The Impact on Inter-Subject Discriminability of Dimensionality Reduction in fMRI

Eric Bridgeford
Johns Hopkins University
3300 N Charles Street
Baltimore, MD 21218
ericwb95@gmail.com

Tanay Agarwal
Johns Hopkins University
3300 N Charles Street
Baltimore, MD 21218
tagarwa2@jhu.edu

Abstract

The removal of nuisance variables is critical for making effective inferences from fMRI timeseries. Nuisance variables such as physiological noise, motion, scanner heating, and many other factors contribute to changes in the blood-oxygen level dependent (BOLD) signal measured at each timestep during an fMRI scanning session. In this work, we design and implement several techniques for the removal of nuisance related signal error, and examine the implications on the quality of inferences that can be made from the resulting fMRI timeseries.

1 Background

For the past thirty years, fMRI has presented one of the most intriguing challenges for computational neuroscience. While researchers have long understood that an increase in blood flow was necessary for increased brain function (such as during periods of high brain activity), it was not until a seminal study by (1) that researchers were able to image the flow of blood in the brain. Oxygen-rich blood (high in Hb, the oxygenated form of hemoglobin) is slightly more magnetic than deoxygenated blood (high in dHb, or deoxygenated hemoglobin). As Hb has different magnetic properties than dHb, a high strength magnetic field such as that of an MRI scanner is able to pick up time-dependent fluctuations in the concentration of Hb. While regions of the brain showing high activity see a brief drop in the concentration of oxygen in vessels supplying the region, the arteries rapidly overcompensate by the hemodynamic response (2), which sees a significant increase in the concentration of oxygenated blood pushed towards the regions with high brain activity. This fluctuation in the Blood Oxygenation Level Dependent (BOLD) signal is what is imaged in an fMRI scanner.

Over the course of an fMRI scanning session, many sources of noise arise that must be corrected for in order to make quality data inferences. The scanner inadvertantly heats up (producing a high strength magnetic field for sessions lasting up to ten minutes produces an enormous amount of heat). As the scanner heats, the fine electronic equipment inadvertantly drifts in the signal it picks up (first demonstrated by (3) when they showed that a heated scanner detected "brain activity" in cadavers). This drift has been shown to be approximately quadratic, as fitting higher order polynomials does not seem to improve the timeseries quality (see section 3 for details about our meeting with a domain expert).

More interestingly, the magnetic field passed over brain tissue is altered by minor fluctuations in physiological related stimulus. Factors such as heart rate, blood pressure, and respiration slightly alter the magnetic field around brain tissue, leading to corruptions in the true signal the scanner is supposed to detect (4) (the hemodynamic response in relation to stimulus in gray matter tissue, which will be referred to as the stimulus related response, is what we want to keep). For an accurate measure of true brain activity, the scanner drift and physiological noise must be removed without disturbing the latent brain signals present in a given timeseries.

2 Methods

2.1 Data Processing

The data was analyzed using the FNGS[1] leg of the NDMG pipeline (5), a python package for fMRI and Diffusion Tensor Imaging (DTI) analysis. As fMRI is collected, the scanner acquires the entire $3D$ head volume in $2D$ slices over the course of one repetition time (TR, or the time it takes to acquire one $3D$ volume). To account for the fact that the entire $3D$ volume is not collected at the same time and we want to make inferences at discrete points in time, the observed timeseries is interpolated to center the actual acquisitions of each slice using the slicetimer package (6). Following slice timing correction, we motion correct the fMRI scans to correct for variations in signal resulting from the subject moving their head in the magnetic field using the MCFLIRT package (7).

After data preprocessing, the subject is registered to the MNI-152 atlas. Essentially, an atlas is an "average" brain of a large number of (in this case 152) test subjects of varying characteristics (ie, age, ethnicity, etc). The MNI atlas is parcellated by professional psychologists into discrete functional brain regions of interest (ROIs), called a labelled atlas. With our brains registered to the MNI brain space, we then easily mask the brain for each ROI and average the voxels in a labelled brain region over time to approximate the signal in this brain region. For this experiment, we use two labelled atlases: the Desikan 70 parcellation (8) and the Talairach 959 parcellation (9) in which we downsample our timeseries (ie, our ROIs give us a mapping of each voxel in the brain to a particular ROI, and at each timestep, we average the intensities for the voxels corresponding to each ROI).

2.2 Machine Learning Techniques

Here, we explain some of the intuition of each algorithm. See Appendix A for pseudocode[2].

2.2.1 Component-Based Correction

Component-based Correction (CompCor) was performed to remove physiological noise from the observed BOLD signal. Over the course of an experiment, natural physiological fluctuations (blood pressure spikes, breathing, etc) can impact the signal picked up by the fMRI scanner, as they slightly alter the magnetic field passed over tissue (4). As the cerebral blood flow (the BOLD signal detected by fMRI) is 3-7 times lower in the white matter than the gray matter, the white matter tissue signal is more representative of physiological fluctuations than gray matter signal (10). First, we segment our high-resolution T1w-MRI (T1 weighting is a popular technique for acquiring high resolution structural MRI, where structural MRI is used for) into probability maps for each class of brain tissue[3] using the FAST algorithm (11) (giving us the probability that each voxel in the MRI belongs to a particular brain class). We then threshold our probability map for the white matter class for voxels with a probability greater than .99 of being gray matter to ensure that we will have a low probability of masking any non-white matter voxels when we correct[4] our image.

Moreover, the signal at each voxel in an fMRI influences the signal of neighboring voxels. For instance, a voxel representing white matter tissue (whose signal we want to regress) near another voxel representing gray matter tissue (whose signal is stimulus related, which we want to include) will still show some fluctuations due to the stimulus response that is present in the gray matter voxel. If we were to remove signal based on these white matter voxels that have a stimulus-related response present, we would inadverntantly end up fitting variance that directly correlate with the stimulus, and therefore, we would end up removing variance due to the stimulus response (and reduce the viability of our timeseries, since this the stimulus response is what researchers will analyze when making downstream inferences). To avoid this, we erode our mask by 2 voxels in every direction [5], essentially just shrinking the mask volume we have on all sides by 2 voxels.

Finally, using our eroded mask, we mask our fMRI timeseries to obtain the timeseries in our voxels that have the highest probability of being white matter. We then perform principal component analysis to estimate the components contributing the high-

[1] see here for a schematic of the FNGS workflow

[2] Appendix A, Algorithm 1 shows the entire nuisance correction algorithm

[3] cerebrospinal fluid, white matter, and gray matter

[4] See Algorithm 4 in Appendix A

[5] See Algorithm 5 in Appendix A

est variance in the white matter region. These high variance components effectively capture the major components of the white matter signal; as the white matter represents a large portion of physiological noise, we interpret these high variance components as capturing the variance of the physiological noise. For PCA, we assume that our dataset can be decomposed such that (note that here we show the singular value decomposition of a timeseries T):

$$T_{t\times w} = U_{t\times t}\Sigma_{t\times w}V^*_{w\times w} \tag{1}$$

where $\Sigma_{t\times w}$ represents a rectangular diagonal matrix whose diagonal elements are s_t. By convention, we order the singular values s_t in decreasing order. The vectors in the orthonormal basis U can be thought of as capturing the different "dimensions" of the variance; ie, the vectors correspond to different linearly independent vectors in which we could have variance in our data. The singular values, in this case, give us an indication of how much each individual component contributes to the observed data. Therefore, components with a higher singular value are better at "explaining the variance" of the overall data than components with lower values. These lower value components, can be thought of as more random effects, since the data projects to a very narrow scaling along these components. The number of components taken in PCA determines which side of the bias/variance tradeoff we will fall on; picking too many components captures all of the variance in the data, but take bias from random noise in the dataset, whereas picking too few components risks capturing only a small percentage of the variance in the data and thereby give us high model bias.

For our implementation, we consider:

$$U_{t\times t}, s_t, V_{w\times w} = \text{svd}(T_{t\times w})$$

using the singular-value decomposition of the masked white-matter timeseries with w voxels and t timesteps (note that we use the full-rank version of svd)[6]. We assume that the top n (where n is a variable we test in section 4.2) components effectively describe the physiological noise present in our study, as we know that the white matter regions correlate strongly with physiological noise (12).

[6]See Algorithm 2 in Appendix A

2.2.2 General Linear Model

Nuisance correction was performed by fitting a General Linear Model (GLM) to various components known to be nuisance variables, which was shown in (13) to be directly applicable to fMRI nuisance regression. For our n voxels, the t timestep BOLD signal is written as:

$$T_{t\times n} = L_{t\times n} + S_{t\times d}Y_{d\times n} + P_{t\times p}Z_{p\times n} + \epsilon \tag{2}$$

Where L is the true latent signal we wish to observe, SY represents the d dimensional scanner nuisance parameters, PZ represents our p dimensional physiological nuisance parameters, and ϵ measures the error of the scanner measurement. For the purposes of this experiment, our full nuisance model assumes $d = 3$, where we assume that the scanner contributes constant, linear, and quadratic drift to our observed BOLD signal. Additionally, we assume $p = 1 + c$, where we have 1 nuisance variable for the mean CSF signal and c nuisance variables for the number of components of noise we fit in our CompCor method. Assuming the error is low, we rearrange our terms:

$$T_{t\times n} = L_{t\times n} + \beta_{t\times(d+p)}R_{(d+p)\times n}$$

We have an equation T that depends on a constant L with a linear dependence β on the regressors R. Then our goal is to identify:

$$\hat{\beta} = \arg\min_{\beta} ||T - \beta R||_2^2 \tag{3}$$

the coefficients $\hat{\beta}$ that minimize the squared error. To fit these terms, we use the OLS solution for a linear model to fit the model functions β of our nuisance regressors R to the observed timeseries T:

$$\hat{\beta}_{t\times(d+p)} = (R^T R)^{-1}R^T T$$

The OLS solution provides a minimum fit of β given the sum of squared errors (geometric intuition), and is also equivalent to the maximum likelihood estimator of the linear model. We then follow 4 to find the corrected signal L.

$$L = T - \hat{\beta}R \tag{4}$$

3 Work Performed

The most challenging aspect of this project was ensuring that we stayed consistent with the current cutting edge of fMRI processing. We sought the advice of Dr. Cameron Craddock[7], the director of imaging at the Childmind Institute. Dr. Craddock gave us much of the literature cited in this article, and encouraged us of which nuisance reduction methods to stay away from (particularly, in our original plan, we sought to extract the top-n PCs and see how representative of the overall signal they were; according to Dr. Craddock this was heavily debunked (14)). Collectively, we spent roughly ten - fifteen hours perusing literature to formulate our idea. Also, we had to extensively learn about dimensionality reduction, eigendecomposition, SVD, and PCA prior to designing our implementation (since we did not cover these in depth in class), which was probably another ten hours of online lectures and taking notes. Existing literature indicated that SVD was far more robust for the purposes of PCA than eigendecomposition (and unfortunately, a far more complex process to understand and implement). We chose to use the numpy linalg package's implementation of SVD; this package provides a robust, fast implementation that is effective on high dimensional data like fMRI voxel timeseries (approximately 2 million dimensions, and 200-500 observations).

