

# UC Berkeley

## UC Berkeley Electronic Theses and Dissertations

### Title

High Dimensional Statistical and Computational Methods for Knowledge Discovery and Data Mining in Biomedical Data

### Permalink

<https://escholarship.org/uc/item/2gn5j62f>

### Author

Shi, Funan

### Publication Date

2018

Peer reviewed|Thesis/dissertation

**High Dimensional Statistical and Computational Methods for Knowledge  
Discovery and Data Mining in Biomedical Data**

by

Funan Shi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Haiyan Huang, Chair

Professor Peter J. Bickel

Professor Lewis J. Feldman

Summer 2018

**High Dimensional Statistical and Computational Methods for Knowledge  
Discovery and Data Mining in Biomedical Data**

Copyright 2018  
by  
Funan Shi

## Abstract

High Dimensional Statistical and Computational Methods for Knowledge Discovery and Data Mining in Biomedical Data

by

Funan Shi

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Haiyan Huang, Chair

Biomedical sciences have seen radical growth in recent decades, inspired by a plethora of technological breakthroughs, of which sequencing and imaging are two particular technologies whose advancements have enabled scientists to explore areas that were previously impossible. High-throughput sequencing, for instance, is perhaps one of the most groundbreaking advancements in biology; it allows genetic material (e.g DNA, RNA, proteins) to be identified cheaply and accurately, granting investigators unprecedented insight into the inner workings of the genome—the blueprint of all living organisms. Therefore, high-throughput technology, and in recent years single cell sequencing in particular, has become the cornerstone of genetics research. Sequencing can reveal the genomic location of a gene, but often times the physical locations where a gene is expressed in a cell are also biologically meaningful, and with imaging technologies like florescent tagging and powerful electronic microscopes, this information is now possible to ascertain. Of course, the field of imaging technology is vast, and other areas have also seen tremendous leaps forward. For instance, with the development of CT scans and better PET tracers, researchers now have an *in vivo* view of the metabolic activities in organs, allowing researchers to monitor and study diseases as they progress, thus generating an unprecedented level of understanding of devastating conditions such as Alzheimer’s.

In response to the profusion of quality data, statistical techniques that attempts to analyze these data have also flourished into the field of computational biology and statistical genomics, which has since emerged as an indispensable part of scientific discovery pipeline as well as an important interface between statistics/machine learning and biomedical sciences. In this thesis we examine applications of statistical techniques to three vastly different data sets. In the first work we analyze data from PET brain scans of Alzheimer’s Disease patients and explore how linear mixed effect model offers a powerful and flexible alternative for gauging  $\beta$ -Amyloid accumulation. The data we study in the second work consists of single-cell RNAseq data from mouse embryonic, human embryonic, and human cancer cells, from

which we introduce a biclustering method to simultaneously extract biologically relevant cell clusters and genes that are active in those clusters. In the third work, multiple sources of biological databases consisting of both imaging and sequencing data were leveraged into a machine learning problem, on which random forest is applied to mine organogenesis master regulators.

*To my parents*

This humble work is but a small token of appreciation for all the love and support you have shown me throughout life. But I will never stop give you a hard time for not coming to any of my graduations.

# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Identifying cell subpopulations and their genetic drivers from single cell RNA-Seq data using a biclustering approach</b>	<b>4</b>
2.1 From RNAseq to scRNAseq: A brief overview of whole transcriptome sequencing	5
2.2 Brief review of selected biclustering algorithms	6
2.3 BiSNN-Walk: biclustering using SNN and Walktrap	7
2.4 Results	15
2.5 Discussion and Future Work	26
<b>3 A more powerful and flexible method for measuring Amyloid-<math>\beta</math> accumulation through linear regression</b>	<b>33</b>
3.1 Introduction	33
3.2 Data	36
3.3 Explore the linear relationship between target and reference SUVs	37
3.4 Measures for gauging A $\beta$ accumulation	38
3.5 Compare statistical properties of SUVR and $\Delta$ -measure under assumed parametric generating model	41
3.6 Results	46
3.7 Conclusion	48
<b>4 Data-driven discovery of organogenesis master regulator candidates for <i>D. melanogaster</i> and <i>C. elegans</i></b>	<b>51</b>
4.1 Introduction	51
4.2 Data	52
4.3 Methods: constructing a learning problem	55
4.4 Results	59

4.5 Conclusion . . . . .	62
<b>Bibliography</b>	<b>64</b>
<b>A Supporting material for “Identifying cell subpopulations and their genetic drivers from single cell RNA-Seq data using a biclustering approach”</b>	<b>70</b>
A.1 Construction of SNN Network . . . . .	70
A.2 Overview of irreducible discovery rate . . . . .	72
A.3 Overview of Walktrap Clustering . . . . .	74
A.4 Overview of Adjusted Rand Index . . . . .	77
<b>B Supporting material for “A more powerful and flexible method of measuring Amyloid-<math>\beta</math> accumulation using linear model”</b>	<b>81</b>
B.1 Linear relationship between target vs reference SUV persists across time . .	81
B.2 ADNI linear regression diagnostic plots . . . . .	88
B.3 Using feasible generalized least squares to verify that $\beta$ is constant across time	88
B.4 Deviation details of Proposition 3.5.1 . . . . .	90



