NeuroData SIMPLEX Report

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 **3**
- 1.1 New Datasets to the Cloud . . . 3

2 Tools **4**
- 2.1 neurodata.io . . . 4
- 2.2 meda . . . 4
- 2.3 ndstore . . . 4
- 2.4 ndreg . . . 6
- 2.5 ndramondb . . . 8
- 2.6 ndviz . . . 9
- 2.7 Graph Explorer . . . 9
- 2.8 FlashX . . . 9
- 2.9 FlashY . . . 13
- 2.10 ndmg . . . 13
- 2.11 Science in the Cloud (SIC) . . . 15
- 2.12 Non-Parametric Shape Clustering . . . 16
- 2.13 Reduced Dimension Clustering . . . 17
- 2.14 Randomer Forest (RerF) . . . 17
- 2.15 Discriminability . . . 19
- 2.16 Law of Large Graphs . . . 22
- 2.17 Robust Law of Large Graphs . . . 23
- 2.18 LOL . . . 25
- 2.19 Multiscale Network Test . . . 26
- 2.20 Multiscale Generalized Correlation (MGC) . . . 27

Manuscripts

[1] S. Chen, K. Liu, Y. Yang, Y. Xu, S. Lee, M. Lindquist, B. S. Caffo, and J. T. Vogelstein, "An M-Estimator for Reduced-Rank High-Dimensional Linear Dynamical System Identification," Pattern Recognition Letters, 2016.

[2] N. Binkiewicz, J. T. Vogelstein, and K. Rohe, "Covariate assisted spectral clustering," Accepted for publication in Biometrika, 2016.

[3] T. M. Tomita, M. Maggioni, and J. T. Vogelstein, "Roflmao: Robust oblique forests with linear matrix operations," in Proceedings of the 2017 SIAM International Conference on Data Mining. SIAM, 2017.

[4] J. T. Vogelstein, K. Amunts, A. Andreou, D. Angelaki, G. Ascoli, C. Bargmann, R. Burns, C. Cali, F. Chance, M. Chun et al., "Grand Challenges for Global Brain Sciences," F1000Research, 2016.

[5] N. C. Consortium et al., "To the cloud! a grassroots proposal to accelerate brain science discovery," Neuron, vol. 92, no. 3, pp. 622–627, 2016.

[6] A. K. Simhal, C. Aguerrebere, F. Collman, J. T. Vogelstein, K. D. Micheva, R. J. Weinberg, S. J. Smith, and G. Sapiro, "Probabilistic fluorescence-based synapse detection," arXiv preprint arXiv:1611.05479, 2016.

[7] G. Kiar, K. J. Gorgolewski, D. Kleissas, W. G. Roncal, B. Litt, B. Wandell, R. A. Poldrack, M. Wiener, R. J. Vogelstein, R. Burns et al., "Science in the cloud (sic): A use case in mri connectomics," arXiv preprint arXiv:1610.08484, 2016.

Invited Talks

[1] J. T. Vogelstein, "Discovering relationships across disparate data modalities." Brown University, 2016, invited talk.

[2] ——, "The international brain station," Coordinating Global Brain Projects, Sep 2016, invited talk.

[3] ——, "Optimal decisions for discovery science." Brainhacks Vienna, Sep 2016, invited talk.

[4] ——, "Neuroinformatics social," SfN 2016 Annual Meeting, Nov 2016, invited talk.

[5] ——, "Nano-scale neurocartography," SfN 2016 Annual Meeting (Mini-Symposium), Nov 2016, invited talk.

[6] K. Lillany, "Neurodata.io," Human Brain Project Satellite Symposium to SfN 2016., Nov 2016, invited talk.

Conferences

[1] A. D. Baden, F. Collman, K. Lillany, J. T. Vogelstein, and R. Burns, "Storing and analyzing large array tomography datasets in the cloud," SfN 2016 Annual Meeting Poster Session, Nov 2016.

[2] W. R. Gray Roncal, J. Matelsky, M. Pekala, D. Kleissas, J. T. Vogelstein, and G. D. Hager, "ndparse: tools and interfaces for scalable neuroscience discovery," SfN 2016 Annual Meeting Poster Session, Nov 2016.

[3] A. K. Simhal, C. Agguerrebere, F. Collman, J. T. Vogelstein, K. D. Micheva, R. J. Weinberg, S. J. Smith, and G. Sapiro, "Query based probabilistic synapse detection in immunofluorescence data." SfN 2016 Annual Meeting Poster Session, Nov 2016.

1 Data

1.1 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:

All resolutions

1. bock11
2. kasthuri11
3. kasthuri11cc
4. lee14
5. takemura13
6. Ex10R55
7. Ex12R75
8. Ex12R76
9. Ex13R51
10. Ex14R58
11. Ex2R18C1
12. Ex2R18C2
13. Ex3R43C1
14. Ex3R43C2
15. Ex3R43C3
16. Ex6R15C1
17. Ex6R15C2
18. acardona_abd1_5
19. acardona_0111_8
20. ZBrain
21. kristina15
22. bloss16
23. WannerAA201605
24. xbrain

Only resolution 0

1. Fear199
2. Cocaine175
3. Cocaine178
4. Cocaine174
5. Control258
6. Fear187
7. Fear200
8. Control181
9. Control182
10. Control239
11. Fear197
12. Control189
13. tobin16

2 Tools

2.1 neurodata.io

The new website (neurodata.io) is up and functional. It showcases our current tools and datasets that are available including links to code, documentation, and examples.

2.2 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 the package: https://github.com/neurodata/checklists/blob/master/matrix.md, the top five of which include summaries of data type, fraction of missing data values, generating heatmaps, and detecting outliers.

2.2.1 Synaptome Statistics

We have continued to examine the Kristina15 synaptome dataset and have now added the Weiler Chessboard dataset to our explorations. Using `meda`, and other methods along the way, we have started to compare the structure of these two datasets, see figure 1.

2.3 ndstore

We have now added a resource RESTful service which allows users to create, list and delete project management information. This service is very useful for power users to manage their projects without having to log into the management console.

We continue to improve our caching layer in the cloud. We have now added a cache manager which manages the Redis cache for ndstore. The manager monitors the cache and starts evicting data when the data in cache exceeds a pre-set limit. This allows for efficient management of Redis memory and data can be moved transparently. We also implemented a Readers-Writer Lock using Redis atomic services. This lock can be now used by different processes using the redis-client. We plan to work towards trying to add this lock into the open-source redis-py client.

We continue to enable authentication for ndstore and converting all RESTful calls to HTTPS from HTTP. We have two different ways to authenticate to ndstore for different use-cases. First, a single persistent token for ndio users who should authenticate for every RESTful call. This token can be downloaded from the management console as a secret token and saved locally by the ndio user. Second, ephemeral session tokens for third-party applications which can authenticate users via a login screen. Third-party applications forward these credentials to the authentication server for session based tokens. The use case for this are applications

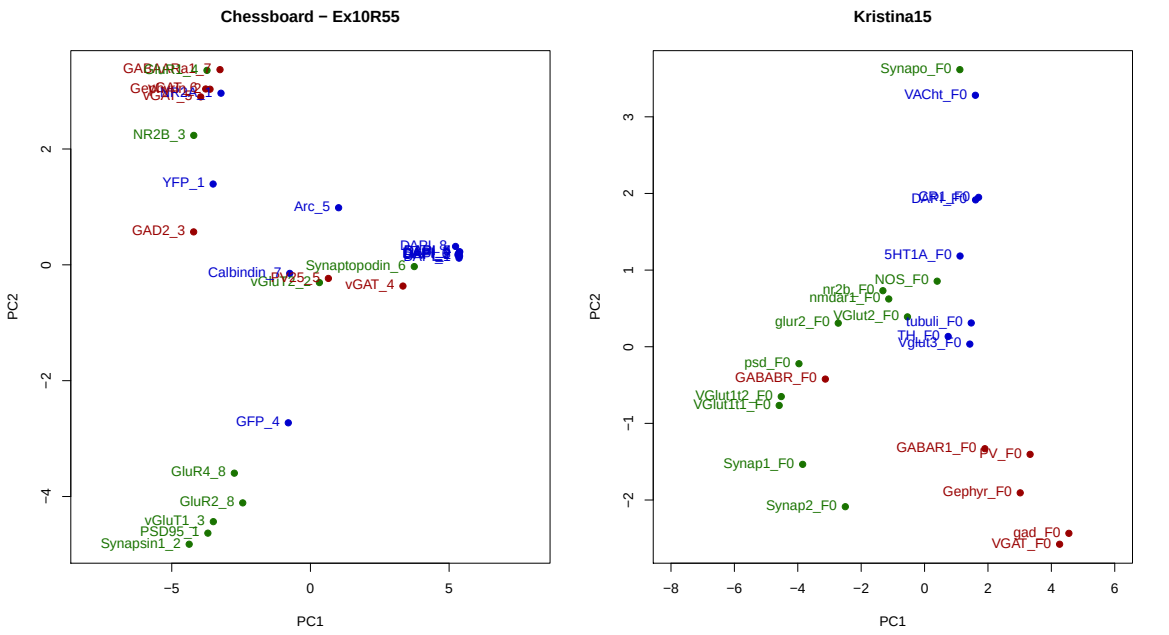


Figure 1: Principal Component Analysis (PCA) has been run on the correlation matrices of both of these dataset respectively. The first two principal components are plotted against each other. The colors correspond to the type of marker, ("excitatory" – green, "inhibitory" – red, "other" – blue). Notice that some markers are different between datasets, but there is a similar "claw" structure present.

are ndviz, ndtilecache, CATMAID, etc. This feature is under active development and is undergoing integration and testing. We also added continuous integration for automated testing with Travis–CI. This enables running our tests automatically for every commit.