The primary code we needed to implement ourselves was PCA (for Compcor), mask erosion, mask extraction, and GLM. All of our algorithm code is found following this link: NDMG nuisance code. All analysis with the brain images is done using the nibabel package (a python package for loading, saving, and creating nifti images, which store the brain data and associated transformation information about the resolutions of the dimensions). Like many machine algorithms, verifying the success of algorithms performed on brains often relies on qualitative means (although advancements are being made rapidly to this end in the case of brain imaging). To evaluate the quality of our analysis, we added 3 new quality control images to the FNGS leg of the NDMG pipeline which give a good sense of the quality of the segmentation and PCA estimation in the compcor step. Additionally, the FNGS leg already implements a quality control image of a timeseries, which similarly aids in the qualitative assessment of the quality of the nuisance correction (errors in nuisance correction are incredibly apparent in fMRI timeseries; if stimulus related response is removed, the timeseries end up being globally correlated)[8]. We successfully incorporated the nuisance correction module into NDMG, and you can go through a tutorial following the guide here if you are so inclined. Writing the code itself and completely integrating it with our pipeline and web service took approximately fifteen hours once we felt confident in our understanding of PCA.

We spent an extensive amount of time ensuring that we fully understood the implications of our model, and were fully confident in exactly how it would perform on the real data, given our understanding of the problem at hand. We spent approximately ten hours iterating back and forth on simulations, finding simulations that would work and demonstrating that they worked on our model, updating the model based on simulations, updating simulations to fit new weaknesses in the model, and so on. Ultimately, we constructed a simulation that we feel effectively models the role of one of the biggest weaknesses of fMRI processing (see 4.1 for details).

Analyzing brains is quite computationally expensive, and as we committed to analyzing roughly 54 subjects (each with 2 scans) using 5 separate pipelines, we needed to have our algorithms implemented a week and a half in advance just to have enough time to process all the data we wanted to process. We chose one sizable dataset, the BNU1 (54 subjects, 108 scans) (15). For analysis, we used a cluster located in Clark Hall (owned by Dr. Vogelstein). As this cluster does not have any dependencies installed, to run this pipeline required the deployment of a docker container, a self-contained virtual machine containing all the dependencies for a project. To run the data in the small amount of time we had, we repeated a process of manually tweaking the pipeline for whatever settings we needed, deploying ten instances with this pipeline setup installed, and then running all ten instances in parallel on subsets of the data. With automation, the human

[7] Dr. Craddock's bio

[8] See Appendix B for Quality control details

component still took roughly ten hours; the setup of each individual docker instance took a fair amount of time just by way of the sheer number of instances that had to be deployed (roughly 150 over the course of the week) for us to get our data in time.

For post-processing the data, we were fortunately able to recycle a lot of code used in (16). We spent some time discussing visualization methods of our multi-pipeline effort, and ultimately settled on looking at a line plot of each discriminability score with respect to the pipeline option as our "summary" figure. Additionally, in the appendix we added summaries of the discriminability statistic, summary plots for each individual pipeline and what to look for, and finally examples of some data derivatives. The total time spent on the formulation and analysis phase of the project (excluding the final writeup) is approximately 70-80 hours; split between the two of us, approximately 35-40 hours per person before the writeup (which took us a very long time).

4 Results

4.1 Simulations

To structure our simulations [9], we looked at modeling each nuisance parameter approximately representative of how it might actually be represented in real signal. We use a simulated brain with 3 ROIs corresponding to one cerebrospinal fluid ROI, one gray matter ROI, and one white matter ROI. To try to make our experiment as close to real as possible, we pick 200 timesteps (which is an average number of timesteps for an experiment) and we assume a TR of 2 seconds (ie, each slice is taken at 2 second intervals). We define the latent timeseries for each ROI as follows, where the latent signal is the "truth" signal for each ROI:

$$
\begin{aligned}
C_l(t) &= \mathcal{N}(0, \Sigma_c) \\
W_l(t) &= a_1 \sin_1(b_1 t) + a_2 \sin_2(b_2 t) + \mathcal{N}(0, \Sigma_w) \\
G_l(t) &= a_g \sin_g(b_g t) + W_l(t)
\end{aligned}
$$

where the two sinusoids in the white matter latent ROI represent expected physiological noise (heart rate and breathing by the sinusoids, and random noise from the normal distribution). Note that the gray matter includes both a stimulus related response ($\sin_g$) as well as the physiological noise present in the white matter. We define our observed timeseries:

$$
\begin{aligned}
C_o(t) &= C_l(t) + D(t) + \mathcal{N}_\epsilon(0, \Sigma_\epsilon) \\
W_o(t) &= W_l(t) + D(t) + \mathcal{N}_\epsilon(0, \Sigma_\epsilon) \\
G_o(t) &= G_l(t) + D(t) + \mathcal{N}_\epsilon(0, \Sigma_\epsilon)
\end{aligned}
$$

Where we add the scanner drift parameter to our observed signal $D(t)$ and an error parameter to account for any errors in signal measurement that take place. For our simulations, we attempt to model a latent sine wave in the gray matter and, given the corruption of the signal, see how effectively we can recover this sine wave using our nuisance correction module[10]. We measure this effectiveness by employing the R^2 metric, which measures how well an estimated data series fits a known one:

$$
\begin{aligned}
R^2 &= 1 - \frac{SS_{res}}{SS_{tot}} \\
SS_{tot} &= \sum_i (y_i - \bar{y})^2 \\
SS_{res} &= \sum_i (y_i - f_i)^2
\end{aligned}
$$

where SS_{tot} is the total sum of squares and SS_{res} is the sum of squares of residuals. The R^2 metric indicates how well our post-correction GM ROI matches the desired sine wave. We consider the data points in our desired sine wave to be the y_i and the post-correction ROI points as f_i. With this setup, we evaluate and compare the performance of nuisance correction on simulations with different parameters.

Here we see that nuisance correction performs very effectively at removing noise that is consistent from ROI to ROI. As can be seen in 1, if we have some high dimensional noise that is present in our white matter as well as our gray matter ROI, we will be able to effectively remove it using our correction technique. However, the primary introduction of error from our estimation, we found, was varying the error term $\mathcal{N}_\epsilon(0, \Sigma_\epsilon)$ which is a different sampling of measurement error for each ROI.

We see in 2 that the effectiveness of our nuisance correction model is correlated to the measurement

[9] Our simulation code is found simulations

[10] See Appendix C for how we chose parameters in our simulation setup

Figure 1: Effect of nuisance correction on simulated grey matter ROI. We compare the latent signal to the post-correction signal (top) to show how much of the sinusoid we are able to recover from the corrupted, pre-correction gray matter signal (bottom, red line).

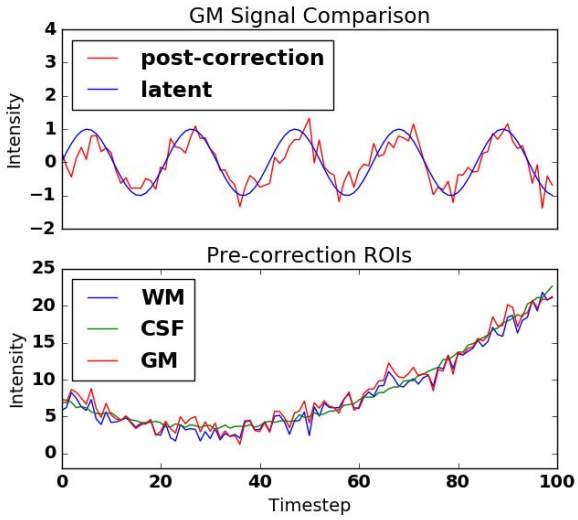
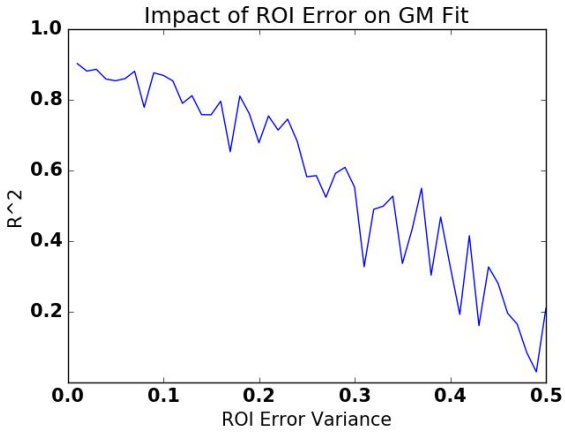


Figure 2: Effect of increasing measurement error variance $\mathcal{N}_\epsilon(0, \epsilon)$ on fit between post-correction and latent grey matter signals.

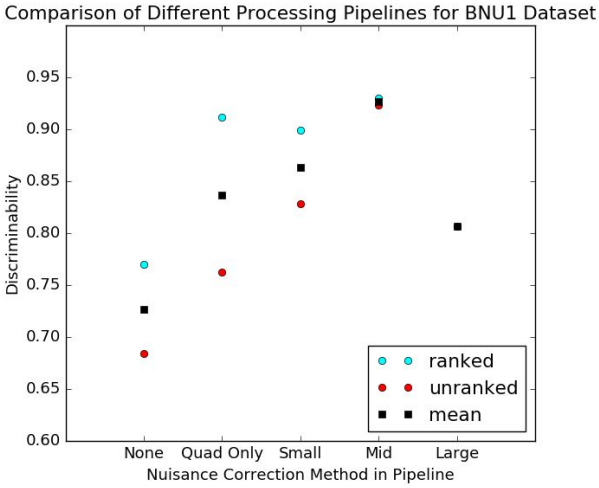


error levels. As expected, the fit is near perfect at low variances and gets significantly worse as the variance increases. When the variance of error increases to $\epsilon = 0.5$, we are not able to effectively determine the latent signal from the pre-correction timeseries, indicated by a low R^2 score.