# List of Figures

2.1	BiSNN-Walk algorithm flowchart . . . . .	8
2.2	Entropy curves from toy simulation . . . . .	10
2.3	Entropy curves of initial similarity matrices from the three scRNA-Seq data sets . . . . .	12
2.4	First 3 outer loops of BiSNN-Walk on mouse embryo data set. . . . .	15
2.5	BiSNN-Walk clusters compared to ground truths . . . . .	18
2.6	Clustering comparison on mouse data for biclustering algorithms . . . . .	20
2.7	Clustering comparison on human embryo data for biclustering algorithms . . . . .	21
2.8	Clustering comparison on human cancer data for biclustering algorithms . . . . .	22
2.9	Clustering comparison BiSNN-Walk against GiniClust . . . . .	23
2.10	Overlap of top characteristic genes in each cluster . . . . .	25
3.1	Tracer binding schematic in target vs reference ROI . . . . .	35
3.2	Target vs reference ROI SUV . . . . .	38
3.3	Residuals at entry scan vs follow-up . . . . .	39
3.4	Change in target SUV vs that in reference . . . . .	40
3.5	$\Delta T \sim \Delta R$ regression diagnostic plots for BLAZE . . . . .	41
3.6	$T_1 \sim R_1$ regression diagnostic plots for BLAZE . . . . .	43
3.7	QQ plots of reference SUV for ADNI and BLAZE . . . . .	43
3.8	Power curve comparison between $\Delta$ -measure and $\Delta SUVr$ using simulated data . . . . .	49
4.1	Example of in-situ imaging technology . . . . .	54
4.2	Schematic for construction of the predictor matrix . . . . .	57
4.3	Schematic for repeated runs of balanced random forest. . . . .	58
A.1	Positions of cells in gene expression space . . . . .	70
A.2	Shared nearest-neighbor network construction . . . . .	71
B.1	ADNI: $T_1$ vs $R_1$ . . . . .	82
B.2	ADNI: $T_2$ vs $R_2$ . . . . .	83
B.3	ADNI: $T_2 - T_1$ vs $R_2 - R_1$ . . . . .	84
B.4	BLAZE: $T_1$ vs $R_1$ . . . . .	85
B.5	BLAZE: $T_2$ vs $R_2$ . . . . .	86
B.6	BLAZE: $T_2 - T_1$ vs $R_2 - R_1$ . . . . .	87

B.7 ADNI $T_1 \sim R_1$ linear regression diagnostic plots . . . . .	88
--	----

# List of Tables

2.1	Cluster ARIs of BiSNN-Walk clusters using different initial similarity matrices .	11
2.2	Performance comparison between BiSNN-Walk, SNN-Cliq, and GiniClust. . . .	26
2.3	Collection details of mouse embryonic cells . . . . .	27
2.4	Collection details of human embryonic cells . . . . .	28
2.5	Collection details of human cancer cells . . . . .	29
2.6	Summary of BiSNN-Walk output on the three data sets. . . . .	30
2.7	EMAPA Term enrichment of BiSNN-Walk clusters for mouse embryo data . . .	31
2.8	GO Term enrichment of BiSNN-Walk clusters for human cancer data . . . . .	32
3.1	Testing whether $R_1$ and $R_2$ are from the same distribution . . . . .	44
3.2	Estimates of model parameters for BLAZE and ADNI data sets . . . . .	46
3.3	Performance comparison between $\Delta$ -measure and $\Delta SUVr$ using BLAZE data. .	47
3.4	Performance comparison between $\Delta$ -measure and $\Delta SUVr$ using ADNI data. . .	47
4.1	Examples of importance differential for transcription factor groupings. . . . .	59
4.2	Master regulator candidates for D. melanogaster gut. . . . .	60
4.3	Master regulator candidates for D. melanogaster nervous system. . . . .	60
4.4	Top master regulator candidate for C. elegans tissues. . . . .	61
A.1	Example contingency table used to calculate adjusted Rand index . . . . .	78

## Acknowledgments

First I would like to express my sincerest gratitude to Dr. Haiyan Huang for being the best advisor one could ever hope for. My graduate experience was wonderfully enriched from our collaborations and communications as I have learned tremendously both as a scholar and as a person under her mentorship. In particular, this work could not have possible without her patience, guidance, and support.

I would also like to extend my gratitude to the other members of my thesis committee: Dr. Peter Bickel for his circumspect remarks and constructive criticism, and Dr. Lewis J. Feldman for his timely and insightful feedback and much-welcomed encouragement.

I am much indebted to Dr. Thomas Bengtsson and Dr. David Clayton at Genentech for their guidance and advice through an extremely enjoyable and fruitful internship, the result of which became part of this thesis.

I am grateful for my collaboration with Dr. Ben Brown, whose advice and support was invaluable for making this work a reality.

I thank La Shana for perpetually being available and guiding me through an ocean of paperwork.

Besides my advisors and mentors, I'm also grateful for the many students and faculties at Berkeley Statistics department and Lawrence Berkeley National Laboratory with whom I have had the luck and pleasure of meeting and collaborating. These experiences have been invaluable for molding me into an independent thinker and researcher.

Last but not least, I would like to thank my parents Dr. Xiquan Shi and Guofeng Shang and my wife Haiyun Li, who have provided unconditional love and support throughout this journey and life in general.

