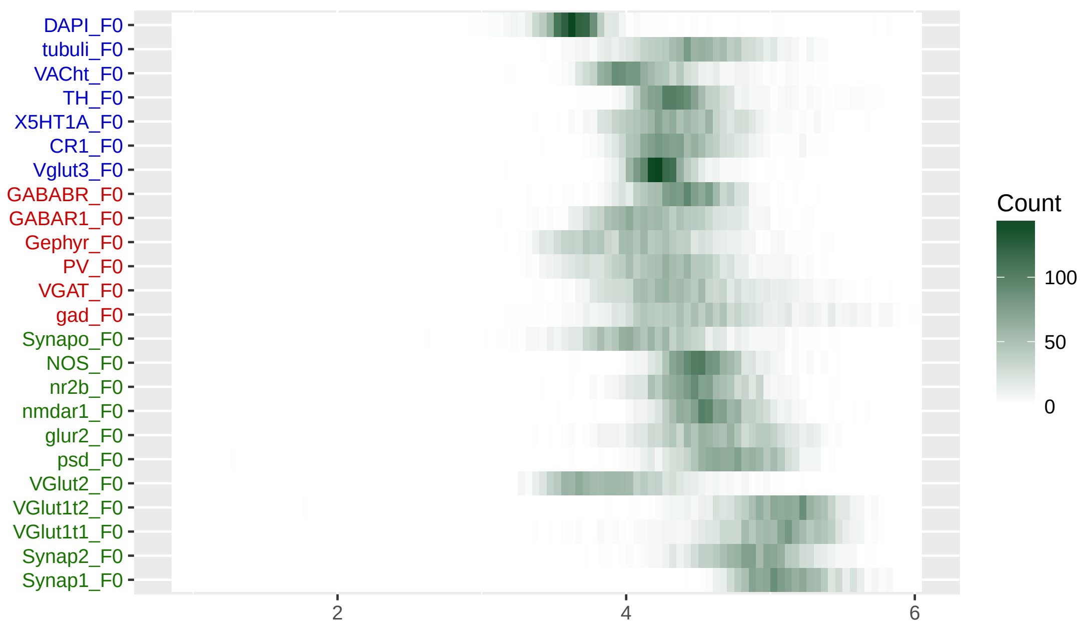


Figure 1: A one dimensional heatmap for each of the $\log_{10}$ transformed feature columns of the Kristina15 synaptome dataset. Colors correspond to the count of data points falling in to each bin. Scott's binning streategy is used which, in this case, yields 105 bins of equal width $w = 0.05$. Label 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 Graph Explorer

3.3 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 don't deal with incommensurate predictors, aren't 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. Additionally, oblique methods tend to be more sensitive to data corruption than RF is. Our proposed method, which we call RerF, seeks to address these limitations.

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.

Classification performances of RF, RerF, RR-RF, and XGBoost were compared 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. Therefore, a negative value indicates that an algorithm had a lower error rate than RF. 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.