4.2 Real Data

To examine the benefit of our nuisance correction module, we consider five potential processing pipelines, and analyze all 108 scans on each pipeline[11]. We consider 1 pipeline with no nuisance correction at all (raw fMRI timeseries), 1 pipeline with only quadratic drift removal (GLM model from above, with only the $D(t)$ regressor), 1 pipeline with the top 5 white matter components removed, 1 pipeline with the top 50 white matter components removed, and 1 pipeline with the top 150 white matter components removed. All pipelines except for the "no nuisance" pipeline also have the mean signal in voxels that are cerebrospinal fluid removed. We then downsample our timeseries (which are approximately one million dimensions, 200 observations in time) as described above to the desikan parcellation atlas (with 70 labelled ROIs). To compare our pipelines, we use the discriminability statistic outlined in (16) [12]. where a higher discriminability score indicates more discriminable connectomes.

Figure 3: A comparison of different pipeline options investigated. The $n = 50$ mid pipeline shows the highest discriminability, with a mean score of 0.92 between the ranked and unranked models.

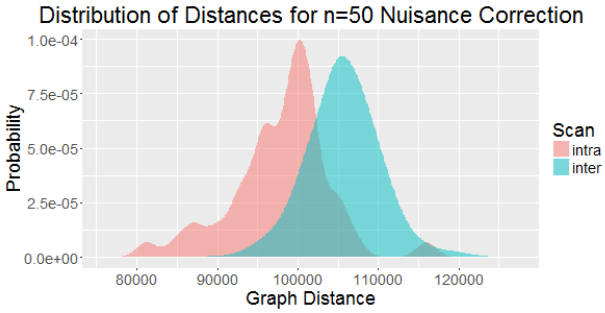


The best pipeline option is nuisance correction

[11] See Appendix F for examples of timeseries data derivatives for several of the pipelines, as well as a scree plot of the components being corrected.

[12] See Appendix D for details

Figure 4: A density estimate of the intra subject (same subject, different scan) and inter subject (different subject) graphs for the best pipeline, $n = 50$ nuisance correction. More discriminable pipelines will show greater separation of the intra and inter subject density estimates.



with fifty component removal and ranking, which shows a discriminability score of 0.92. We had expected the pipeline with $n = 5$ nuisance regressors to perform best (this was the one recommended to us by Dr. Craddock), so this was certainly a surprise for us. This may have been the case because the $n = 5$ pipeline does not effectively capture the highest singular value components in the dataset before we reach an elbow in the scree plot [13]. In future investigations, it is clear based on the relatively small fluctuations in discriminability for the quad, $n = 5$ small, and $n = 50$ mid pipelines that we need much more data to make an accurate conclusion. Given more time, we would have expanded our investigations to the remainder of the CORR dataset (15), which features a full 2000 subjects of data and almost 6000 scans (some datasets have more than 2 scans per subject). In its current state, one or two outliers in a particular processing pipeline would lead to a difference in the discriminability of about 0.036. [14], which exceeds the amount that our best pipeline exceeeds our second best pipeline (which appears to be the small $n = 5$ pipeline based on mean discriminability). A planned intercession project is to look at task-fMRI datasets (in which we know we will expect a stimulus related response, since task fMRI involves presenting stimulus such as questions at known intervals and seeing how different brain regions respond). Then, we can see how components in the white matter correlate with the stimulus response in the gray matter. This will allow us to more effectively choose a particular threshold of principal components, or choose an algorithm to select a threshold on a per subject basis, as we will know exactly what the stimulus response that we want to keep looks like.

[13] See Appendix F for the scree plot of the singluar values

[14] see Appendix D for where this number comes from.

5 Comparison to Original Proposal

We appear to have successfully implemented an effective nuisance correction algorithm for fMRI, and were able to effectively merge it with our original FNGS leg of the NDMG python package. One of the primary aspects that we were not able to deliver compared to our original proposal was that we found after meeting with Dr. Craddock that Principal Component Regression is non-viable for fMRI. It has been shown that regressing the top PCs corresponds approximately to regressing to the global signal of the fMRI investigation. Delving into the article Dr. Craddock recommended, we found that fitting to the global signal apparently zero-centers the correlation graphs we use as our inference criterion for this experiment, which is not representative of the actual performance of brain regions (14). This interesting finding did not have much of an impact on our overall progress, however, as we were able to effectively pivot and focus our efforts on a more robust investigation of compcor than we had originally planned. Dr. Craddock also recommended we shift our goal of looking at both different thresholds of variance and number of components to only looking at different numbers of components in our nuisance, and just check a wider range of components than we had originally planned.

Results wise, we constructed an effective simulation to show the results of our data, and were able to find a very significant experimental criterion in the scanning session that would significantly impact the confidence of our nuisance regression (we found that errors between voxels would break our model most significantly; errors that were consistent among voxels could be removed with incredible effectiveness). Finally, we were able to effectively demonstrate our method on a substantial real dataset, and had our algorithm completed on time to be able to analyze a full dataset of subjects on all five of our pipelines.

Acknowledgments

References

[1] S Ogawa, T M Lee, A R Kay, and D W Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences of the United States of America*, 87(24):9868–72, dec 1990.

[2] Richard B. Buxton, Kâmil Uluda, David J. Dubowitz, and Thomas T. Liu. Modeling the hemodynamic response to brain activation. *NeuroImage*, 23:S220–S233, jan 2004.

[3] Anne M Smith, Bobbi K Lewis, Urs E Ruttimann, Frank Q Ye, Teresa M Sinnwell, Yihong Yang, Jeff H Duyn, and Joseph A Frank. Investigation of Low Frequency Drift in fMRI Signal. 1999.

[4] X Hu, T H Le, T Parrish, and P Erhard. Retrospective estimation and correction of physiological fluctuation in functional MRI. *Magnetic resonance in medicine*, 34(2):201–12, aug 1995.

[5] Gregory Kiar, William R Roncal, Eric Bridgeford, Disa Mehembere, Randal Burns, Joshua T Vogelstein, et al. NDMG: NeuroData Mri Graphs. *GitHub*, 2016.

[6] Ronald Sladky, Karl J. Friston, Jasmin Tröstl, Ross Cunnington, Ewald Moser, and Christian Windischberger. Slice-timing effects and their correction in functional MRI. *NeuroImage*, 58(2):588–594, sep 2011.

[7] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–41, oct 2002.

[8] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, Marilyn S Albert, and Ronald J Killiany. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. 2006.

[9] J L Lancaster, M G Woldorff, L M Parsons, M Liotti, C S Freitas, L Rainey, P V Kochunov, D Nickerson, S A Mikiten, and P T Fox. Automated Talairach atlas labels for functional brain mapping. *Human brain mapping*, 10(3):120–31, jul 2000.

[10] Jodie R. Gawryluk, Erin L. Mazerolle, and Ryan C. N. D'Arcy. Does functional MRI detect activation in white matter? A review of emerging evidence, issues, and future directions. *Frontiers in Neuroscience*, 8:239, aug 2014.

[11] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, jan 2001.

[12] Y Behzadi, K Restom, J Liau, and T T Liu. A component based noise correction method (CompCor) for BOLD and perfusion based fMRI. *Neuroimage*.

[13] G M Boynton, S A Engel, G H Glover, and D J Heeger. Linear systems analysis of functional magnetic resonance imaging in human V1. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 16(13):4207–21, jul 1996.

[14] Kevin Murphy, Rasmus M. Birn, Daniel A. Handwerker, Tyler B. Jones, and Peter A. Bandettini. The impact of global signal regression on resting state correlations: Are anti-correlated networks introduced? *NeuroImage*, 44(3):893–905, feb 2009.

[15] Xi-Nian Zuo and et al. An open science resource for establishing reliability and reproducibility in functional connectomics. *Scientific Data*, 1:140049, dec 2014.

[16] Shangsi Wang, Zhi Yang, Xi-Nian Zuo, Michael Milham, Cameron Craddock, Gregory Kiar, William Gray Roncal, Eric Bridgeford,

Carey E Priebe, and Joshua T Vogelstein. Optimal Decisions for Discovery Science via Maximizing Discriminability: Applications in Neuroimaging. *request for preprint*.

1 Appendix A: Algorithms

1.1 Nuisance Correction

Algorithm 1: Nuisance Correction

Input:

f $\in R^{x,y,z,t}$: the fMRI timeseries loaded into a matrix.

a $\in R^{x,y,z}$ the anatomical image data.

$a_0 \in R^{x,y,z}$ an integer to indicate the type of anatomical image: 1 for T1w, 2 for T2w, 3 for PD

m $\in R^{x,y,z}$: the binary mask associated with the brain we register to in the registration step.

Result:

n $\in R^{x,y,z,t}$: a nuisance corrected brain in a matrix form.

v = f(m, :).T // extract the voxel timeseries over the mask we are registered to and transpose so that time is the 0th dimension

fast f -n 3 -t a_0 // use FSL's FAST to segment the brain into white matter, grey matter, and cerebrospinal fluid probability maps. each value will indicate a probability of a particular voxel being part of each class (WM vs GM vs CSF).

wm = load_nifti(/path/to/WM/mask) // load the white matter mask into a matrix

self.extract_mask(wm, .99) // thresholds the white matter mask to get all voxels with p > .99 of being WM

ewm = self.erode_mask(wmm, 2) // erodes wmm by 2 voxels

wm_ts = f(ewm,:).T // extract the timeseries for the white matter regions over the course of the session

csf = load_nifti(/path/to/csf/mask) // the csf mask in a matrix

csfm = self.extract_mask(csf, .95)

csfm = self.erode_mask(csf, 2)

csf_ts = f(csfm, :)

r = [] // begin an array of regressors, where r $\in R^{t,n_r}$

r(:,0) = 0:t // linear drift regressor

r(:,1) = $(0{:}t)^2$ // quadratic drift regressor

r(:,2) = mean(csf_ts, axis=0) // take the mean csf signal over all voxels at every time point

comp = self.compcor(v, wm_ts, c) // Calculate compcor regressors with wm mask, where c is either a fixed number of components or a threshold of variance explained

r(:,3) = comp(0) // the components element returned by compcor method

W = self.regress_signal(v, r) // use the OLS solution for a GLM given regressors

n = (v - W).T // subtract out regressors to remove, giving us our nuisance corrected data and transpose back to standard space

return n

Algorithm 2: CompCor

Input:
mask_ts $\in R^{t,n}$ the timeseries extracted over a particular mask that we want to perform component correction for.
c $\in R^1$: the number of components to calculate.