# Chapter 1

## Introduction

The field of computational biology is the application of statistical and computational techniques to areas of biomedical sciences such as genomics, system biology, and medicine. There has been long history of cross pollination between statistics and biomedical fields; in fact, many of the frequently used statistical tools today were developed to answer biological questions. Karl Pearson, considered by many to be the father of mathematical statistics, developed a suite of commonly used statistical apparatus such as the principal component analysis, method of moments, chi-squared test, correlation, ...etc, to better understand heredity in Darwinian evolution. Sir Ronald Fisher, another luminary of equal standing as Pearson, developed his own statistical toolbox consisting of p-values, maximum likelihood, randomization, and design of experiments, much of which was originally developed for the analysis of agricultural data and application to genetic research. Although the field of computational biology is not new, it has seen rapid advancement in recent decades, becoming an indispensable apparatus of scientific discovery and an important interface between statistics/machine learning and biomedical sciences; and this interface has brought about new problems and challenges that has since enriched the field of statistics itself. Its emergence was propelled by two forces: the rapid development in the biomedical technology that generates ever growing quality, quantity, and variety of data; and commensurate development in computational technology that enabled the building of ever more complex statistical machinery to process and analyze the profusion of data timely and efficiently.

One prominent example is high throughput sequencing. One of the biggest scientific achievements of the 21st century was the Human Genome Project, aiming to chart out the blueprint of human genetics. The first draft of the human genome was estimated to have cost \$0.5-1 billion over the course of 10 years; since then, with the advent of high throughput sequencing, a complete human genome can be sequenced for under \$2000 [63]. The crux of high throughput sequencing is shifting the challenge from labor intensive wet-lab work (Sanger sequencing) to a computational one, where millions of randomly sliced overlapping short reads are pieced together algorithmically in order to reconstruct the genome, much like a computational jigsaw puzzle. As one can imagine, the alignment algorithm is highly

computationally intensive, and only became a reality with a strong computation backdrop. With orders of magnitude of improvements in both time and cost, high throughput sequencing technology has since become the cornerstone of genetic research.

Imaging is another area where the marriage of biotechnology and computation drove tremendous scientific development. Positron emission tomography, commonly known as the PET-scan, is a non-invasive imaging technique that allows users to see the metabolic activity of internal organs. The technique has seen successful application in a wide variety of fields such as oncology, neuroimaging, and cardiology. Due to the non-invasive nature, the technique has contributed immensely to the advancement of Alzheimer’s Disease research. Prior to PET imaging, data points about the AD can only be obtained from biopsied post-mortem brain samples, which means the disease can only be studied at its maturity. With the advent of PET imaging, however, the disease can be studied *in vivo*, allowing *progression* of the disease to be monitored and studied. The PET-scan was so instrumental in many medical and scientific advancements that it was named TIME magazine’s scientific discovery of the year in 2000. The wide adoption of PET technology in biomedical research is not possible without strong computational support. Medical image registration, for instance, is a computationally intensive process in which images from various modalities (e.g CT, PET, MR...etc) are spatially aligned to synergize the insights offered by individual images, which are often complementary. For instance, CT and PET scans are often executed together because CT gives high resolution anatomical details, thus tissue-wise metabolic information extracted from PET+CT scans is generally of higher fidelity than that from PET scans alone. The medical image registration literature is immense [31], but techniques generally involve extracting features from, comparing intensities of, and optimizing over images with millions of pixel; and with the prevalence of 3D images and videos in recent years, new challenges are constantly surfacing that require more complex and intricate algorithms, which in turn rely on robust computing platforms.

In addition to providing computational solutions for biomedical technologies, increased computational power also universalized statistical techniques that were previously deemed too time-consuming. Parallelization, for instance, made it possible to run many computationally intensive algorithms that are currently the backbones of modern statistical repertoire, such as (but not limited to) Monte Carlo methods, graphical models, and linear/dynamic programming. Random forest, for instance, is a prediction technique that involves training hundreds to thousands of simple tree predictors. This architecture allows the method to take full advantage of parallelization and derives multi-order of magnitude improvement in speed. The marriage of statistics and computer sciences (parallelization, optimization, etc) emerged as a new and exciting field of machine learning, which has seen continual widespread in analysis of biomedical data.

Computational biology and statistical genomics is an exciting field that tackles the ever-evolving statistical challenges that comes with the rapid advancements in biomedical sciences.

In this work, we present three case studies analyzing data sets generated by the aforementioned technologies, and use three vastly different statistical methodologies for analysis. Due to the broad range of background dealt in each project, we will defer details to each individual chapters.

## Chapter 2

# Identifying cell subpopulations and their genetic drivers from single cell RNA-Seq data using a biclustering approach

Single-cell RNA-Seq (scRNA-Seq) has attracted much attention recently because it allows unprecedented resolution into cellular activity; the technology, therefore, has been widely applied in studying cell heterogeneity such as the heterogeneity among embryonic cells at varied developmental stages or cells of different cancer types or subtypes. A pertinent question in such analyses is to identify cell subpopulations as well as their associated genetic drivers. Consequently, a multitude of approaches have been developed for clustering or biclustering analysis of scRNA-Seq data. In this paper, we present a fast and simple iterative biclustering approach called “BiSNN-Walk” based on the existing SNN-Cliq algorithm.

In this chapter we introduce a fast, simple, self-correcting iterative biclustering method named “Biclustering using Shared-Nearest-Neighbor and Walktrap” (BiSNN-Walk for short, pronounced “bison walk”). The BiSNN-Walk expands on the idea of clustering on Shared Nearest Neighbor (SNN) network constructed from gene expression matrix proposed in Xu et al’s SNN-Cliq algorithm [84] by adding a gene clustering component to SNN-Cliq’s cell clustering framework. One of BiSNN-Walk’s differentiating features is that it returns a ranked list of clusters, which may serve as an indicator of a cluster’s reliability. Another important feature is that BiSNN-Walk ranks genes in a gene cluster according to their level of affiliation to the associated cell cluster, making the result more biologically interpretable. We also introduce a simple entropy-based measure to guide our initial similarity matrices selection, which serves as a starting point for BiSNN-Walk. In our exploratory analyses, the entropy measure shows promise for gauging the “clusterability” of similarity matrices.