All existing data projects currently ingested indo ndstore have been migrated to the cloud. In addition, services for MRI data ingesting have been integrated and the BNU1 dataset has begun to be ingested into ndstore (Figure below), and a slice can be accessed at the following link: http://mri.neurodata.io/nd/ca/BNU1/DTI_0025906_1/xy/0/0,182/0,218/90/1/.

2.3.1 ndingest

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

We have now built wrappers for AWS lambda, SQS, S3 and DynamoDB in ndingest. The next stage is to write AWS security policies and ARN roles so that different aspects of these services can communicate with each other in the cloud. We are also developing scripts to

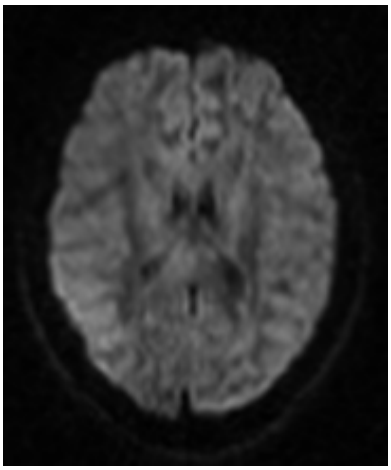


Figure 2: A new tile interface was added to support filtered metadata tiles. Previously, when the user wanted to visualize an annotation project, all annotations were rendered as different colors based on ID. The filtered metadata tiles service allows the user to specify which annotation IDs to include when the tile is rendered. Using the service, combined with the RAMON Metadata services described in ndramondb below, a user can dynamically filter for specific IDs (e.g. all synapses), resulting in a cleaner and simpler view.

deploy these services or update configurations from the command line. In addition, we have also developed a migration script to migrate our existing data to the cloud. This is essential since ndingest is designed to ingest raw data not data already in ndstore.

We have also developed an ingest-client with JHU-APL to allow users to be able to use this service. The ingest-client is designed to be modular allowing users to add their respective plugins which match their data naming conventions. This is an improvement over the earlier phase where users had to convert their data to match a standard naming convention. This tool is also complemented with an ingest service which allows users to create, join, track and delete ingest jobs.

2.4 ndreg

The Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm is an image registration method used to compute a smooth invertible transform ϕ_{10} which aligns template image I_0 to target image I_1. The log Jacobian determinant of the mapping $\log |D\phi_{10}|$ measures local volume change during LDDMM. Wherever it's negative the template CLARITY brain expanded to match the targeted Allen Reference Atlas (ARA). Wherever it's positive the CLARITY brain contracted. Figure 3 shows the $\log |D\phi_{10}|$ overlaid on a CLARITY brain and a histogram of its values. It's clear that this brain expanded in most places to match the ARA. This was also the case for 8 of 9 of the of the other brains. Our calculations showed that CLARITY brains were 21% smaller than the ARA on average. This likely occurred due to shrinkage introduced by the CLARIfying processing.

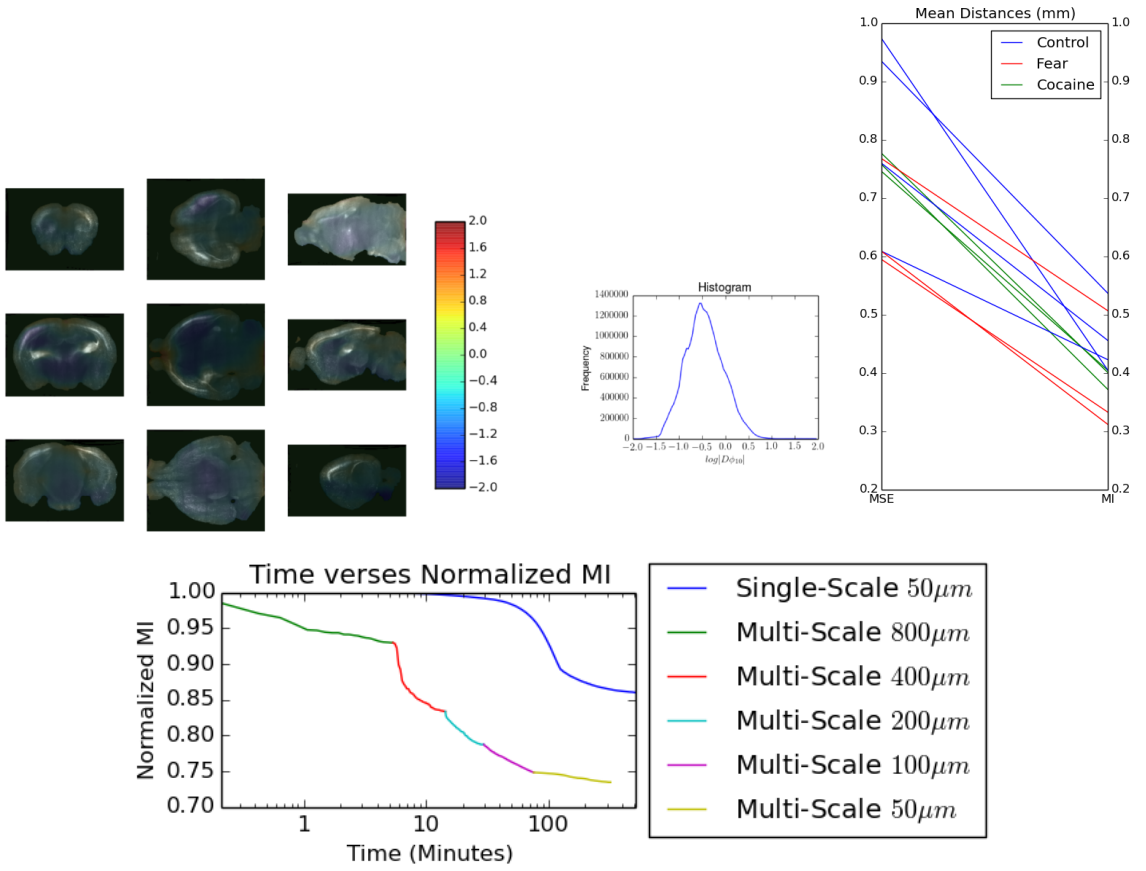


Figure 3: The metric distance between the CLARITY and ARA was computed at the end point of the registration by integrating the time varying velocity fields from LDDMM. It is a measure of how much the CLARITY brain differed from the ARA. The next figure compares the calculated distance when a Mean Square Error (MSE) cost is used during LDDMM to registration with Mutual Information (MI). The metric distance was consistently smaller for MI under all 3 conditions (Control, Cocaine and Fear). This indicates that the quality of the registration was higher than under a MSE cost.

NeuroData's registration module (ndreg) uses the Large Deformation Diffeomorphic Metric Mapping (LDDMM) for image alignment. LDDMM computes a smooth invertible mapping between template image I_0 and target image I_1. The plot below shows the timing results of experiments registering a CLARITY brain template to the Allen Reference Atlas (ARA) target. In the first experiment the CLARITY image was registered to the (ARA) using a single-scale approach on a 50 μ m grid. In the second experiment registration was done by a coarse to fine multi-scale method. Registration that was done at a lower resolution was used to initialize the algorithm at the subsequent higher resolution level. Resolution levels of 800, 400, 200, 100 and then 50 μ m were used. The Mutual Information (MI) between the deformed CLARITY and ARA images was normalized to a range of $[0, 1.0]$ where 1.0 indicates that no registration

occurred and 0 indicates the best possible scenario ($I_0 = I_1$). The plot in figure 3 shows that the multi-scale optimization was much faster than the single-scale method.