Result:
U(:,0:c) $\in R^{t,c}$: the components we will regress out.

```
mask_ts = self.normalize_signal(mask_ts) // detrend and normalize by std
U, s, V = np.linalg.svd(mask_ts, full_matrices=False) // singular value decomposition; use numpy
  here because our data is very high dimensional and dense
return (U(:, 0:c), s)
```

Algorithm 3: Normalize Signal

Input:
v $\in R^{t,n}$ a voxel timeseries.

Result:
v $\in R^{t,n}$ a voxel timeseries that is mean-centered and normalized by standard deviation.

```
v = v(:, std(v, axis=0) != 0) // remove voxels with no standard deviation, since normalizing will
  give them NaN values
v = v - mean(v, axis=0)
v = elementwise_divide(v, std(v, axis=0)) // normalize each voxel by the standard deviation
return v
```

Algorithm 4: Extract Mask

Input:
m $\in R^{t,n}$ a thresholded probability map for a particular class of brain matter (ie, white matter, grey matter, CSF).
t $\in R^1$ a threshold over which to consider a member of that class. Ie, if we define this as .99, that means that all voxels in our probability map with probability $> .99$ will be considered the class the map represents.

Result:
mask $\in R^{t,n}$ the mask.

```
mask = (m > t) // elements of probability map greater than threshold
saveimg(mask)
return mask
```

Algorithm 5: Erode Mask

Input:
m $\in R^{t,n}$ a brain mask to erode.
$n_v \in Z^1$ the number of voxels to erode, where erosion means that wherever we consider a masked segment "part" of a particular class, we also remove this number of voxels in every direction around the masked segment.

Result:
e $\in R^{t,n}$ the eroded mask.

```
for i = 1 : v do
    e = zeros(m.shape)
    x,y,z = idswhere(m! = 0) // get coordinates of masked entries
    for j = 1 : length(coords) do
        if (m[x,y,z] & m[x + 1,y,z] & m[x,y + 1,z] & m[x,y,z + 1] & m[x − 1,y,z] & m[x,y − 1,z] &
          m[x,y,z − 1]) then
            eroded[x,y,z] = 1
    end
    m = e
end
```

1.2 Simulations

Algorithm 6: Simulate Brain

Input:
t $\in R^{1,50}$: "timeseries" representing grey matter truth
$\Sigma_w \in R^d$: list of variances for each dimension of white matter noise
$\Sigma_c \in R^1$: variance to use for near-constant CSF region of interest
a, b, c : quadratic coefficients for drift included in observed timeseries

Result:
brain $\in R^{3,50}$: a simulated brain with a wm, csf, and gm ROI.

wmmask = zeros((3,1,1))
wmmask[0] = 1 // white matter mask
lvmask = zeros((3,1,1))
lvmask[1] = 1 // csf mask
save(wmmask, lvmask)
sw=0
for *i in 1:d* **do**
  sw += $\mathcal{N}(\mu = 0, \Sigma = \Sigma_{w_i})$ // sampled white matter ROI
end
sc = $\mathcal{N}(\mu = 0.5, \Sigma = \Sigma_c)$ // sampled CSF ROI
drift = $at^2 + bt + c$ // $R^{1,50}$ array for drift
ow = sw + d
oc = sc + d
og = sw + sc + t + d
brain[0,:] = ow
brain[1,:] = oc
brain[2,:] = og // $R^{3,50}$ stack of three ROIs
save(brain)

Algorithm 7: Analyze Brain

Input:
p $\in R^1$: measure of the period for the sinusoidal grey matter truth
amp $\in R^1$: measure of the amplitude for the sinusoidal grey matter truth

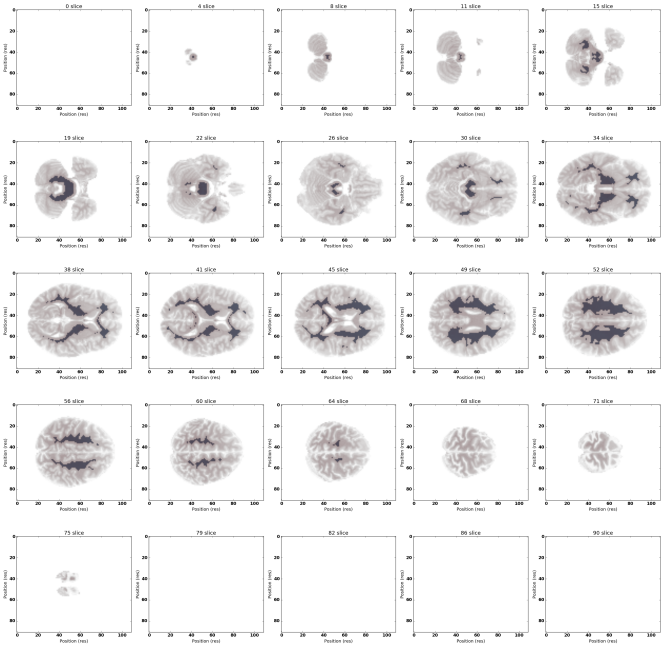
Result:
$r2 \in R^1$: value between GM truth and post-correction retained GM

t = sineWave(period = p, amplitude = amp) // create grey matter truth signal
createBrains(t) // call function to create simulation brains
b = load(simulationBrains) // load brains created by createBrains function
regressNuisance(b) // perform nuisance correction on simulated brain
// R-squared value calculation
$SS_{tot} = \sum(t - \bar{t})^2$
$SS_{res} = \sum(t - cg)^2$ // cg is the corrected grey matter roi signal
r2 = $1 - \frac{SS_{res}}{SS_{tot}}$ // r^2 coefficient
return r2

2 Appendix B: Quality Control

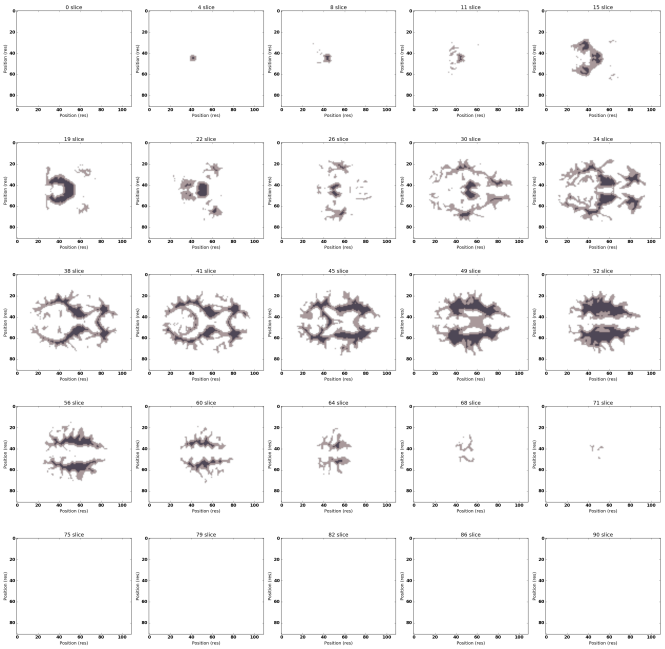
2.1 Segmentation

To evaluate the quality of image segmentation, we look at a slicewise plot of the white matter mask (dark) overlaid with the anatomical image (faded). In a properly segmented image, the white matter will appear to mostly consiste of the central portion of each individual lobe. Below is an example of a properly segmented image.



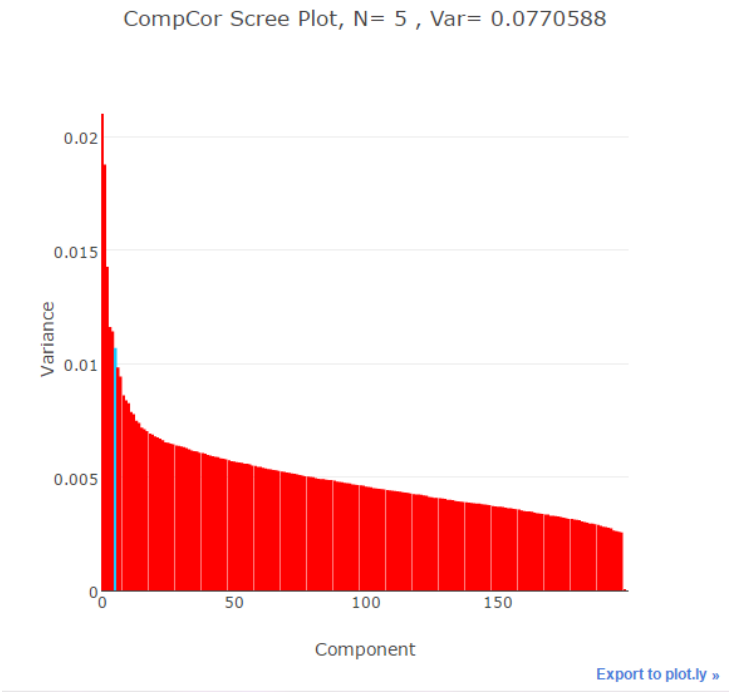
2.2 Mask Erosion

To evaluate the quality of mask erosion, we look at a slicewise plot of the eroded white matter mask (dark) overlaid with the white matter mask (faded). In a properly eroded image, the eroded mask will appear to be a "smaller" version of the original mask, as it is the original mask reduced in size by 2 voxels in every direction of the $3D$ brain.



2.3 Scree Plots

To evaluate the quality of the components estimated in compcor, we look at a scree plot of the expected variance of each component. We should see that the expected variance of the top components is relatively high, and quickly levels off in the less-significant components. We do a more in depth look at scree plots in appendix F: 6.2.