We applied BiSNN-Walk to three public single cell RNA-Seq data sets and found that



the algorithm not only maintained SNN-Cliq’s clustering capability, but also produced biologically interpretable results by establishing genes that are characteristic to those clusters. This chapter follows Shi and Huang [69].

The chapter is organized as follows: Section 2.1 gives a brief overview of RNAseq and single-cell RNAseq. An overview of selected biclustering algorithms are given in Section 2.2. Section 2.3 will outline BiSNN-Walk and detail the key steps. Section 2.4 will describe three scRNA-Seq datasets used for validation, compare our cell clusters against SNN-Cliq, offer visual comparisons against selected biclustering algorithms, and finally evaluate the gene clusters via gene overlap and ontological term-enrichment analysis.

## 2.1 From RNAseq to scRNAseq: A brief overview of whole transcriptome sequencing

RNAseq is a technique where next-generating sequencing [62] is applied to sequencing the transcriptome, thus is also called whole transcriptome shotgun sequencing. Since the transcriptome contains RNA transcripts of a cell that are then translated into proteins that carry out various tasks around the cell, RNAseq has become a cornerstone of genomic research by allowing us to see what transcripts are floating around in a cell. Because the starting material for RNAseq needs to go through cleaning and isolation, the final input material into the sequencer may not be enough to produce reliable signal; therefore RNAseq is typically performed with bulk starting material to ensure reliable return of signal; however, this process means the natural variation between cells are lost in the process. Moreover, if the cell type is rare or hard to obtain, such as circulating cancer cells or embryonic stem cells, one may not even be able to obtain enough for starting material.

To combat this limitation, a new RNAseq extension called single-cell RNAseq has been gaining traction in recent years that enables sequencing with minute amount of starting material. The main advancement lies in transcript amplification, which allows the minute amount of RNA material to be amplified into a viable amount for RNAseq. Multiple displacement amplification is the most widely used technique, but other variants and approaches have also been developed to tackle this problem, e.g Smart-seq [60], Smart-Seq2 [55]. In light of the rise of the technology, an increasing number of experiments have been conducted and yielded excellent results [14, 60, 85, 83]. The unprecedented resolution into cell states provides hope for a better understanding of cell function and dysfunction [18], for which scRNA-Seq was bestowed the honor of “Method of the Year” by Nature in 2013 [51].

Being a high-throughput technique, scRNA-Seq data poses interesting statistical problems. One such problem is to cluster cells into biological categories, e.g distinct develop-

mental stages, cell types...etc, to discover cell-based biologics. Compared to other clustering tasks, algorithms developed for scRNA-Seq data need to take into account the increased variation in data that comes with sequencing individual heterogeneous cells (e.g. [84, 5]). To improve upon existing algorithms, a natural extension is to simultaneously identify biologically important genes for each cell category while performing clustering. Under this setting of bi-clustering, it is reasonable to expect that the identified signature genes would not only aid clustering the cells by denoising the data, but also help answer questions such as “what genes are heavily recruited in the 2-cell stage of mouse embryonic development?”

## 2.2 Brief review of selected biclustering algorithms

Since a gene may be involved in multiple cell conditions, the biclustering problem we consider allows for overlapping gene (row) clusters but non-overlapping cells (columns). For instance, it’s reasonable to assume similar genes would drive 2-cell embryonic and 4-cell embryonic development due to their chronological proximity. Even though the field of bi-clustering is vast, there are few methods that specializes in the specific bicluster structure we consider. Most existing biclustering algorithms such as Block partitioning [27] are not suitable because they do not allow overlapping gene clusters. More flexible models such as the Cheng & Church model [12], which considers a bicluster as a submatrix with consistent column and/or row effects, are often too computationally expensive for the problems we consider. Coupled Two-way Clustering [24] is a popular method that sequentially divides an initial cluster until a stable child cluster is found. However, since the method cannot self-correct, the quality of the child clusters may be entirely dictated by the quality of the initial cluster.

We select a few reference methods that are suitable for our problem type and are characteristics of different approaches to biclustering. They are Plaid [41], Cheng & Church [12], Xmotifs [79], and BiMax [47]. We also applied a recently published clustering algorithm, GiniClust [34], designed to handle scRNA-Seq data. Coupled Two-way Clustering is not considered since there are no viable implementation of the algorithm, and the authors could not be reached to obtain one.

*Plaid, Cheng & Church.* As one of the landmark papers in the field, Plaid and CC are frequently used as benchmarks. Both Plaid and CC assumes that gene expression can be expressed in an additive fashion of the form  $\mu + a_i + b_j$ , where  $\mu$  is the background constant,  $a_i/b_j$  are row/column specific constants, respectively. The  $\mu$ ,  $a_i$ , and  $b_j$  are treated as parameters in the Plaid model to be fitted, whereas they are set as row, column and overall means, i.e constants, in CC. The chosen bicluster would have expression values that fall most consistent around  $a_i$ ,  $b_j$ , and  $\mu$ . For Plaid model, after a bicluster is found, its values are then subtracted, and the residual expression matrix is used to find the subsequent biclusters.