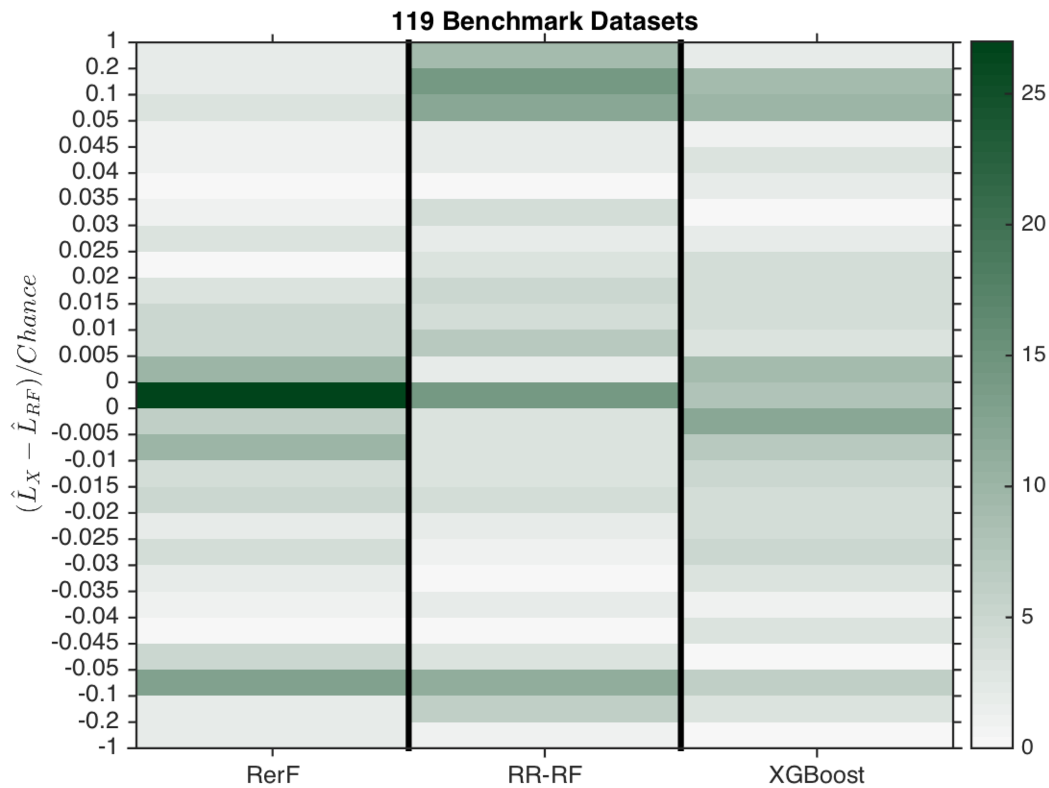


Figure 2: Classification performances of RF, RerF, RR-RF, and XGBoost on 119 benchmark datasets.

3.4 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.

3.4.1 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.

3.5 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 $\mu m \times$ 585 $\mu m \times$ 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).

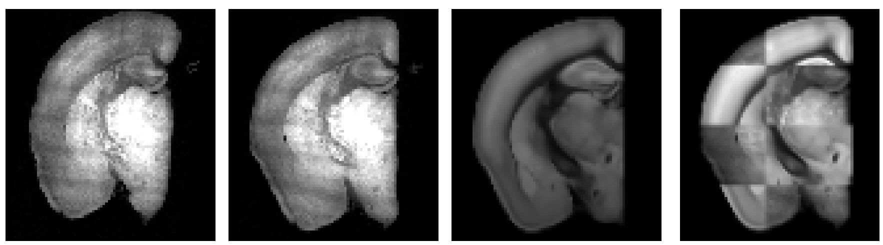


Figure 3: Coronal slices from image volumes. From left to right: CLARITY before LDDMM, CLARITY after LDDMM, ARA, checkerboard composite of ARA and CLARITY after LDDMM.

3.6 ndramondb

3.7 ndviz

3.8 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 4, 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.

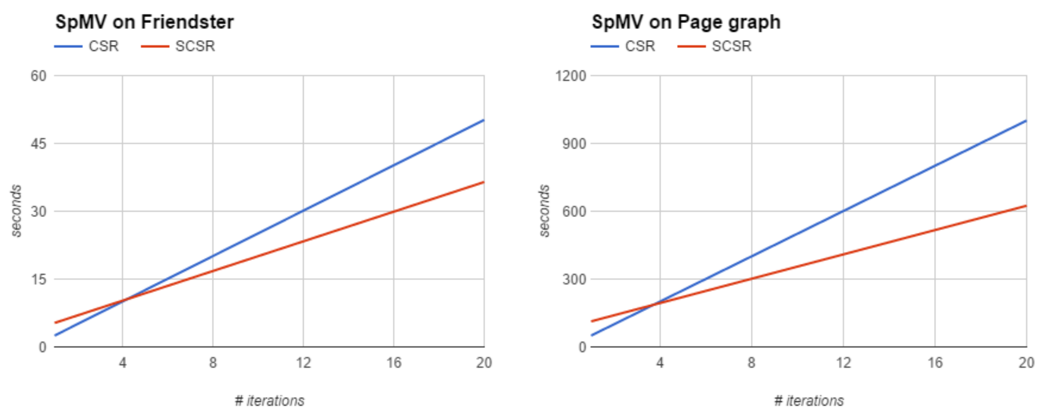


Figure 4: Benefits of using SCSR.