3 Appendix C: Simulations

We define latent timeseries for each simulated ROI:

$$
\begin{aligned}
C_l(t) &= \mathcal{N}(0, \Sigma_c) \\
W_l(t) &= a_1 \sin(b_1 t) + a_2 \sin(b_2 t) + \mathcal{N}(0, \Sigma_w) \\
G_l(t) &= a_g \sin(b_g t) + W_l(t)
\end{aligned}
$$

In the white matter ROI, we assume that we have two physiological noise stimuli here (the first is heart rate, the second for respiration). Since the hemodynamic response takes roughly 16 seconds for a full response, we choose a period of $\frac{16}{TR} = \frac{16}{2} = 8$ timesteps. Similarly, we choose b coefficients to make the period of the heart rate approximately 2 seconds, and the period respiration approximately eight seconds (participants are to remain calm in the scanner during resting state fMRI, so their breathing and heart rate should be relatively constant). We arbitrarily set the amplitudes $a_1 = a_2 = .2a_g$ (theoretically, our brain signal will be stronger than any physiological noise that the scanner might pick up) where the main factor of our accuracy will not be specific values we choose to set our parameters to but rather the magnitude of parameters in the scope of the parameter set.

We define our observed timeseries:

$$
\begin{aligned}
C_o(t) &= C_l(t) + D(t) + \mathcal{N}_\epsilon(0, \Sigma_\epsilon) \\
W_o(t) &= W_l(t) + D(t) + \mathcal{N}_\epsilon(0, \Sigma_\epsilon) \\
G_o(t) &= G_l(t) + D(t) + \mathcal{N}_\epsilon(0, \Sigma_\epsilon)
\end{aligned}
$$

where we add both an error parameter $\mathcal{N}_\epsilon(0, \Sigma_\epsilon)$ that is normally distributed and uniquely sampled for each ROI as well as a scanner drift quadratic $D(t) = at^2 + bt + c$ to model the impact of scanner heating on our observed signal.

4 Appendix D: Pipeline Interpretations

4.1 Discriminability

The discriminability statistic essentially measures a "fingerprint" of each brain. We compute the correlation coefficient between the timeseries of each pair of ROIs in the brain to obtain a connectivity matrix (or connectome). Using the discriminability statistic, we are able to compare the similarity of brains from different scan sessions. For example, if each subject's paired scan is distinguishable with perfect accuracy (ie, if given an array of connectomes and one reference connectome for a particular subject, we can perfectly identify which other connectome is also from this subject) we will have a discrminiability statistic at the upper bound of 1. If, on the other hand, we have absolutely no idea which scan is paired with this particular subject, we would have a discriminability value near 0.5.

The description of discriminability can be found discrminability tutorial, but key pieces are summarized briefly here. Essentially, when analyzing brains, we are attempting to find the most discriminable pipeline, with the idea that if resting state fMRI is reliable, each subject's individual scans should be unique, with inferences being unique to that subject apparent enough to be able to identify connectivity patterns that are unique to that subject. As such, a pairwise comparison of individual connectomes should yield that those of individual subjects show relatively consistent correlation values. We are seeking in this experiment to find the processing options ψ (in this case, the set of the 5 pipelines) and post processing options ϕ (in this case, just considering ranking and non-ranking in the analysis) that satisfy the objective:

$$\max_{\psi,\phi} d(\psi, \phi)$$

When considering the entire dataset, the discriminability can be thought of as an average for each scans's individual "reliability density" score r_i, for $n = |scans|$. Another way, the dataset discriminability d can be written:

$$d(\psi, \phi) = \sum_{i=1}^{n} \frac{r_i}{n}$$

We begin our computation by first computing the pairwise distances between each observed correlation matrix x, y, one for each scan:

$$D_{x,y} = ||C_x - C_y||_2^2$$

Therefore, our distance matrix D will be $D_{n \times n}$. For each individual scan i, we order the individual array of distances in ascending order, where a lower ranking indicates greater similarity in the connectomes.

$$q_i = \text{rank}(D(i, :))$$

We then find the index i' corresponding to the rank of the second scan for this particular subject, and our reliability score is:

$$r_i = 1 - \frac{i'}{n}$$

So if a particular scan has its corresponding second scan (ie, the other scan of the same subject) ranked 0 (assuming zero indexing), then it gets a perfect reliability score of 1; if the corresponding second scan is non-distinguishable from the current scan, it will get a reliability score somewhere around 0.5, and if the corresponding scan is the most dissimilar from our current scan, then it gets a value of 0.

Like many statistics, the discriminability is only as good as the size of the sampling you feed it. Simply having a small number of outliers in one direction or another (ie, if our pipeline inadvertantly misprocesses

a particular scan, leading to it being poorly ranked in relation to the other subject). Simply having two subjects out of 108 mixed up in the discriminability computation would impact our discriminability by:

$$\frac{2 \text{ subjects}}{108 \text{ scans}} \frac{2 \text{ scans}}{\text{subject}} = \frac{.036}{\text{match}}$$

Since if one subject's timeseries gets misprocessed, it will probably get a near 0 score in the reliability ranking.

4.2 Ranking

Points with labels of "ranked" include ranking the connectome from lowest to highest value with a scalar. We detail this process in [1], but essentially our hypothesis is that allows relative robustness to connectivity outliers, is affine-transformation invariant (ie, a global scaling of the correlations will not be reflected after ranking), and it allows us to recover monotonic nonlinear fluctuations that may be present better than a linear transformation such as a z-scoring of the correlations.

5 Appendix E: More Real Data

Below, we show the distance matrices for different processing pipeline options combined with the distribution of the distance weights. In the distance plot, the scans are organized by subject label; that is, scan 1 corresponds to subject 1, scan session 1, scan 2 corresponds to subject 1, scan session 2, scan 3 corresponds to subject 2, scan session 1, and so on. Since we want to intra subject relationships (same subject, different scan session) to show better discriminability than the inter subject relationships (different subject), we naturally want closer distances to occur along the diagonal. The density estimate is classified by intra subject and the inter subject density estimates of graph distances, where we want the intra subject density estimate to show greater separation from the inter subject density estimate.

5.1 Large $n = 150$ Pipeline

Figure 1: Graphs ranked before computing distance; $n = 150$ large pipeline, mnr=.81.

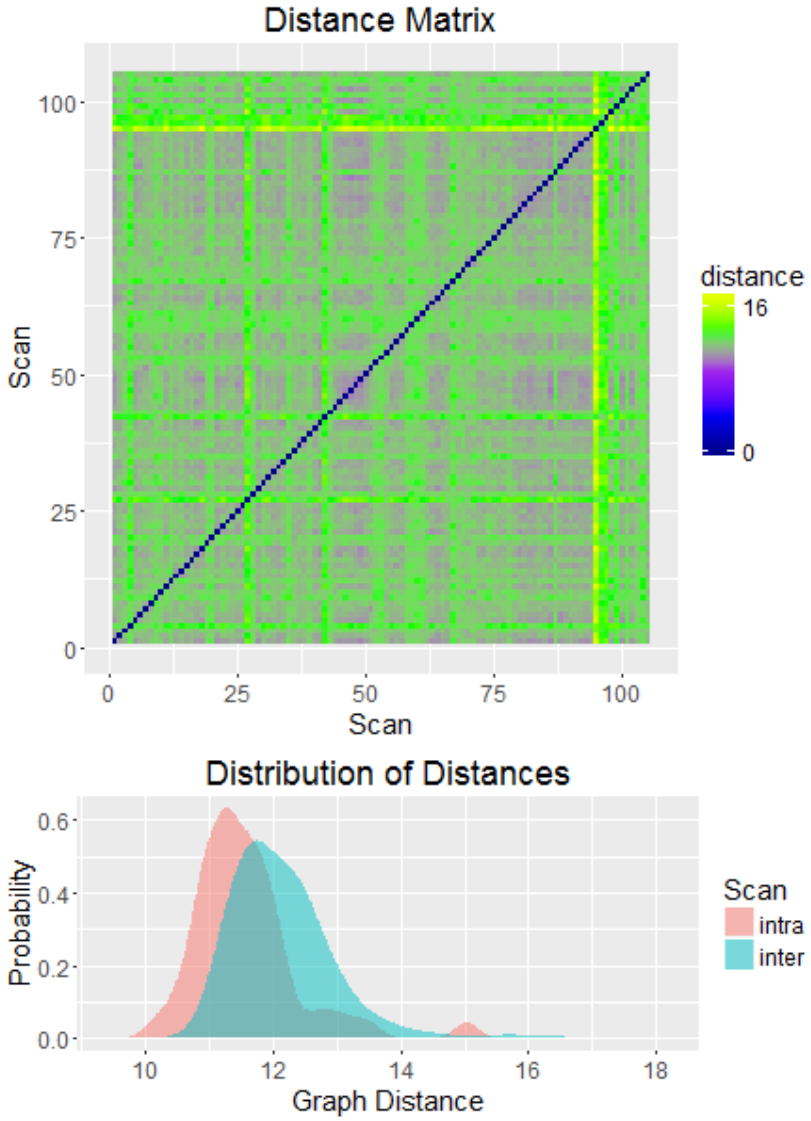
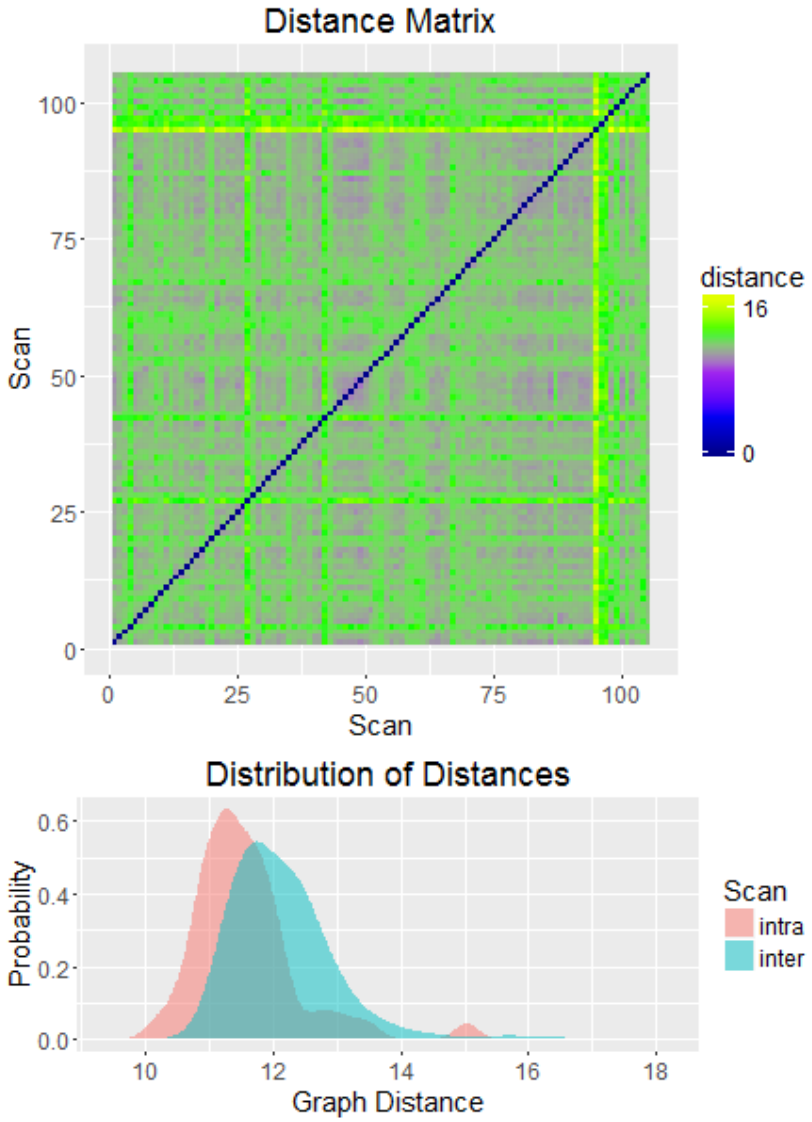