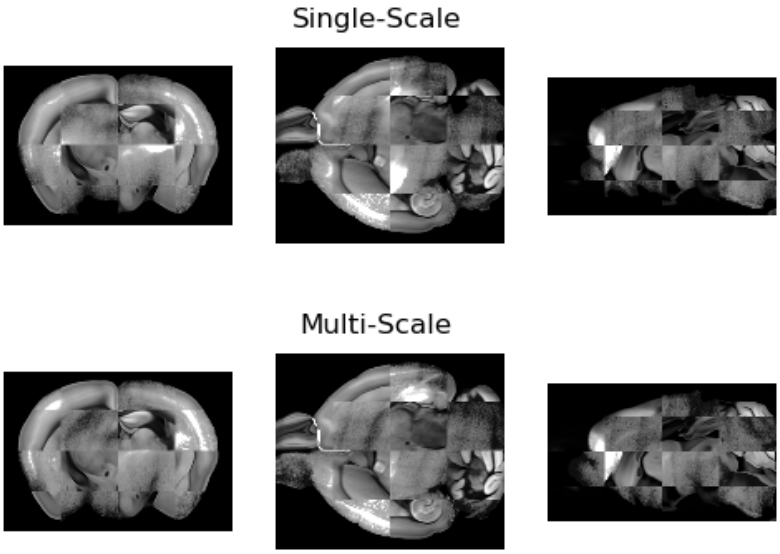


Figure 4: The images below are checkerboard composite images of the deformed CLARITY image and the ARA. They show that usage of the multi-scale approach did not reduce registration quality when compared to the single-scale method.

2.5 ndramondb

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

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

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

New Webservices that return JSON formatted object describing metadata were implemented for all RAMON metadata queries. The will be preferred to the existing interfaces that

return HDF5 objects for Web applications. JSON is a simple text serialization format that is widely supported. It removes the HDF5 software dependency to access neuroscience metadata. HDF5, while powerful, is a cumbersome dependency in many operating systems and frameworks.

The JSON Web services were specifically designed for the ndviz visulation tool, which is built on Javascript. Ndviz now supports clickable metadata for any annotated location, i.e. a marked region in an annotation project. Clicking an annotation brings up the type of annotation (neuron, axon, dendrite, synapse) properties of the annotation (weight, confidence), and other metadata (author, creation date).

2.6 ndviz

Several small ndviz interface adjustments were introduced. First and foremost, a scale bar was added to the viewer screen. Additional visual tweaks and bug fixes were made, and a new beta release was published, http://viz.neurodata.io.

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

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

Finally, both the bounding box and the query by key ndramondb queries were added to ndviz. Ndviz can now center on an arbitrary annotation, as well as list all annotations by type for a given project (see Figure 5).

2.7 Graph Explorer

The graph explorer now has new ability to view statistics conditional on the attributes in the graph, see figure 7. The conditional distributions illustrate the similarity and distinction among the different vertices in the network data. Difference among these distributions are potentially linked to their different functions in physiology.

2.8 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

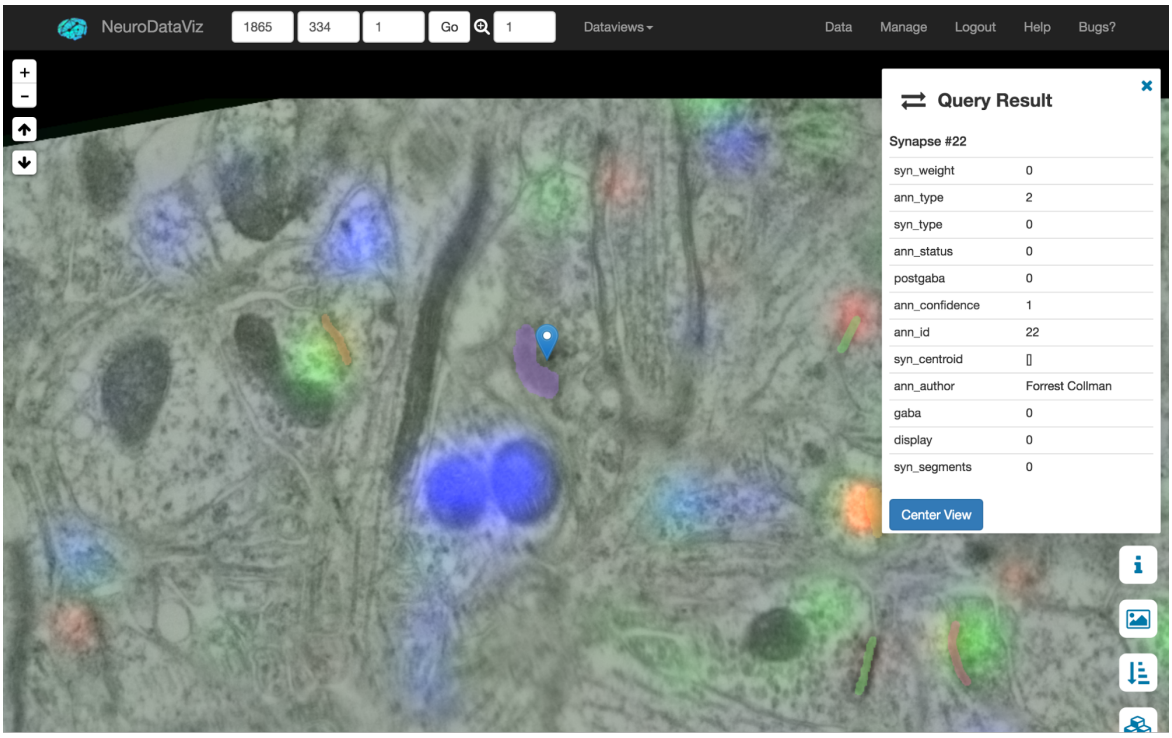


Figure 5: ndviz displaying an annotation project (Array Tomography with conjugate Electron Microscopy) with Synapse #22 centered (blue marker). The figure was obtained by loading annotations, selecting Synapse #22 in the Query Tool box (magnifying glass, picture below), and clicking the blue "Center View" button.

SCSR. Figure 8, 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.

The performance ratio of the in-memory and semi-external memory sparse matrix multiplication in FlashX is affected by multiple factors. Shown in Figure 9, 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

Figure 6: The ndviz Query Tool controls box. Clicking a blue Synapse ID will open the Query Result box in the viewer, and allow the user to center the current view on the selected annotation.

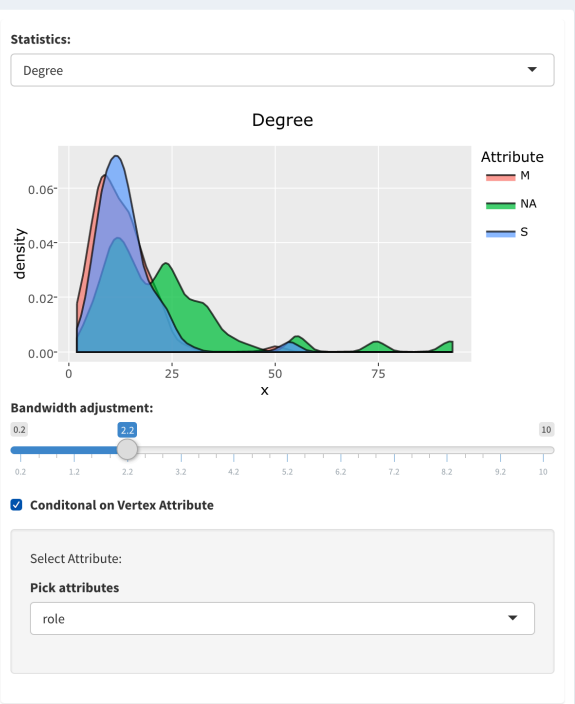


Figure 7: Graph Explorer interactivity example.

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 $n \times 16$

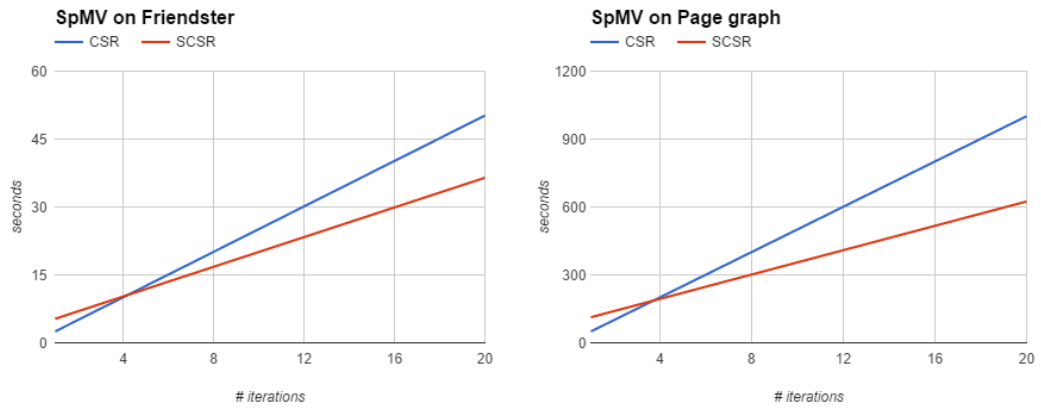


Figure 8: Benefits of using SCSR.