For CC, after a bicluster is found, random gene expression values are used to replace that of the true bicluster, and the resulting scrambled “gene expression matrix” is then used for subsequent runs.

*BiMax.* Devised as a benchmark algorithm, BiMax is the simplest of the benchmarks. It operates on binary matrices, and uses a divide and conquer approach to find all maximal completely bipartite graphs (maximal submatrices containing all 1's).

*Xmotifs.* For any random cell cluster  $C$  we define a gene's expression on the cells in  $C$  as interesting or not interesting, which is called a gene's *state* on  $C$ . We call a gene-cell bicluster  $(G, C)$  an Xmotif if for every cell in  $C$ , all of the genes in  $G$  are in the same state. Further more, an Xmotif is maximal if every genes not in the Xmotif has less than  $\beta$  (some user defined percentage) states in common with genes in  $G$ . The algorithm randomly generates a user-defined number of seed-Xmotifs and attempt to grow them into maximal Xmotifs.

*GiniClust.* Modified Gini index is used to isolate genes of interest. The submatrix with rows being the selected genes and columns being the cells are then passed to the clustering algorithm DBSCAN to obtain cell clusters. Therefore the algorithm will return cell clusters, but only one cluster of genes, so it's in fact more an algorithm for clustering than biclustering.

## 2.3 BiSNN-Walk: biclustering using SNN and Walktrap

Figure 2.1 details the flow of BiSNN-Walk. In essence, the algorithm iterates between an inner loop and an outer loop.

The inner loop cycles through three main steps: cell clustering “SNN-Walktrap”, gene finding, and expression matrix updating (Figure 2.1, steps ①, ②, ③, respectively). We pass an initial similarity matrix into SNN-Walktrap to obtain a candidate cell cluster, which is used to find characteristic genes. Step ③ then produces a gene expression matrix containing only those characteristic genes. The reduced expression matrix is in turn used by SNN-Walktrap to obtain a new cell cluster. The process then iterates until either the cell cluster stabilizes or a pre-set iteration limit is reached. The inner loop will have produced one bicluster upon termination.

The process then goes to the outer loop, where the cell cluster found by the inner loop is removed from the input matrices. The updated matrices are subsequently fed into the inner loop to obtain the next stable cluster. The process continues until stopping criteria (described in Section 2.3.5) is met. The following sections will detail several major steps.