Figure 2: Graphs unranked when computing distance; $n = 150$ large pipeline, mnr=.81.



5.2 Mid $n = 50$ Pipeline

Figure 3: Graphs ranked before computing distance; $n = 50$ mid pipeline, mnr=.93.

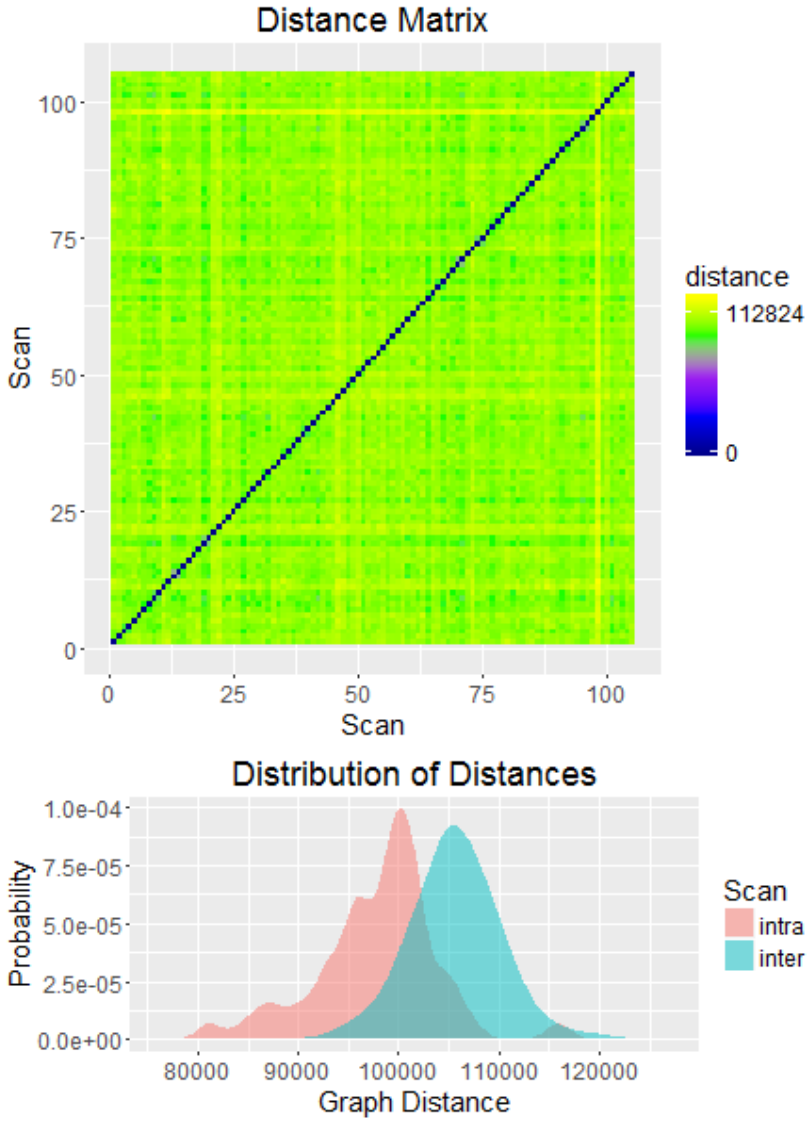
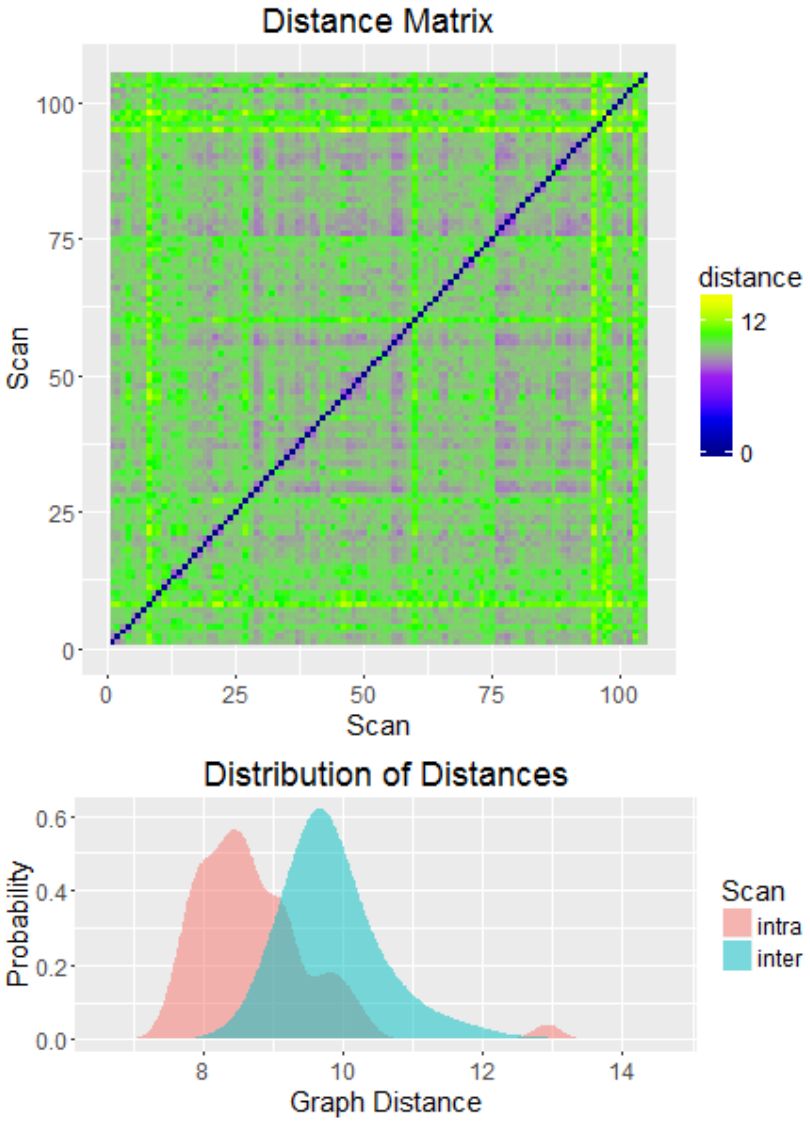


Figure 4: Graphs unranked when computing distance; $n = 50$ mid pipeline, mnr=.92.



5.3 Small $n = 5$ Pipeline

Figure 5: Graphs ranked before computing distance; $n = 5$ small pipeline, mnr=.83.