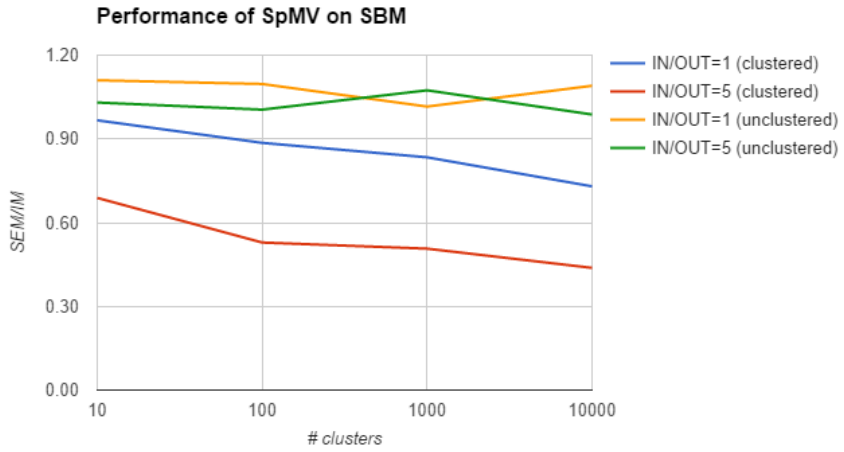


Figure 9: Benefits of using SCSR.

non-negative dense matrices.

We significantly improve the performance of SEM-NMF by keeping more columns of the input dense matrix in memory (Figure 10). The performance improvement is more significant when 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 10). 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.

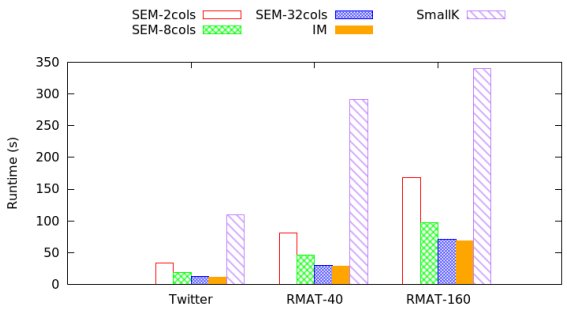


Figure 10: Performance improvements of SEM-NMF.

2.9 FlashY

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

When multiple graphs are involved, the dimensionality problem becomes more challenging – since each graph is a matrix, stacking multiple graphs leads to a 3-dimensional array referred as tensor. There has been little research in handling this type of data in the network data domain. A dimension reduction toolbox, to be used in R, has been created to convert each adjacency matrix into a small vector, while retaining most of the variability across subjects. We tested, as shown in Figure 11, the classification performance using sex as outcome labels and brain network as input, the low dimensional vector produced indistinguishably good performance as the large network data.

We consider improving the stability in clustering objects. When the sample size is low but there is large variability due to the high dimension, the inferences based on the existing clustering methods (k-means, Gaussian mixture model, etc.) tend to produce large error. This problem is critical and common in graph clustering, as the number of graphs is typically low but the number of vertices is high. To solve this, a new clustering method, namely "jk-means" is developed in R. With a truncation, substantial error reduction can be obtained via the bias-variance tradeoff. Shown in the figure, the new jk-means clustering tool (red) produces significantly better result than the other models.

2.10 ndmg

We have continued the development of a reliable DWI and fMRI processing pipeline package, with current emphasis on performing intermediate derivative quality control. The quality of claims made from derived data products is heavily dependent upon the quality of input data to each processing step. To that end, it is necessary to perform quality control at each processing step of the pipelines including: registration, tensor estimation, and fiber estimation. We generate quality control images for a qualitative analysis enabling the user to evaluate

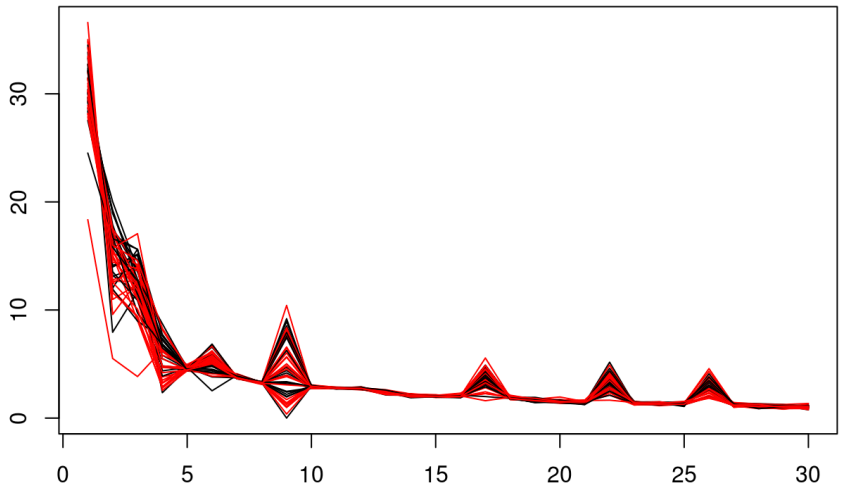


Figure 11: View of the low-dimensional vectors from 46 subjects, dimension is reduced from 4,900 to only 30 for each subject.

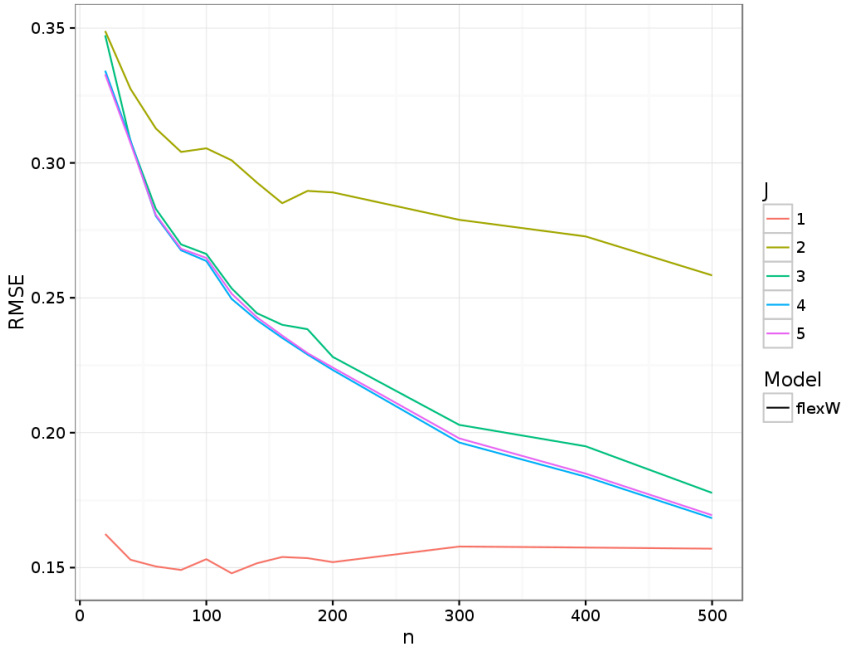


Figure 12: Illustration of the jk-means, compared with Gaussian Mixture Model (j=5). The j=1 jk-means model (red) has significantly lower error compared to the others, when the sample size is smaller than 500.

the quality of derivatives produced (see, Figure 13). Quality control is essential to ensure accuracy of and to substantiate claims derived from ndmg results. In particular, investigating quality control figures has led to impactful improvements in the diffusion pipeline, which now is nearly perfectly discriminable for a variety of datasets tested. Moreover, we are developing a web service for fMRI uploading to streamline the process of analysis. Our web service will facilitate the dissemination of results and ensure that researchers have easy access to to a substantial array of pre-processed data as well as metrics computed upon them.

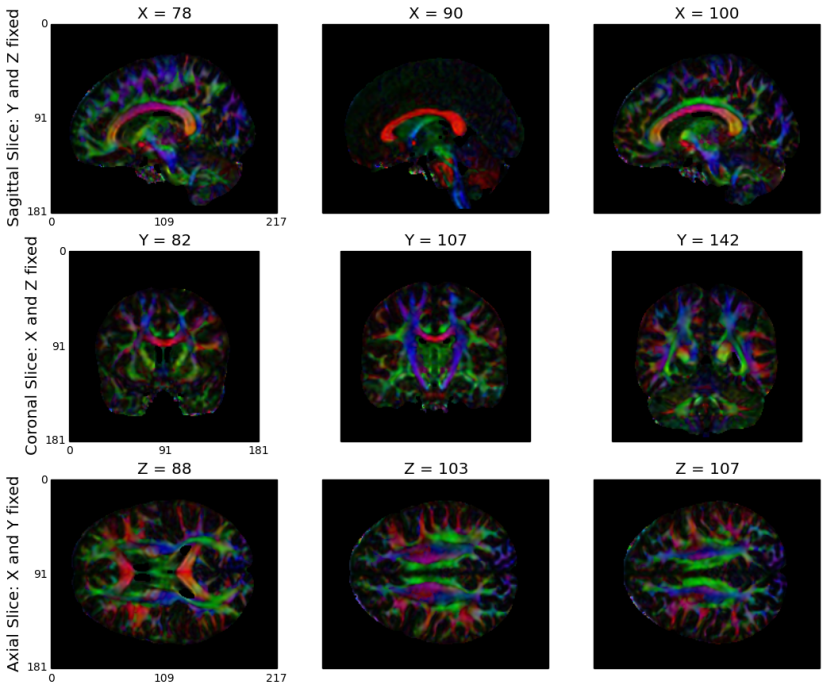


Figure 13: NDMG various output views.