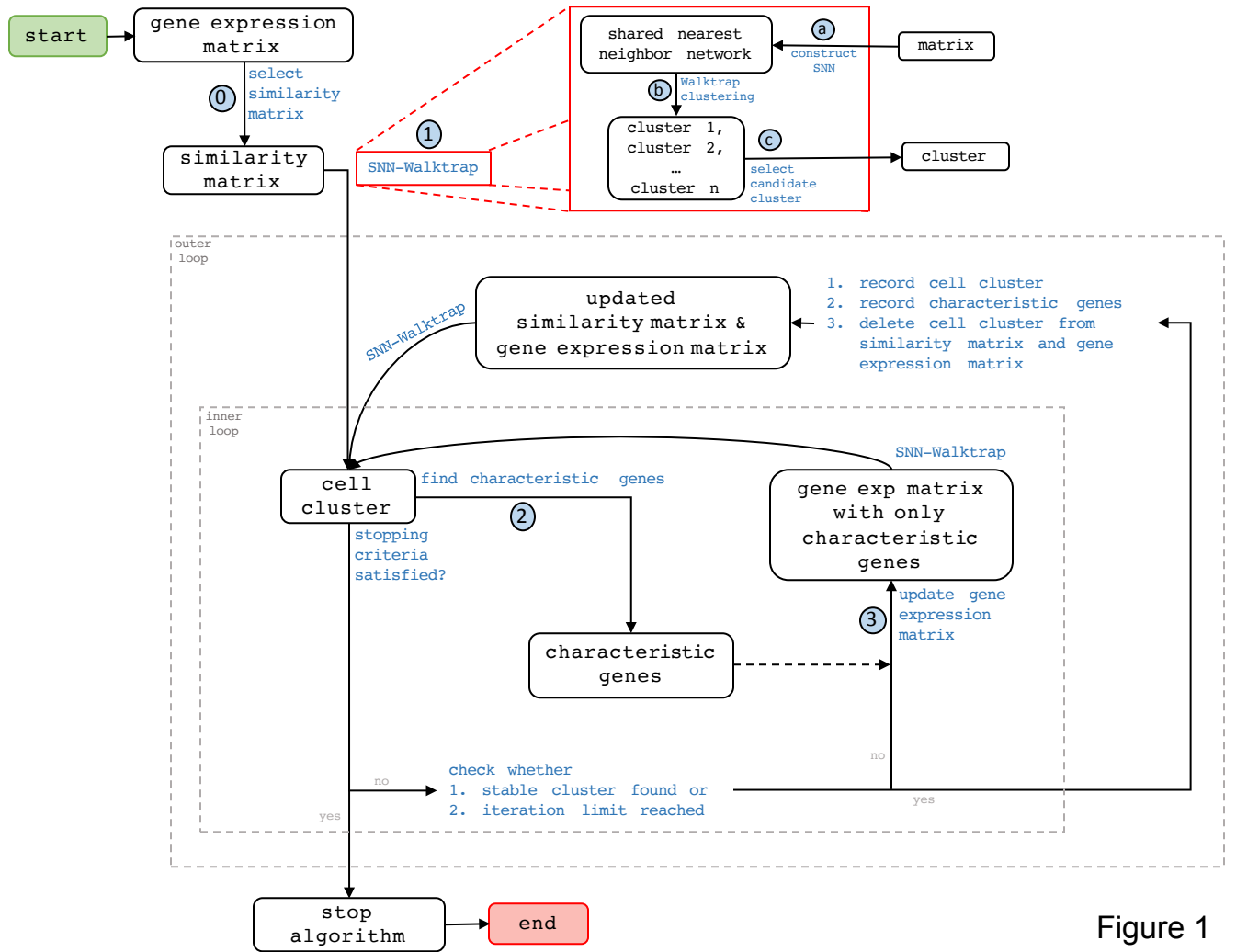


Figure 2.1: *BiSNN-Walk* algorithm flowchart. Inputs and outputs are in rounded boxes, functions are in blue texts. The function *SNN-Walktrap* is a bit complicated, and its details are laid out in the red box. Steps with circled numbers are crucial steps that will be repeatedly referenced.

### 2.3.1 Step 0: Selecting a Similarity Matrix

The first step is to choose a similarity matrix to be used to obtain the initial cell cluster (Figure 2.1, step ①). We consider an initial similarity matrix with high contrast to be ideal, i.e the correlation between cells of the same type should concentrate tightly near 1, while that between different types should concentrate around near 0. For illustration, we will examine four types of similarity matrices: Euclidean distance, Spearman correlation, Pearson correlation, and the irreproducible discovery rate (IDR) matrix [45]. IDR measure the dissimilarity between two cell’s active genes, e.g two cells that have a similar set of active genes and similar expression profiles across those genes will generate a small IDR value, and vice versa. IDR stands apart from existing cell similarity measures in that it does not resort to using all of genes or a pre-selected set of ”relevant genes”. The former is not desirable since a large portion of a cell’s genetic profile consists of housekeeping and non-active genes, which may lower a method’s power to identify the relationship between two cells. The latter is not ideal either since the threshold for “active” genes will vary across cells. The use of IDR bypasses these difficulties. Our results (Appendix G, Table 2) demonstrates that IDR matrix consistently provides high quality final clustering. For continuity’s sake, a brief overview of IDR is provided in Section A.2.

To choose which similarity matrix to use, we propose a simple entropy-based measure.

**Definition** (Entropy of a similarity matrix). Let  $x$  be the vector obtained from the upper triangle of the similarity matrix, we put the values of  $x$  into  $m$  equal sized bins, akin to what is done for histograms. Let  $\mathbf{p} = [p_1, \dots, p_m]$  denote the proportion of values that fall into each bin<sup>1</sup>. The entropy for an  $m$ -bin configuration is calculated as

$$Entropy(\mathbf{p}, m) = \sum_{i=1}^m p_i \log p_i$$

This entropy can be interpreted as the amount of noise in a similarity matrix, thus a similarity matrix with large amount of clustering information should have low entropy. Because entropy calculation depends on  $m$ , it is illustrative to compare entropy at several  $m$ ’s varying in an appropriate range. We found that the entropy measure performs as expected in simulation and provides good initial similarity matrix for real data. More detailed exploration of the behaviors of the various initial matrices are organized in Section 2.3.1.1.

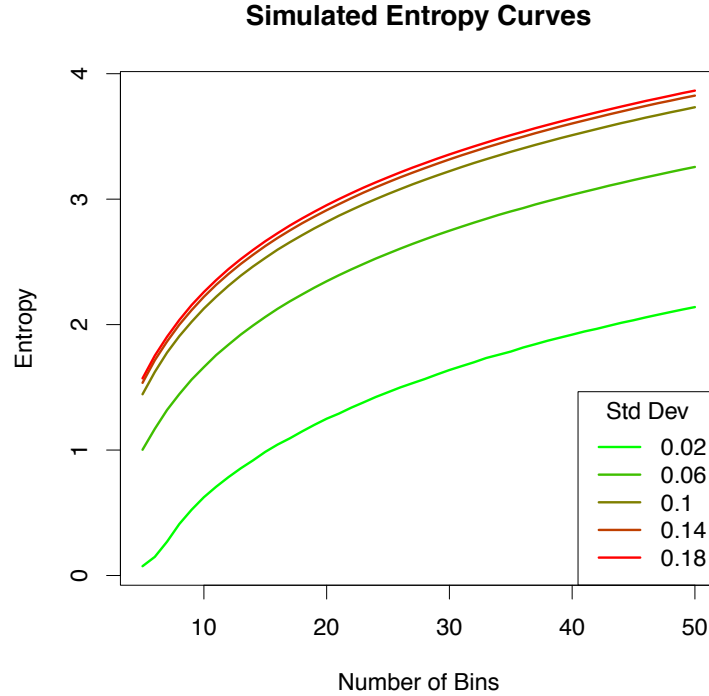
#### 2.3.1.1 Exploration of Entropy Measure via Simulation and Real Data

To explore the behavior of the entropy curves, we did a toy simulation where we generated 10 clusters, with size of each cluster uniformly chosen between 4 and 20. We assume the similarity (correlation) within a cluster is  $N(0.7, \sigma^2)$ , and similarity between cluster is

---

<sup>1</sup>Ignore bins with 0 counts

Figure 2.2: *Entropy curves from toy simulation. We generate a similarity matrix for 10 clusters where the within-cluster similarity is randomly drawn from  $\mathcal{N}(0.7, \sigma^2)$ , and out-of-cluster similarity is randomly drawn from  $\mathcal{N}(0.3, \sigma^2)$ . Each curve represents a different value of  $\sigma$ , varying from 0.02 to 0.18. As one can see, entropy increase monotonically with  $\sigma$  across all numbers of bins.*



$\mathcal{N}(0.3, \sigma^2)$ , where  $\sigma$  varies from 0.02 to 0.18. As one can see from the resulting curves in Figure 2.2, the curves line up according to  $\sigma$ , with the curve corresponding to  $\sigma = 0.18$  having the highest entropy while the curve corresponding to  $\sigma = 0.02$  having the lowest. When we applied this measure to real data sets, we found it can either help yield the best result among different choices of initial similarity matrices or generate results that are quite comparable to the best result available (See Table 2.1 for more details).