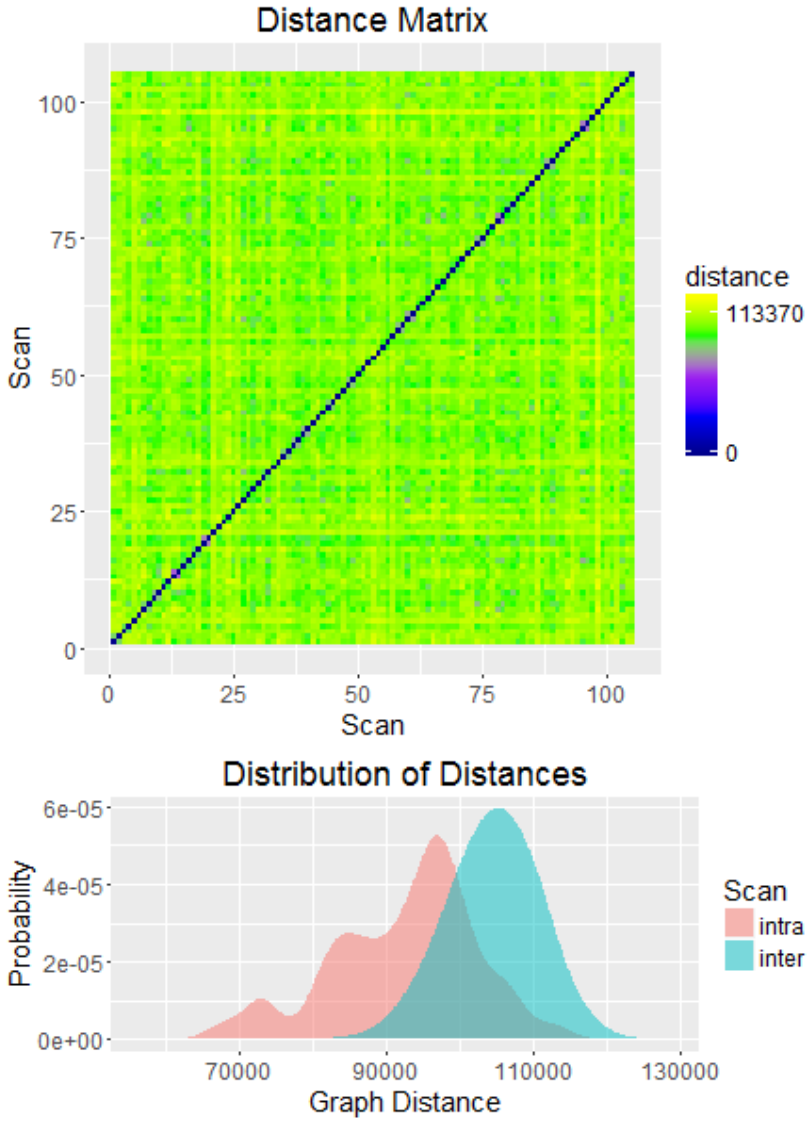
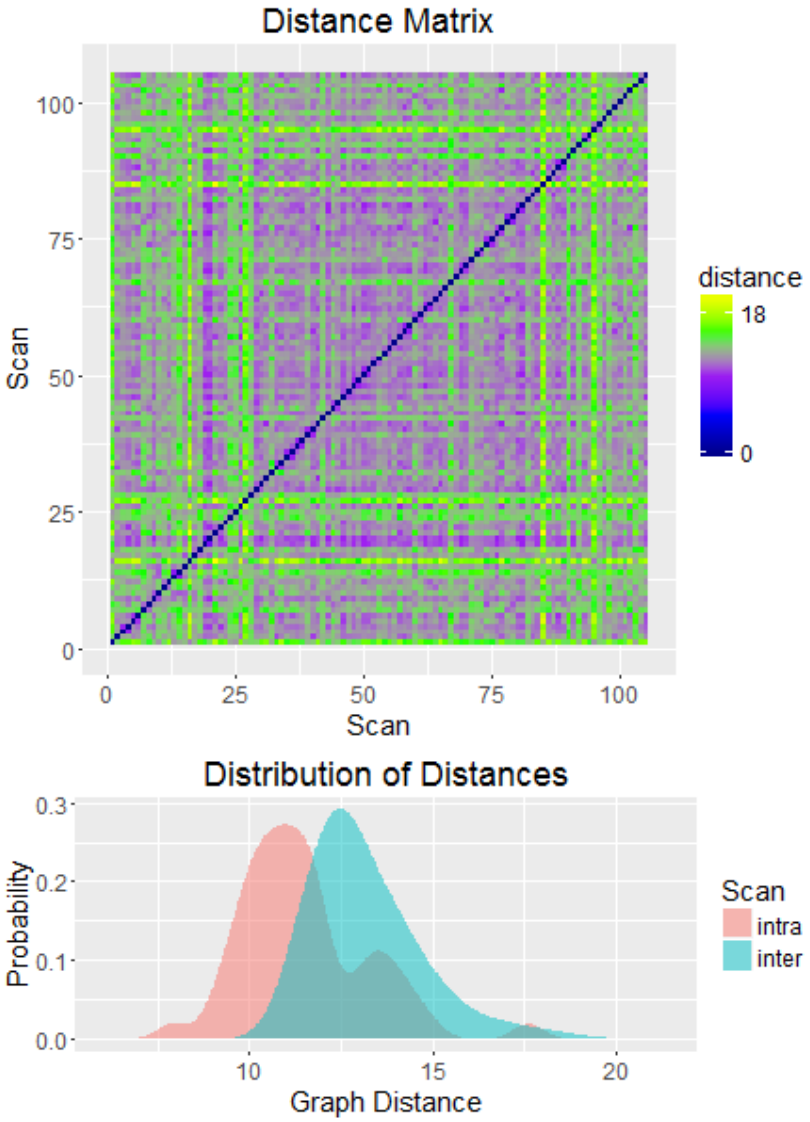


Figure 6: Graphs unranked when computing distance; $n = 5$ small pipeline, mnr=.89.



5.4 Quadratic Drift Removal Only Pipeline

Figure 7: Graphs ranked before computing distance; quadratic drift removal only pipeline, mnr=.91.

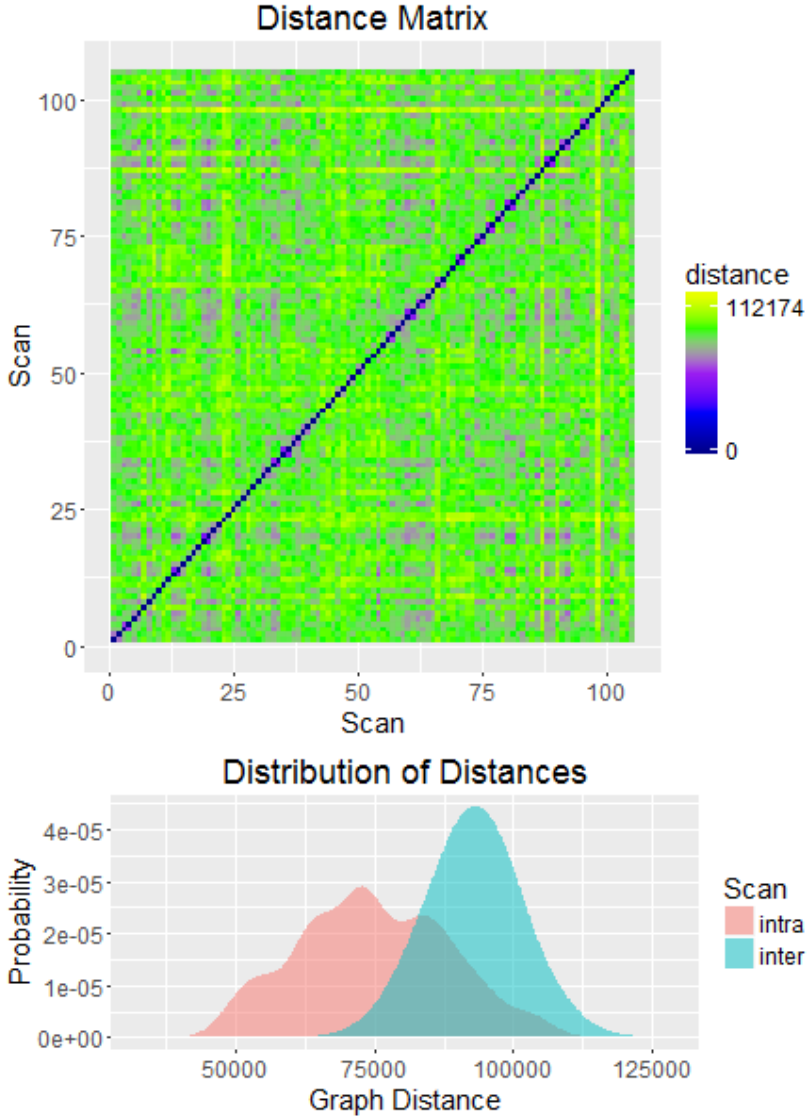
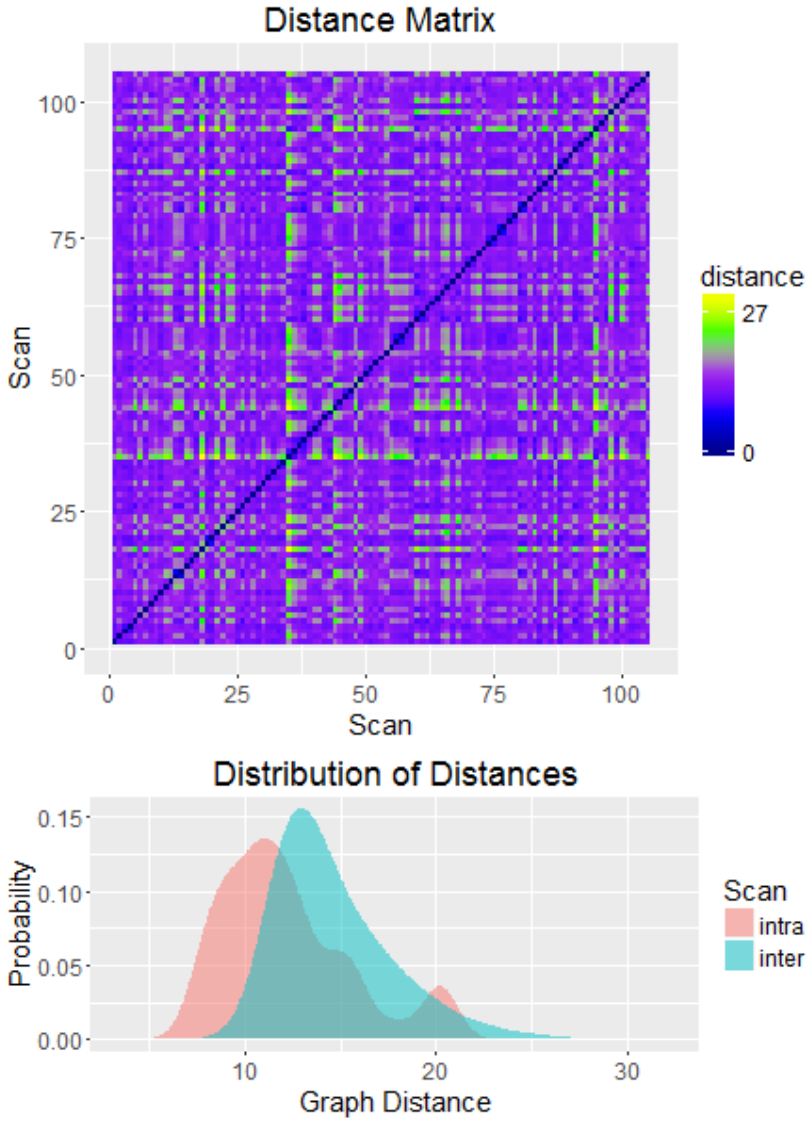


Figure 8: Graphs unranked when computing distance; quadratic drift removal only large pipeline, mnr=.76.



5.5 No Nuisance Correction Pipeline

Figure 9: Graphs ranked before computing distance; no correction, mnr=.76.