The performance ratio of the in-memory and semi-external memory sparse matrix multiplication in FlashX is affected by multiple factors. Shown in Figure 5, 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 un-clustered 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.
We implement Daniel Lee's NMF algorithm with our semi-external memory sparse matrix multiplication (denoted as SEM-NMF) and evaluate its performance by comparing it with a high-performance NMF implementation SmallK on billion-scale graphs. The dense matrices for NMF can be as large as the sparse matrix. As such, we split the dense matrix vertically so that each of the vertical partitions can fit in memory. We also evaluate the effect of the memory size on the performance of SEM-NMF by varying the number of columns in memory from the dense matrices. In the experiment, we factorize each of the graphs into two $n \times 16$ non-negative dense matrices.
We significantly improve the performance of SEM-NMF by keeping more columns of the input dense matrix in memory (Figure 6). The performance improvement is more significant when

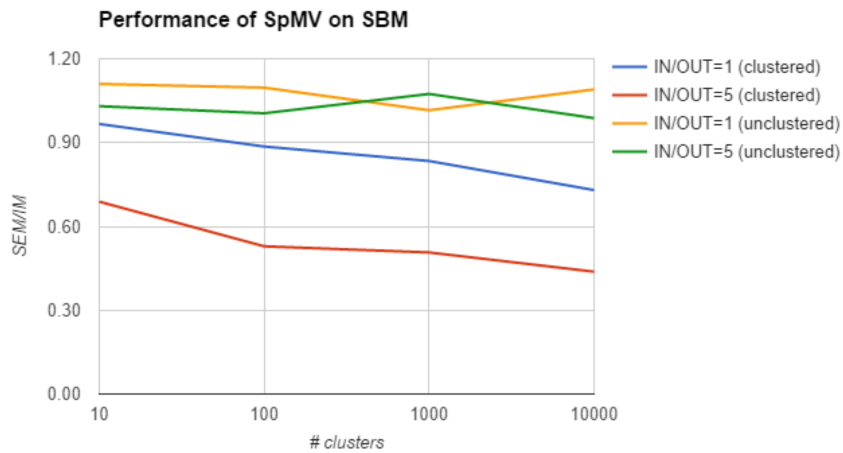


Figure 5: Benefits of using SCSR.

the number of columns that fit in memory is small. When we keep eight columns of the input dense matrix in memory, SEM-NMF achieves over 60% of the performance of its in-memory execution.

SEM-NMF significantly outperforms SmallK and other NMF implementations in the literature. SmallK is the closest competitor. We run the same NMF algorithm in SmallK and SEM-NMF outperforms SmallK by a large factor on all graphs (Figure 6). There are many MapReduce implementations in the literature. They run on sparse matrices with tens of millions of non-zero entries but generally take one or two orders of magnitude more time than our SEM-NMF on the sparse matrices with billions of non-zero entries.

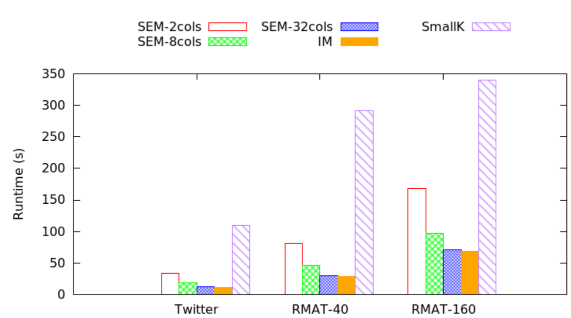


Figure 6: Performance improvements of SEM-NMF.

In the past months, we significantly advanced FlashR in both 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 de-

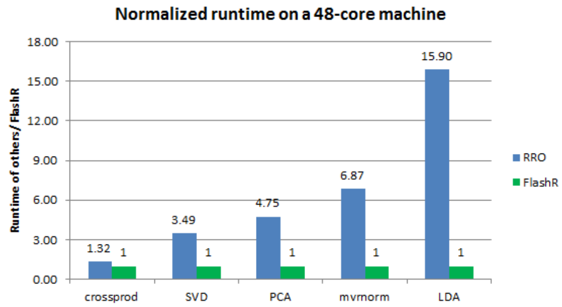


Figure 7: Normalized runtime of FlashR vs. RRO on a dataset with one million data points and 1000 features when running on a large parallel machine with 48 CPU cores.

signed to parallelize and accelerate R code, on a large parallel machine with 48 CPU cores (Figure 7). 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 an order of magnitude.

3.9 FlashY

3.10 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 8, 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.