2.11 Science in the Cloud (SIC)

A framework for science in the cloud (SIC) was developed which was designed for extensibility of scientific findings. This work was developed alongside a live demonstration, available at http://scienceinthe.cloud, which illustrates this framework. There are six key components which much be considered in sic: data storage, data organization, interactive demos, virtualization, computing, and deployment. The selection made for each of these components will have a significant impact on available selections for the others. The final product will be a highly interdependent network of tools and data.

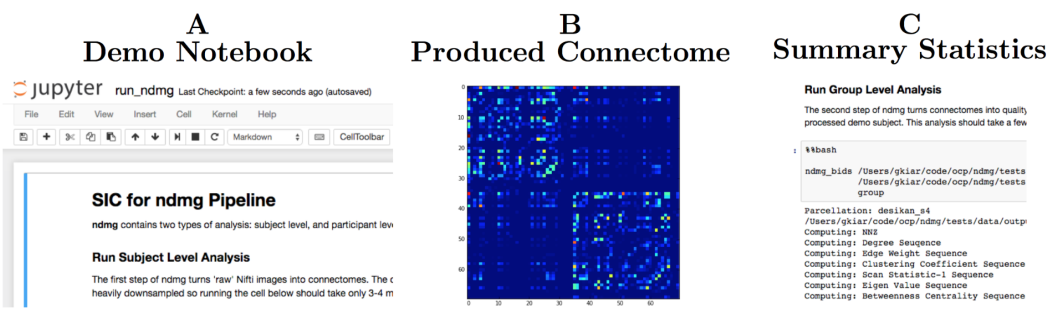


Figure 14: States of the demo notebook in the cloud. A) A Jupyter notebook displaying descriptions and code snippets to be run for both connectome estimation and summary statistic computation. B) After running connectome generation an adjacency matrix will appear to provide a visualization. C) Summary statistic computation calculates several graph features and plots them in a multipanel figure.

We have refined our framework for science in the cloud (sic) and submitted the manuscript to GigaScience. In particular, we have clearly defined six crucial components to adopt this model of extensible scientific tool production. These six components are: cloud data storage, data organization, interactive demonstrations, virtualization, deployment, and computing. Table 15 summarizes some tools for each of these components, as well as the selection made in our particular implementation of sic.

2.12 Non-Parametric Shape Clustering

We are developing non-parametric methods to cluster shapes. A shape consists in the geometric information contained in an object, which is scaling, translational, and rotational invariant. Our method consists in applying a modified version of k-means algorithm which incorporates a mathematical metric between shapes based on Procrustes alignment. This can have several applications, and we are particularly interested in using this technique to analyze similarities between neural synapses. As an example, the figure 16 illustrates the shape extraction and alignment between two handwritten digits. Using the distance provided by this method, we are able to cluster three classes of digits more accurately than pure k-means method.

To test our algorithm we used the well-known MNIST handwritten digits dataset. For each digit we extract landmark points, and align them through Procrustes. This procedure defines a distance between any two objects, and is illustrated in figure 18 (a). Using this distance we then cluster three classes of digits using a modified K-means algorithm.

The example shown in figure 18 (b) shows a pretty low error, around 10%, for an unsupervised clustering method.

Table 1: There are six key components which must be selected for SIC. **Bold** indicates the selections made here, with their positive and negative qualities compared to some alternatives.

Hurdles	Available Tools	Pros of Selection	Cons of Selection
1) Data Storage	**S3** [8], Dropbox [9], Google Drive [10]	API, pay-by-usage	requires user familiarity with Amazon tools
2) Data Organization	**BIDS** [11], NWB [12], MINC [13]	documented, validator, active community	new, not yet fully adopted
3) Interactive demo's	**Jupyter** [14], R Notebook [15], Shiny [16]	versatile, accessible	optimized for Python
4) Virtualization	**Docker** [17], Virtualbox [18], VMware [19]	lightweight, self-documented	--
5) Deployment	**manual**, ECS [20], Kubernetes [21], MyBinder [22], CBRAIN [23]	no additional dependencies	does not scale effectively
6) Computing	**EC2** [20], Google Compute Engine [24], Microsoft Azure [25]	scalable, flexible	requires technological expertise

Figure 15: Excerpted table from [7].

2.13 Reduced Dimension Clustering

We develop a new statistical method, named "automatic repulsive clustering" (ARC, to handle the clustering problem when the dimensionality outgrows the number of the data points. The standard dimension reduction tool such as principle component analysis (PCA) is not optimal for finding the subspace best suitable for clustering. Therefore, we seek the method to do dimension reduction and clustering at the same time: we develop a regularization to encourage good separation of the clusters, forcing the subspace to an angle so that each point can belong to a cluster completely. The plots below shows the different performances in clustering a equally proportioned 3 components: the subspace found by PCA leads to incorrect clustering of the data, due to the orientation of the space; whereas ARC successfully identifies the ideal clustering direction in one dimensional space, hence generating satisfactory clustering result.

2.14 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.

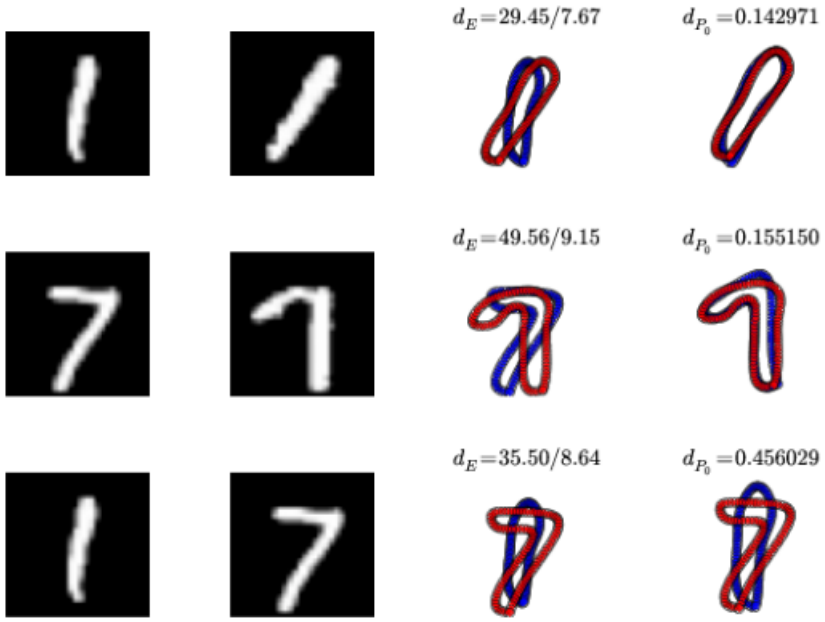


Figure 16: Example: shape extraction and alignment between two handwritten digits.

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.

Most recently, we performed simulations in which we artificially rotated, scaled, rotated + scaled (labeled "affine" in figure **??**), and corrupted two simulated datasets. The results demonstrate that both RerF and another existing oblique method called RR-RF are more sensitive to scaling and corruption of the data than is RF. The variants of these two methods that use rank transformations to rescale the data first, called RerF(r) and RR-RF(r), become significantly more robust to scale and corruption. While rank transformation is not the only scale normalizing method (z-scoring and forcing all values to be between $[0, 1]$ are others), it is the only one that simultaneously helps against data corruption. Across all simulation settings, RerF(r) is the overall best.

We are developing an R implementation of the RerF algorithm in order to make it faster and more easily distributable. The Rerf algorithm requires a random rotation of data at each node during the tree creating process, but this rotation of data at each node increases memory requirements and both training and testing times. So, the initial R implementation will instead rotate the sample data once prior to creating a tree instead of at each node in the tree. This