In real data set, however, using entropy to choose initial matrix does not necessarily produce the highest quality final clusters. For instance, for Human Embryo data set, we should choose Spearman correlation as our similarity matrix (Figure 2.3) according to entropy measure, but Table 2.1 shows us that Euclidean matrix produced the best final clustering result. This, however, should not detract from the idea of using entropy as a measure of clustering potential. First, the entropy based measure is used to select a good starting point for the algorithm, but a good starting point does not guarantee best result. Second, even though entropy-selected matrix does not yield the best result, they are often quite comparable to the best result available (Table 2.1). Therefore, we believe entropy-based measure of clustering-potential is a promising direction, and its theoretical properties will be explored in depth in

a future paper.

Table 2.1: Comparison between ARIs of final clusters using BiSNN-Walk with four different initial similarity matrices

	Mouse Embryo	Human Embryo	Human Cancer
IDR	<b>0.472 (310/317)</b>	0.600 (124/124)	<b>0.883 (83/86)</b>
Euclidean	0.447 (314/317)	0.798 (124/124)	0.873 (82/86)
Pearson	0.481 (297/317)	0.677 (124/124)	0.834 (70/86)
Spearman	0.467 (307/317)	<b>0.776 (124/124)</b>	0.880 (86/86)

Number in parentheses is (number of cells clustered / total number of cells)

Bold font indicates the matrix with lowest entropy

### 2.3.2 Step 1: Cell Clustering by SNN-Walktrap

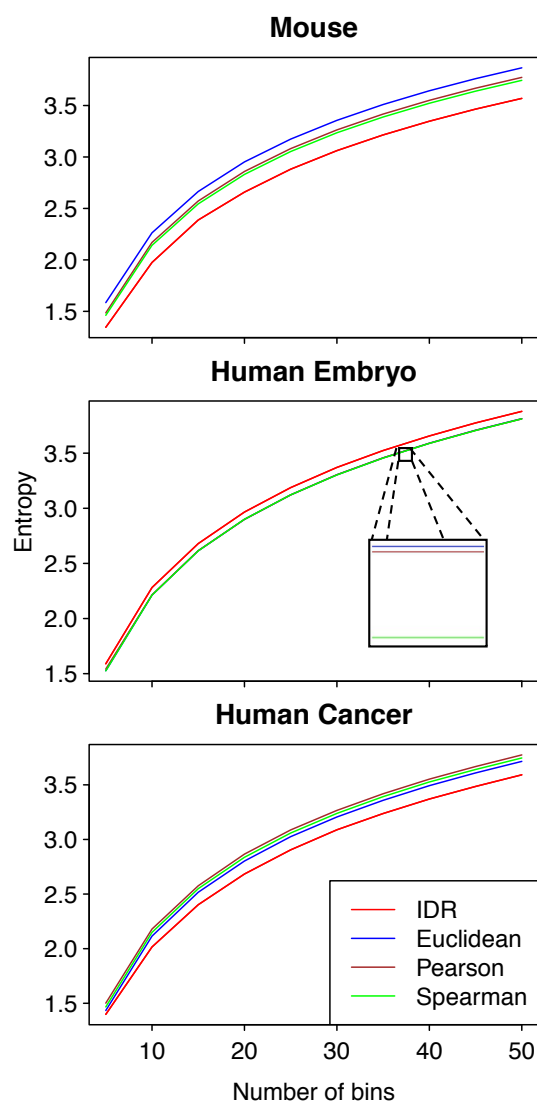
SNN-Walktrap is the main function used in the inner loop, and is the workhorse of our algorithm. SNN-Walktrap (Figure 2.1, step ①) takes a matrix input (e.g gene expression or similarity matrix) and return a cell cluster. The process can be broken down into three steps (Figure 2.1, red box):

**Step ① : Construct SNN Network** Construction of shared-nearest-neighbor is the corner stone of SNN-Cliq [84], the method that BiSNN-Walk expands on. SNN network defines the notion of “distance” between two nodes within the context of a local neighborhood as opposed to a distance quantified by an global measure (e.g Euclidean distance). This localization is desirable for high dimensional data, where the high dimensionality renders global measures like Euclidean norm less useful as a proxy for distance [1]. Please find an overview of the construction of a SNN network and the rationale behind its use in Section A.1.

**Step ② : Walktrap Clustering** After a network is constructed using SNN, we use Walktrap [57] to perform cell clustering. The Walktrap algorithm is an agglomerative hierarchical clustering scheme akin to a complete-linkage hierarchical clustering. The distance between node  $i$  and node  $j$  is related to the difference in the behaviors of two random walks starting at the two nodes. A very important feature of Walktrap is that one does not need specify a priori the number of clusters. Cutting threshold of the tree is set automatically and is related to the distance measure. The intuitiveness of the cutting threshold was a major reason for choosing Walktrap as our clustering algorithm. Please refer to Section A.3 for an overview of the algorithm.