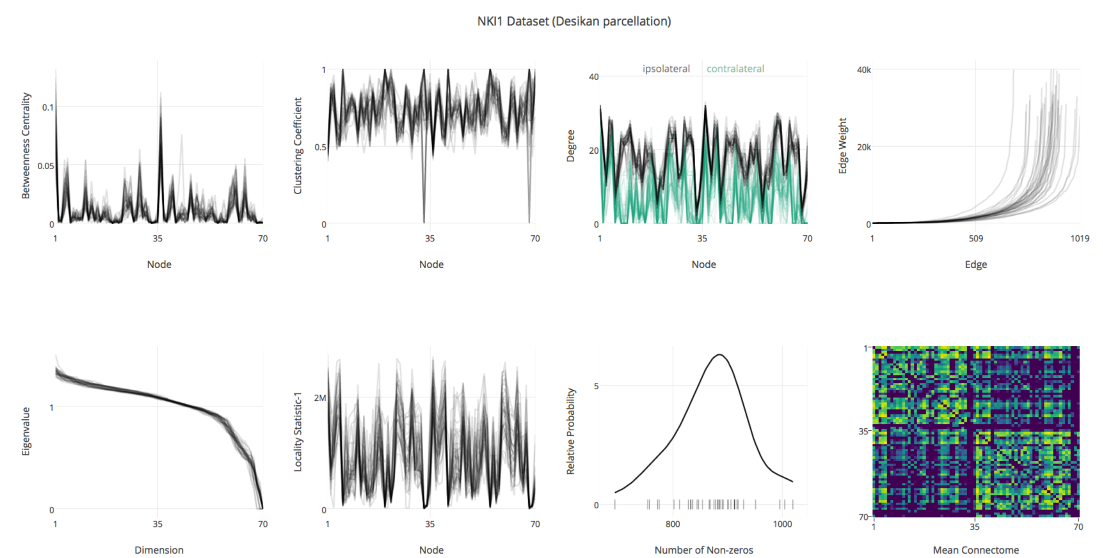


Figure 8: Quality Control of Connectomes

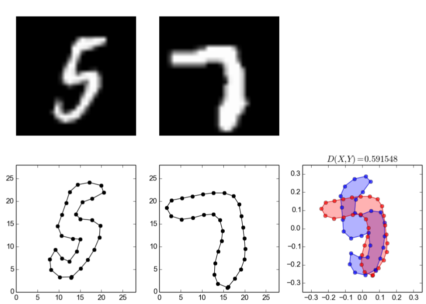
3.11 Non-Parametric Shape Clustering

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. 9 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.

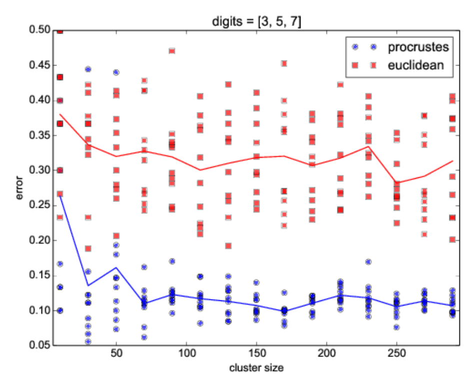
We also started working on a related, and more general, project. We intend to develop non-parametric clustering algorithms with statistical guarantees. We will use an energy-statistics based approach. Given two datasets X and Y, there is an energy function $\mathcal{E}(X,Y)$ test statistic which allows 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 9: MNIST handwritten digits and classification error results.

3.12 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. Our model show substantial gains in the simulations and application over the alternatives.

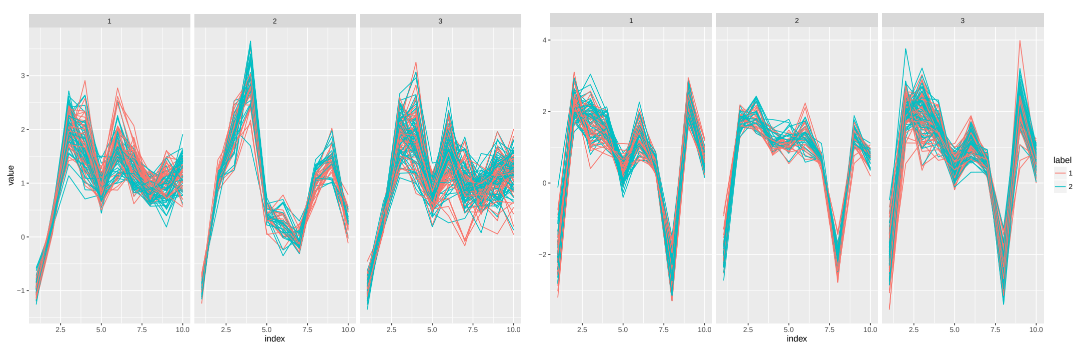


Figure 10: The batch effect removing model reduce the group random variation of the second group, and makes it more similar to the others.