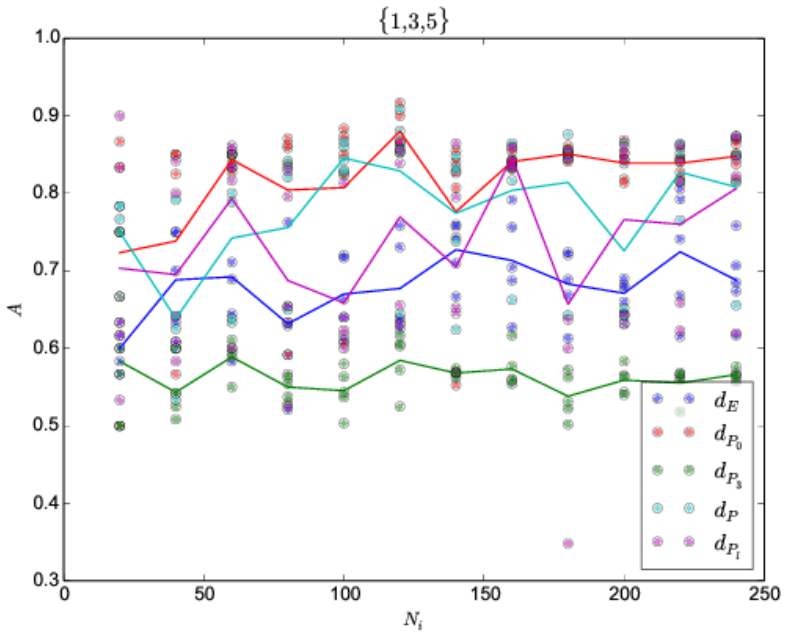


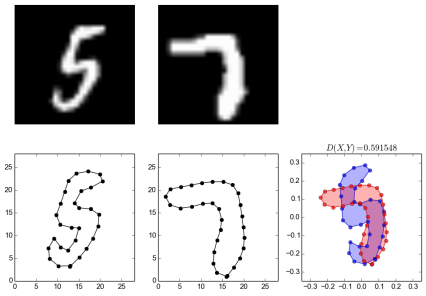
Figure 17: The graph shows the accuracy (1 is the best possible) versus the number of elements in each class. Our shape clustering methods (red and cyan lines) are more accurate than k-means based on Euclidean distance (blue, magenta, and green lines). In this example the accuracy is around 85%, which is quite good for unsupervised clustering on this dataset.

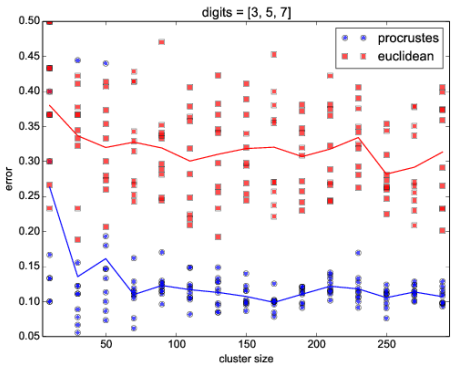
will still increase the same factors but not to the same degree. Initial tests show that this single rotation per tree yields benefits similar to the per node rotation.

The classification performance of RerF2 using the R implementation is similar to the performance of the original RerF2 implemented in MATLAB, but the observed differences are greater than can be explained by the use of different randomization algorithms. Both implementations are being analyzed to determine the source of this observed variability.

2.15 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$^{\text{th}}$ 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. Previously, we process 13 fMRI data sets with 64 pipelines, and show some pipelines yields data sets with high sample discriminability. Currently, we are developing a statistical test to determine whether one pipeline is significant better than the other. To develop a valid test, we need to approximate the distribution of difference between sample discriminability estimates under the null hypothesis that two population discriminability are equal. We design





(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 18: MNIST data and classification error results.

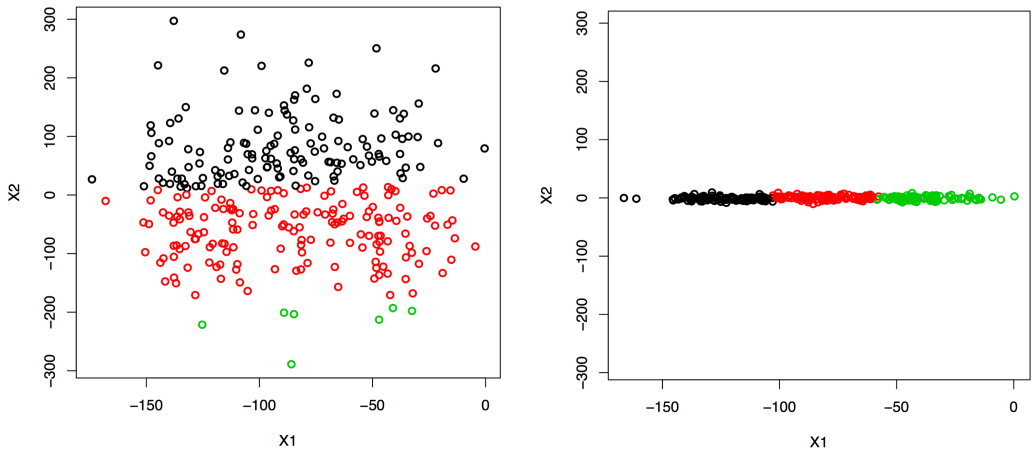


Figure 19: PCA (left) versus Reduced Dimension Clustering (right).

and implement a bootstrap procedure to achieve this which is described below.

Testing the null hypothesis $D(\Psi_1) = D(\Psi_2)$ Input: Data set, Pipeline Ψ_1, Pipeline Ψ_2 Output: Reject or do not reject the null hypothesis

1. Process the data set with pipelines Ψ_1 and Ψ_2
2. Compute $\hat{D}(\Psi_1)$ and $\hat{D}(d)$

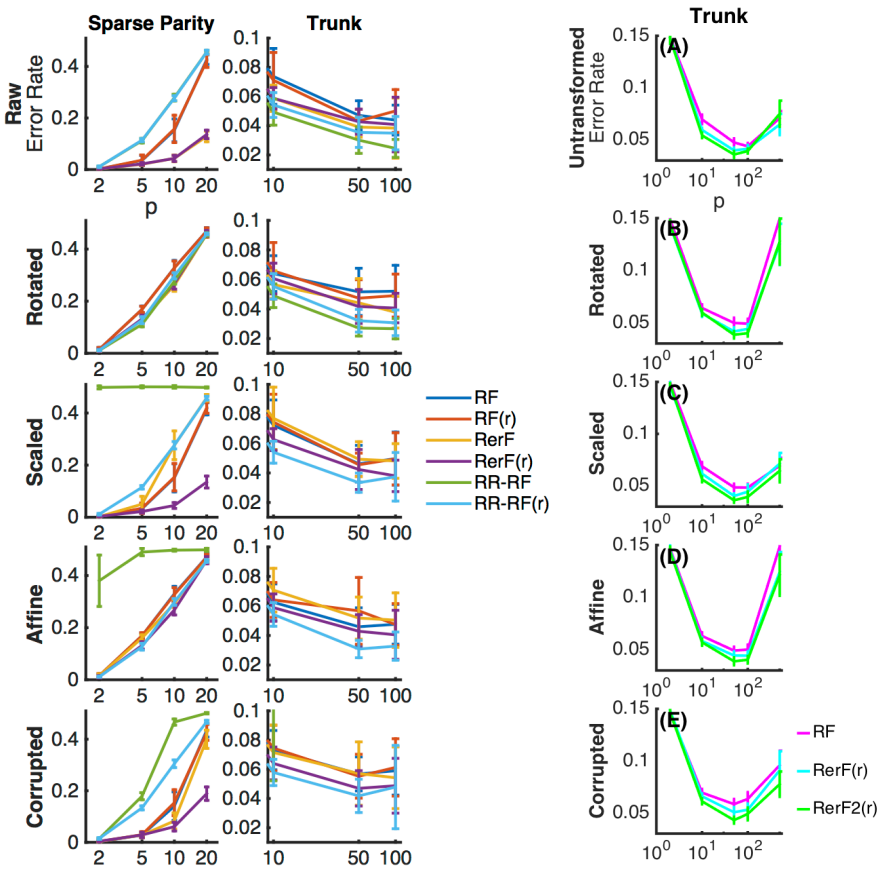


Figure 20: Randomer Forest Comparisons

3. For {i in 1 through number of repeats}

   (a) For {j in 1 through number of subjects}

      i. Randomly select two subjects from data set

      ii. Linearly combine measurements of these subjects

4. EndFor

5. Compute pairwise differences $D^{(i)}(\Psi_1) - D^{(i')}(\Psi_1)$ and $D^{(i)}(\Psi_2) - D^{(i')}(\Psi_2)$

6. Compute p-value which is the fraction of times that $\hat{D}(\Psi_1)$–$\hat{D}(\Psi_2) > D^{(i)}(\Psi_j)$–$D^{(i')}(\Psi_j)$

7. Reject the null hypothesis if p-value is less than 0.05.

We are currently running the test procedure on 13 data sets processed 64 ways. After finish running, we will have 13 p-values for each pipeline. We will use Fisher's method to combine results.

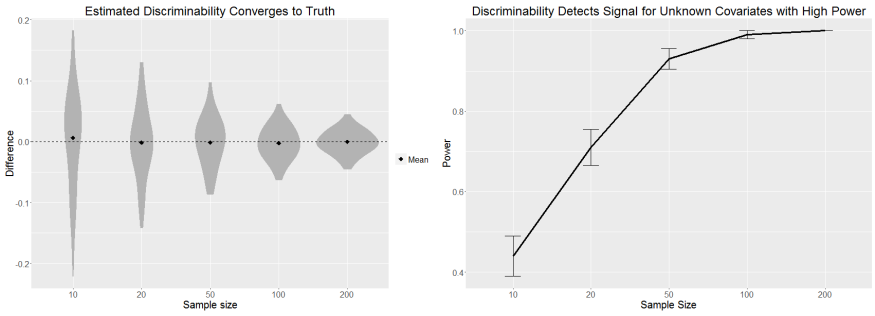


Figure 21: **Convergence of** $\hat{D}$ Distribution of difference between discriminability estimates and truth is shown in the left panel. The physical property and noise are generated from standard Gaussian distribution. The black dots indicate the mean over 100 repeats. As the number of subjects increases, the sample discriminability converges to the true population discriminability. **Discriminability Test Power.** Test power of discriminability with varying sample size is shown in the right panel. The physical property and noise are generated from standard Gaussian distribution as described in the simulation section. At level of 0.05, the power is estimated based on 100 repeats. The test power becomes close to 1 with more than 50 samples.