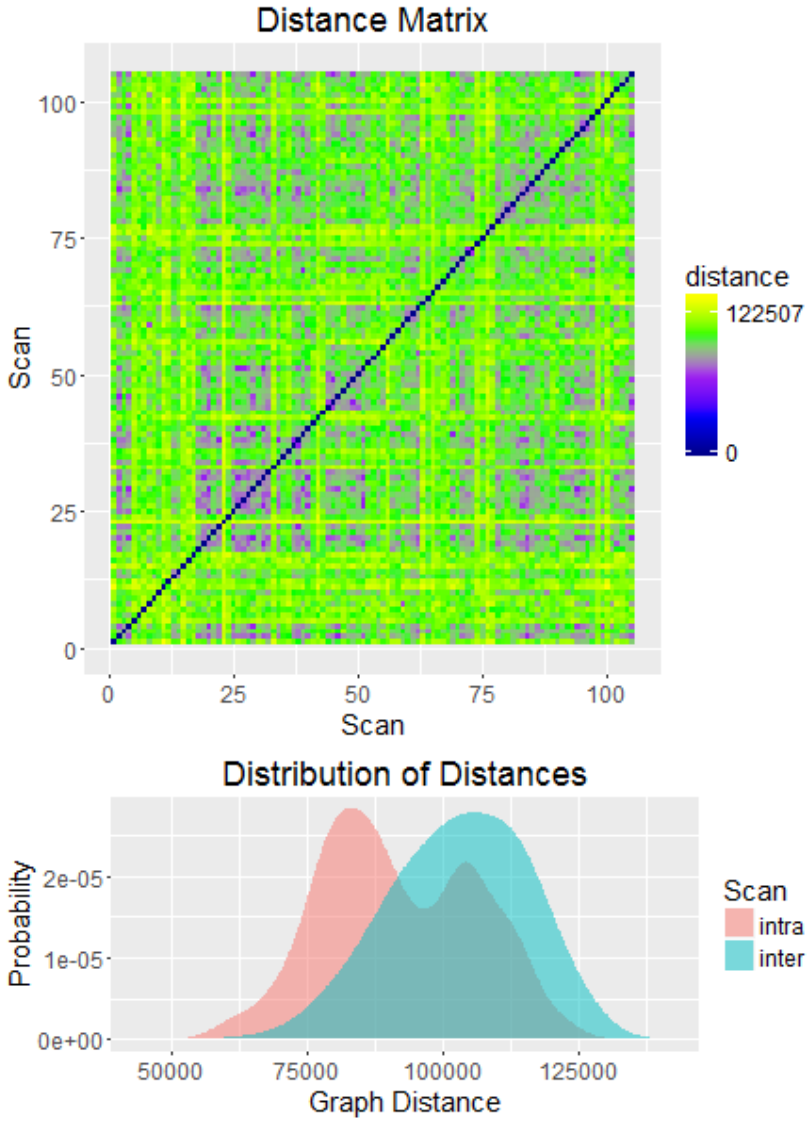
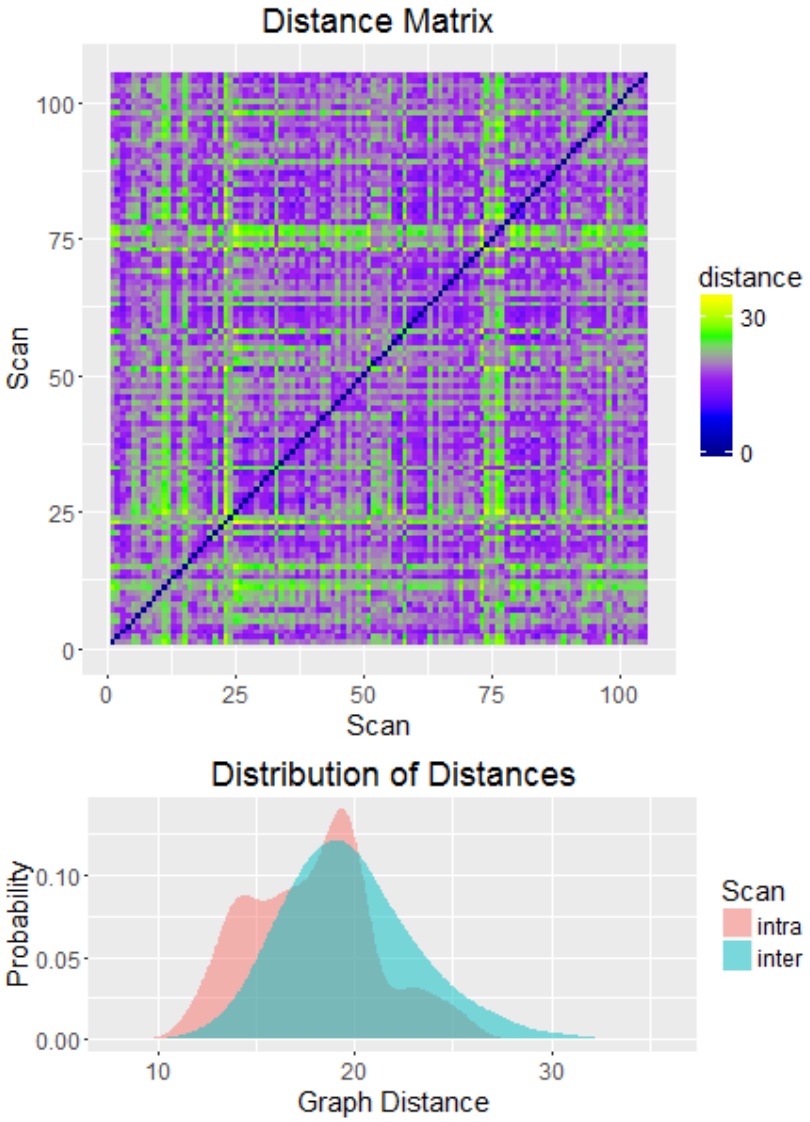


Figure 10: Graphs unranked when computing distance; no correction, mnr=.69.



6 Appendix F: Examples of Data Derivatives

6.1 Timeseries

To put the method into context, here we show an example timeseries without nuisance correction (an example from the no nuisance pipeline) compared to one with nuisance correction (note we updated the quality control figures between the time these two figures were taken, so one is significantly more aesthetically pleasing, but this should give you an idea of what nuisance correction actually does).

Figure 11: Uncorrected fMRI timeseries. Note that the timeseries for each ROI are completely different in intensity (the constant scaling does not contain any valuable information, as we only want to measure fluctuations in activation with fMRI since different regions will show higher intensities just by having more arteries flowing in them), and if you look closely, the figure appears to be slightly "crooked" due to a very slightly linear drift.

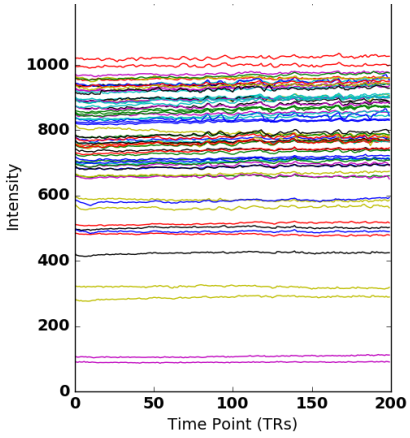
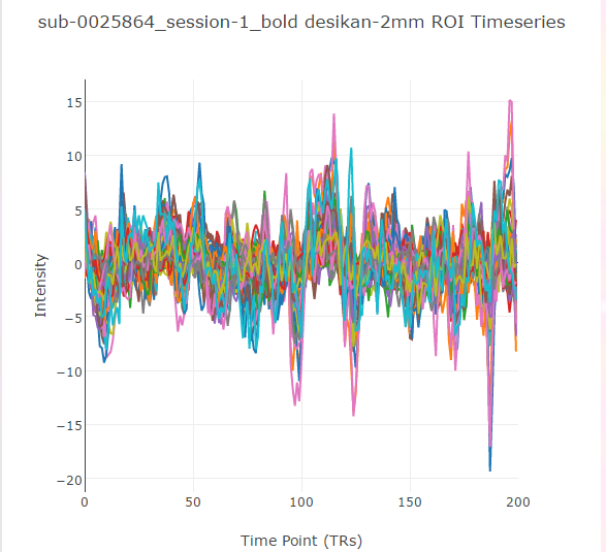


Figure 12: Corrected fMRI timeseries. Note that the linear trend is completely gone, the timeseries are relatively normalized in intensity; other improvements are not immediately obvious from this figure, but it shoudl give an idea of just how improved the timeseries are.

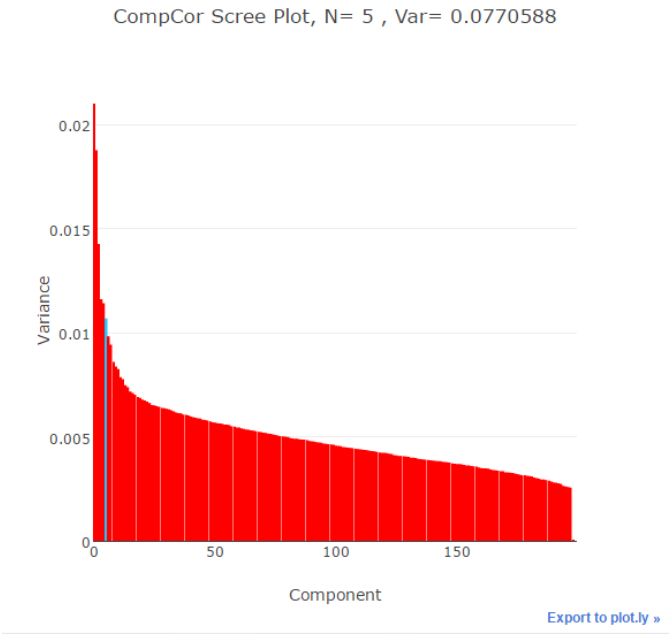


6.2 Scree Plots

Below, we show a scree plot of the variance per component. As you can see, this is from the $n = 5$ pipeline. It would appear as though $n = 5$ captures much of the higher variance components, but the optimal based on the "broken stick" method of choosing the components before the elbow where we see dramatic falloff in the return of successive components is around 10 or so. In future investigations, we will consider algorithms to better select the cutoff number for individuals based on the dropoff of their

singular values; however, that is a completely different Machine Learning problem to investigate further over intercession.

Figure 13: Scree plot taken from one subject of singular values used during nuisance correction of white matter signal.



References

[1] Shangsi Wang, Zhi Yang, Xi-Nian Zuo, Michael Milham, Cameron Craddock, Gregory Kiar, William Gray Roncal, Eric Bridgeford, Carey E Priebe, and Joshua T Vogelstein. Optimal Decisions for Discovery Science via Maximizing Discriminability: Applications in Neuroimaging. *request for preprint.*