3.13 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. In 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, test whether the data is better than random 2, and test whether one processing pipeline is better than another 3. The pseudocode is provided below. For the R code and functions, please see neurodata/discriminability.
Previously, we process 13 fMRI data sets with 64 pipelines, and show some pipelines yields data sets with high sample discriminability. We use the algorithm 3 to carry out a test to compare different pipelines for each data set. Then, a single p-values is calculated by applying Fisher's method to combine p-values from multiple data sets. The result is shown in the Figure 11. The new two sample test provide a smoother ordering of pipelines compared to 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**
          $\hat{D}_{i,t,t'} = \sum_j \mathbb{I}\{PD[i,t,i,t'] < d[j]\}/Length(d)$ ▷ 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
    $p = \sum_j \mathbb{I}\{\hat{D} < d[j]\}/np$ ▷ 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**
        $d[ind] = \hat{D}^{(j)}(\boldsymbol{\psi}_1) - \hat{D}^{(j')}(\boldsymbol{\psi}_1)$
        $ind = ind + 1$
        $d[ind] = \hat{D}^{(j)}(\boldsymbol{\psi}_2) - \hat{D}^{(j')}(\boldsymbol{\psi}_2)$
        $ind = ind + 1$
    $p = \sum_j \mathbb{I}\{\hat{D}(\boldsymbol{\psi}_1) - \hat{D}(\boldsymbol{\psi}_2) < d[j]\}/Length(d)$ ▷ compute p-value

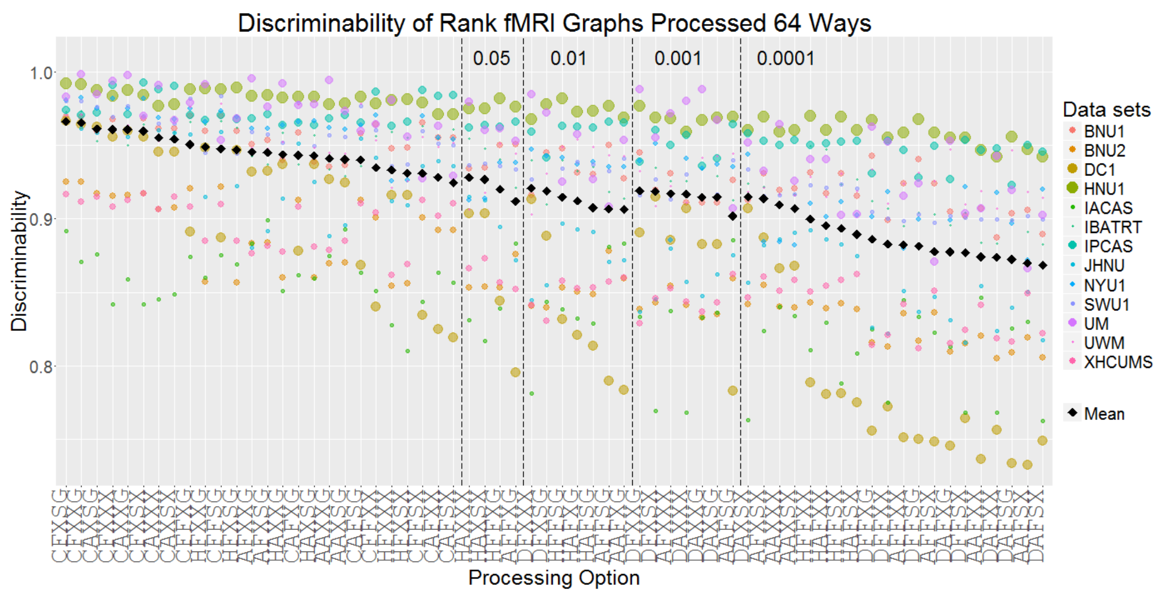


Figure 11: **Discriminability of rank fmri graphs from 13 data sets processed 64 ways.** Discriminability of BNU1, BNU2, DC1, HNU1, IACAS, IBATRT, IPCAS, JHNU, NYU1, SWU1, UM, UWM and XHCUMS pre-processed by 64 pipelines are shown in the plot. Color of each dot indicates data set and size indicates the 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.