Furthermore, we take the previously computed fMRI discriminability data and compare all pipelines to the CFXXG pipeline (CC200 atlas, FSL, No Scrubbing, No Filtering, Signal Regression) use Wilcoxon signed rank test. CFXSG pipeline has the best mean discriminability across data sets. The result is shown in figure 22.

2.16 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.

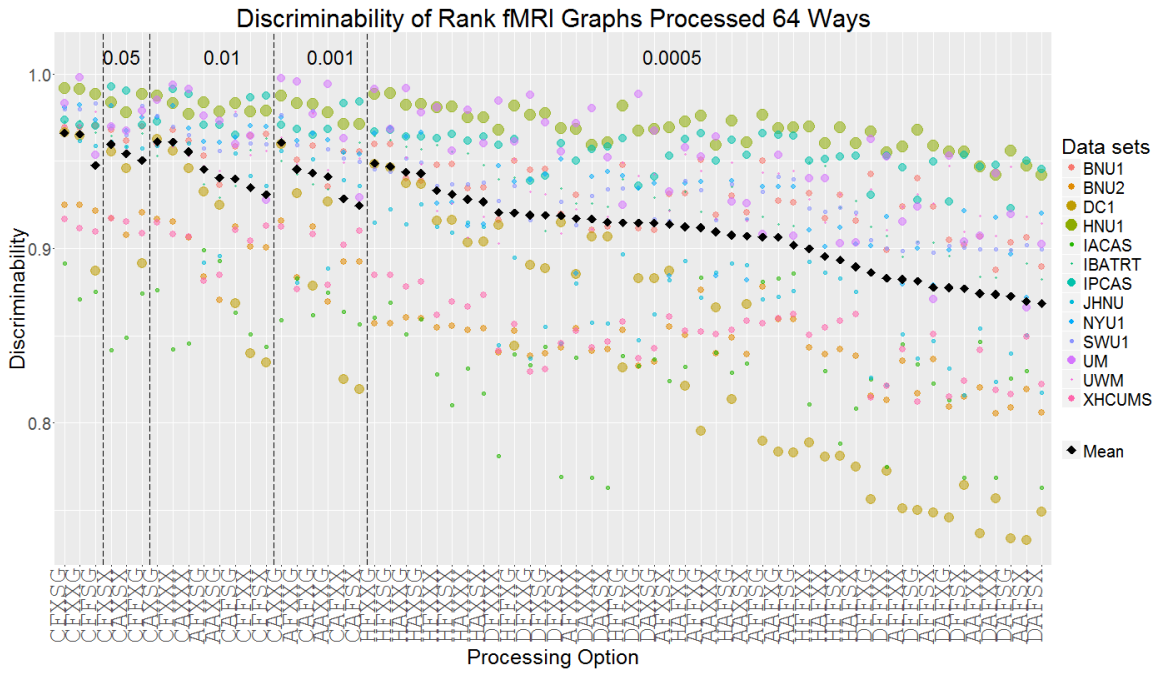


Figure 22: **Discriminability of rank fmri graphs from 13 data sets processed 64 ways.** Discriminability of BNU1, BNU2, DC1,HNU1, IACAS, IBATRT, IPCAS, JHNU, NYU1, SWU1, UM, UWM and XHCUMS pre-processed by 64 pipelines are computed and shown in the top panel. Color of each dot indicates data set and size indicates the number of measurements in data set. The black square indicates the weighted mean discriminability across 13 data sets. The pipelines are first grouped by p-values when comparing to pipeline CFXXG using Wilcoxon signed-rank test. The number at the top indicates the p-value when compared to CFXXG. Within each group, the pipelines are ordered by the mean discriminability. CFXSG pipeline has the best mean discriminability across data sets.

2.17 Robust Law of Large Graphs

To estimate the mean of a collection of weighted graphs under a low rank random graph model (e.g. Stochastic Blockmodel) when observing contaminated graphs, we propose an estimator which not only inherits robustness from element-wise robust estimators but also has small variance due to application of a rank-reduction procedure. Under appropriate conditions, we prove that our estimator outperforms standard estimators via asymptotic relative efficiency. And we illustrate our theory and methods by Monte Carlo simulation in figure 23.

Also, to get the convergence rate of the variance of MLqE, we proved: 1. the uniform convergence $\sup_{\theta\in\Theta}\left|E_f[g(X,\theta)]-\frac{1}{m}\sum_{i=1}^{m}g(X_i,\theta)\right|\to 0$ as $m\to\infty$, where $g(x,\theta)$ is the equation for MLqE; 2. There exists at least one population solution of the MLqE equation which is smaller than the expectation of the MLE.

In summary, MLE outperforms a little bit when there is no contamination, but it degrades dramatically when contamination increases while MLqE still does a good job. Applying ASE to

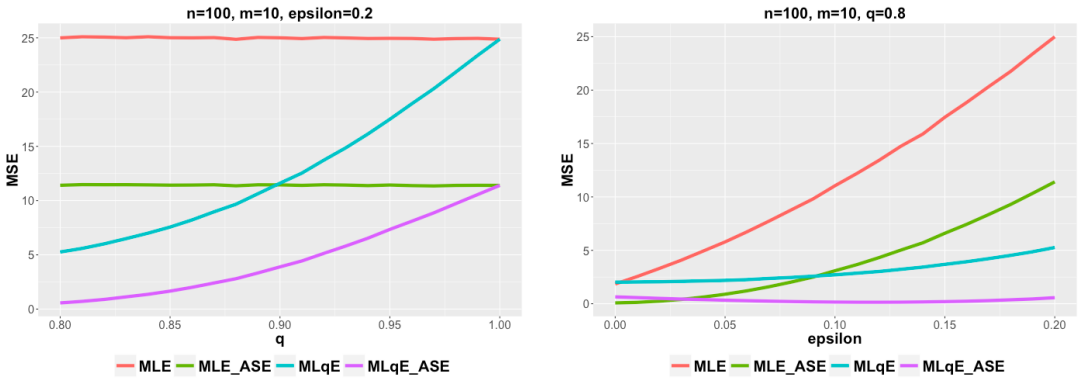


Figure 23: For the figure on the left, we vary the proportion of the contamination represented by and conclude: **1. MLE (red) vs MLqE (blue)**: MLE outperforms a little bit when there is no contamination, but it degrades dramatically when contamination increases; **2. MLE (red) vs MLE_ASE (green)**: MLE_ASE wins the bias-variance tradeoff; **3. MLqE (blue) vs MLE_ASE (purple)**: MLE_ASE wins the bias-variance tradeoff; **4. MLqE_ASE (purple) vs MLE_ASE (green)**: When contamination is large enough, MLaE_ASE is better, since it inherits the robustness from MLqE. For the right panel, we vary the parameter q in the MLqE. Note that when $q = 1$, MLqE becomes MLE so that the curves intersect. As q decreases, we intend to have more bias but still win the bias-variance trade-off by reducing the variance.

either entry-wise MLE or entry-wise MLqE reduces the variance by bias towards the low rank approximation. Our estimator based on ASE of MLqE inherits the robustness from MLqE, and it has low variance. When contamination is large enough, our estimator performs the best among all the four.

We tested our algorithm's performance on a synthetic dataset based on structural connectomic data. The graphs are based on diffusion tensor MR images collected and available at the Consortium for Reliability and Reproducibility (CoRR). It contains 454 different brain scans, each of which was processed to yield an undirected, weighted graph with no self-loops, using the ndmg pipeline.. The vertices of the graphs represent different regions in the brain defined according to an atlas. We used the CPAC200 atlas with 200 vertices. The weight of an edge between two vertices represents the number of white-matter tract connecting the corresponding two regions of the brain. In order to evaluate the performance of the two estimators, we used a cross validation on the 454 graphs of each size. Specifically, for a given atlas, each Monte Carlo replicate corresponds to sampling m graphs out of the 454 and computing the four estimates using the m selected graphs. We then compared these estimates to the sample mean for the remaining 454-m adjacency matrices. We ran 1000 simulations on the CPAC200 atlases for both sample size $M = 2, 5$. We also considered all possible dimensions for adjacency spectral embedding by ranging d from 1 to n in order to investigate the impact of the dimension selection procedures. We plot the result in the following figure.

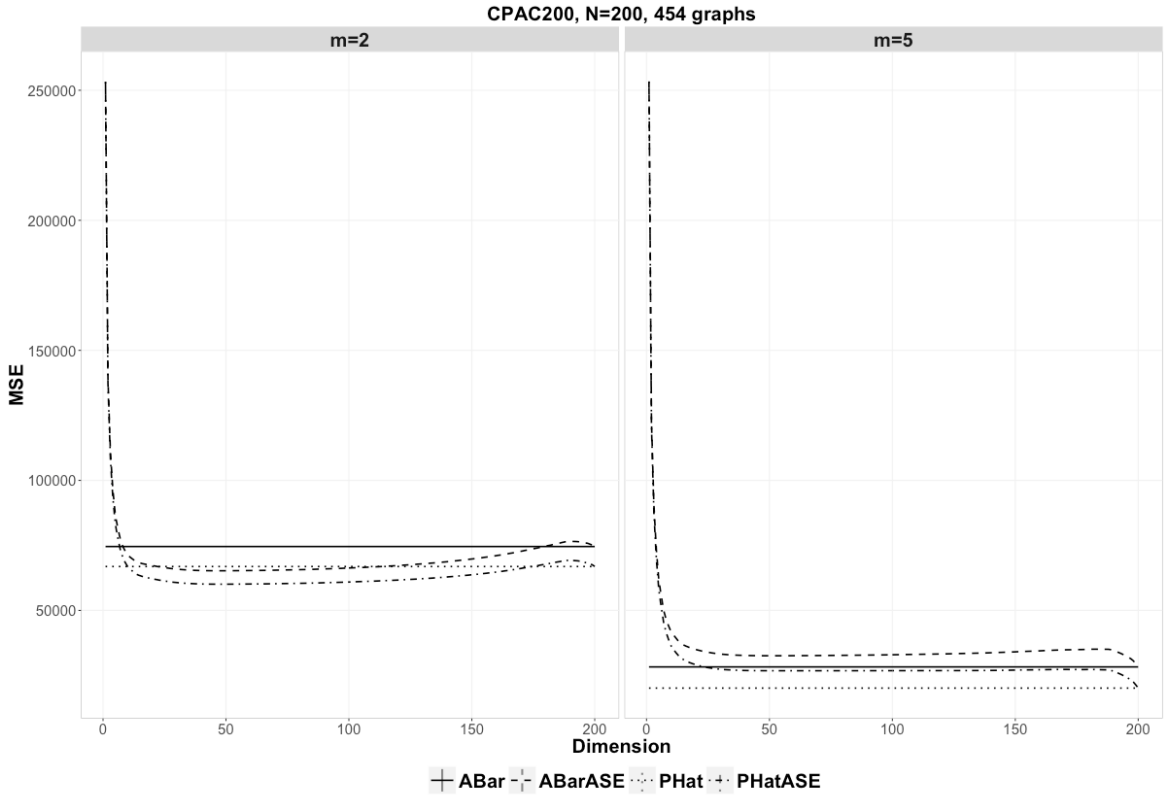


Figure 24: **Comparison of MSEhat of the four estimators for the CPAC200 atlases at two sample sizes for the CoRR data.** For the figure on the left, we examine the four estimators at sample size m=2. **1. MLE (horizontal solid line) vs MLqE (horizontal dotted line):** MLqE outperforms MLE since robust 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. For the figure on the right, we still can see the robustness but the lack of low-rankness for a larger number of samples degrades the effect of ASE.

2.18 LOL

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

LOL are quite pathological.

2.19 Multiscale Network Test

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

For simulation, Stochastic Block Model (SBM) and additive and multiplicative network model have been explored which also exhibit non-linear dependence properties. What MGC does in this case is to test distance matrix of diffusion maps and nodal attributes, considering (k, l) nearest neighbors in each.

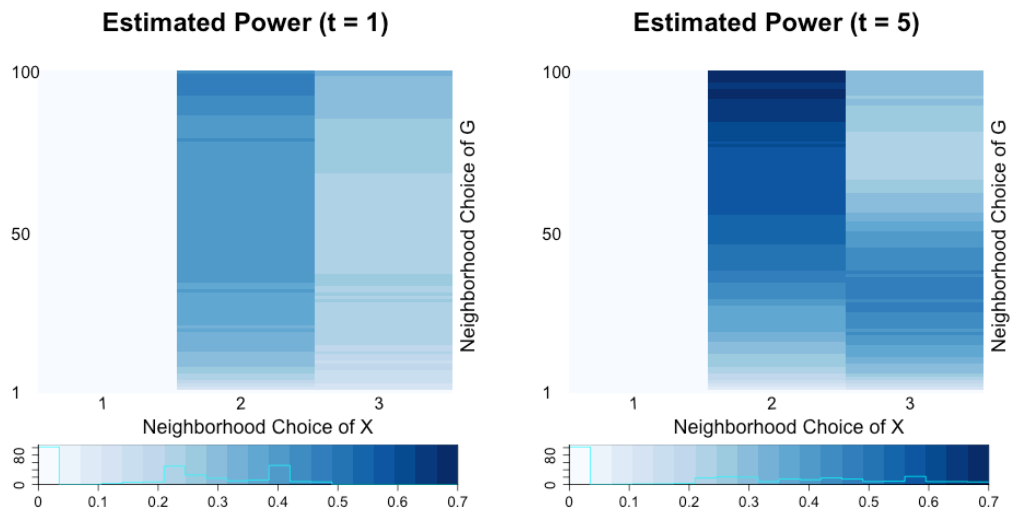


Figure 25: The above figures illustrate power maps of three-block stochastic block model in terms of - nearest neighbor choice in terms of network diffusion maps and k-nearest neighbor choice in terms of nodal attributes. You can also notice that diffusion time matters in testing.

Thus if there exists local dependency structures or nonlinearity, the optimal neighborhood choice of (k, l) would not count every node in network in compute distance correlation matrix. We also demonstrated that testing power of MGC applied to diffusion maps is higher in SBM and also degree-corrected SBM, compared to dCov, Heller-Heller-Gorfine, and latent factor network test proposed by Fosdick and Hoff (2015).

2.20 Multiscale Generalized Correlation (MGC)

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

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

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

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

2.20.1 Testing Node Contribution via (MGC)

We continue the development of the Multiscale Generalized Correlation toolbox, to handle the node contribution task within the MGC testing framework.

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

An important question useful for subsampling and outlier detection, is how important is each sample observation, regarding their relative contribution to the underlying dependency. If this question can be successfully and efficiently answered, those important samples can be kept for later inference, while less important samples may be treated as outliers.

By decomposing the optimal local correlation $C*$ into each sample point, the MGC computation readily offers a weight statistic w_i for each pair of observations (X_i, Y_i) where $\sum_i w_i = C*$. Then each sample can be ranked based on W_i. This is a very efficient algorithm, since computing the MGC statistic for all data automatically yields w_i for all sample pairs.

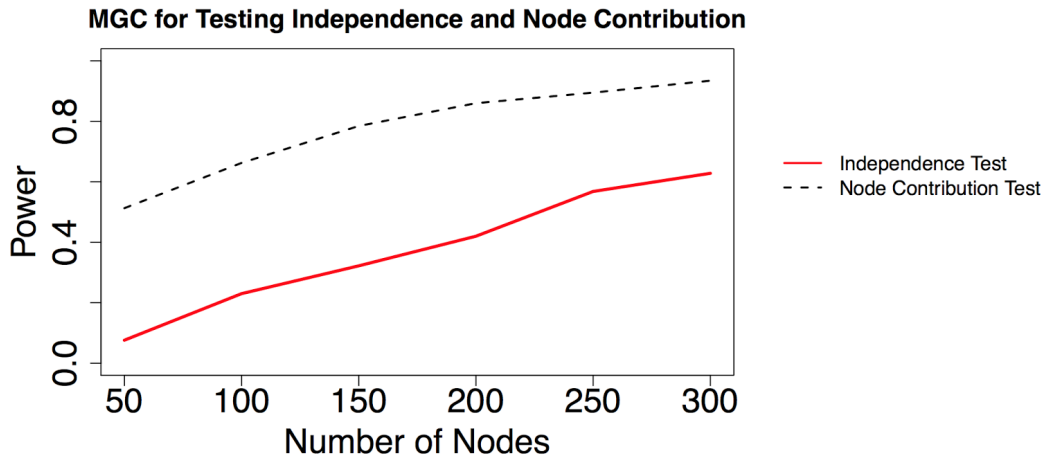


Figure 26: This figure considers a mixture model in the graph domain, where half the nodes are dependent to their attributes, and the other half nodes are independently generated from the attributes. We calculate the MGC statistic and its power for dependence testing, as well as the node contribution statistic and its power, i.e., the percentage of all dependent nodes that are ranked among the first half by w_i. Indeed, the power plot not only shows that our node contribution algorithm can successfully identifies all important nodes at mildly large sample size, but also hints a tight relationship between w_i and $C*$ that is worthy of further investigation.