3.14 Law of Large Graphs

We have submitted it to PLOS Computational Biology. And now here on the arXiv is our first step: http://arxiv.org/abs/1609.01672

Estimating the mean of a population of graphs based on a sample is a core problem in network science. Often, this problem is especially difficult because the sample or cohort size is relatively small as compared to the number of parameters to estimate. While using the element-wise sample mean of the adjacency matrices is a common approach, this method does not exploit any underlying graph structure. We propose using a low-rank method together with tools for dimension selection and diagonal augmentation to improve performance over the naive methodology for small sample sizes. Theoretical results for the stochastic blockmodel show that this method will offer major improvements when there are many vertices. Similarly, in analyzing human connectome data, we demonstrate that the low-rank methods outperform the standard sample mean for many settings. These results indicate that low-rank methods should be a key part of the tool box for researchers studying populations of graphs.

3.15 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 12

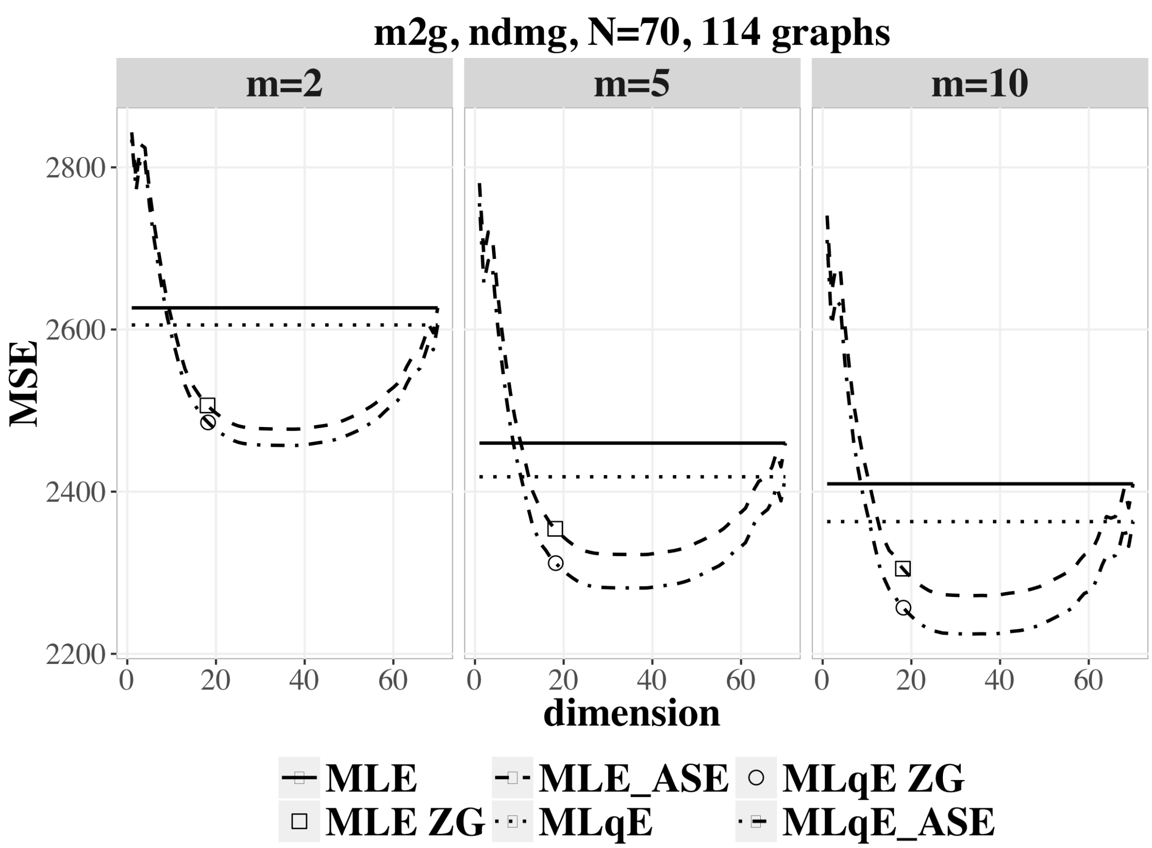


Figure 12: **Comparison of MSE of the four estimators for the desikan atlases at three sample sizes based on m2g and ndmg pipelines. 1. MLE (horizontal solid line) vs MLqE (horizontal dotted line):** MLqE outperforms MLE since robust estimators are always preferred in practice; **2. MLE (horizontal solid 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.

3.16 LOL

3.17 Multiscale Network Test

3.18 Multiscale Generalized Correlation (MGC)

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.