**Step ③ : Select Candidate Cluster** Walktrap, being a clustering algorithm, will identify several cell clusters; our purpose here, however, is to find the best one. To define “best”, we use a heuristic involving three common clustering metrics: conductance, transitivity, and the Jaccard score.

Figure 2.3: Entropy curves of initial similarity matrices from the three scRNA-Seq data sets





**Definition** (Conductance and Transitivity). Let  $G(V, E)$  denote a network, with  $V = \{v_i : i = 1, \dots, n\}$  denoting a set of nodes and  $E = \{e_{i,j} \in \mathbb{R}^+ : i, j = 1, \dots, n\}$  denote a set of edges. Let  $C$  denote a cluster, and let  $S(C) = \{e_{i,j} : v_i \in C, v_j \notin C\}$  be the edges that are connected to outside the cluster, and  $I(C) = \{e_{i,j} : v_i, v_j \in C\}$  be the edges that are connected within the cluster, then conductance is defined as

$$\begin{aligned} \text{conductance}(C) &= \frac{|S(C)|}{\min\{|I(C)|, |I(V \setminus C)|\}} \\ &= \frac{\# \text{ of edges connected to other clusters}}{\min\{\# \text{ of edges in the cluster}, \# \text{ of edges outside the cluster}\}} \end{aligned}$$

Transitivity, or clustering coefficient, is easier described in words

$$\begin{aligned} \text{transitivity}(C) &= \frac{3 \times \# \text{ of triangles in the cluster}}{\# \text{ of connected triplet of vertices, or V shapes}} \\ &= \frac{\# \text{ of triangles in the cluster}}{\# \text{ of total possible triangles if all nodes are connected}} \end{aligned}$$

Conductance is bounded below by 0, where zero-conductance implies cluster  $C$  is isolated from rest of the network, i.e lower conductivity indicates a better isolated cluster. Note if the cluster is the entire node set,  $C = V$ , then we force  $\text{conductance}(C) = \text{conductance}(V) = 0$ .

Transitivity is bounded between  $[0,1]$ , where 0 means all members of the cluster are isolated points, and 1 indicates a perfect clique.

We may encounter two situations here:

1. When the inner loop is first called (at the very beginning of the algorithm or by the outer loop), we need to initialize a candidate cluster. We select a single candidate from the Walktrap clusters according to two well-known network clustering measures: conductance and transitivity. Conductance measures how well separated a cluster is from rest of the network, and transitivity measures the connectedness of nodes within a cluster. To select a single one from the Walktrap clusters, we first rank them with respect to conductivity, and break ties using transitivity.
2. On subsequent iterations of the inner loop, when there already exists a candidate cluster, we want to improve upon the existing one. We calculate the Jaccard score of each Walktrap cluster with the candidate cluster as a measure of agreement, and select the Walktrap cluster of the highest agreement with the candidate cluster as the new candidate. However, if all Walktrap clusters that overlaps with the original cluster are of sizes 2 or less, then we will stop improving upon the old candidate cluster and choose a new initial cluster from the Walktrap clusters using conductance and transitivity as described previously. The rationale behind this heuristic is that Walktrap will sometimes return clusters of the same transitivity and conductance (e.g

isolated perfect cliques), so the Jaccard measure serves as another tie breaker as well as ensuring a sense of continuity among the iterations' candidate clusters.

### 2.3.3 Step 2: Finding Characteristic Genes

Let  $C$  denote a cluster, and  $Q$  denote the quantile matrix, i.e  $Q$  is a  $g \times n$  matrix such that  $Q_{ij}$  = quantile of gene  $i$ 's expression level for cell  $j$ . Quantizing the raw expression level is a form of normalization that makes the expression across cells more comparable. Define the contrast of gene  $i$  on cluster  $C$  as

$$z_{i,C} = \text{median}(Q_{ij} : j \in C) - 75^{\text{th}}\text{Quantile}(Q_{ij} : j \notin C) \quad (2.1)$$

We call gene  $i$  “characteristic to cluster  $C$ ” if  $z_{i,C} > 0$ . In other words, the characteristic genes are those that are generally more highly expressed in  $C$  than the rest of the cells. The characteristic genes will be ranked according to their contrast—genes with higher contrast are more representative of the cluster. We could, of course, replace  $75^{\text{th}}\text{Quantile}$  with Median and the Max in equation (2.1), but our concern is that the median would result in too liberal a list for genes to be called cluster-specific “characteristic genes”, while the max would return too conservative a list and would thus remove potentially useful clustering information;  $75^{\text{th}}\text{Quantile}$ , therefore, was chosen as compromise.

### 2.3.4 Step 3: Using Characteristic Genes for Subsequent Analysis

As noted in the flowchart (Main Paper: Figure 1, ③), the selected genes will be used to subset the gene expression matrix, which will be fed into the SNN-Walktrap procedure. The rationale behind using only the characteristic genes as our next input is to rid of impurities in the cluster. Assuming our initial cluster mostly contains cells of one state, then it's reasonable to believe that the characteristic genes associated with this cluster will most likely be most relevant to that state. Thus we will see these cells forming a tighter group in a SNN network constructed using only the characteristic genes, thus removing cells of a foreign state. Figure 2.4 demonstrates this “purification” step at work.

### 2.3.5 Stopping Criteria

The algorithm stops if all Walktrap clusters obtained from Walktrap clustering are of size 2 or less or all candidate have zero transitivity, when further clustering is meaningless.

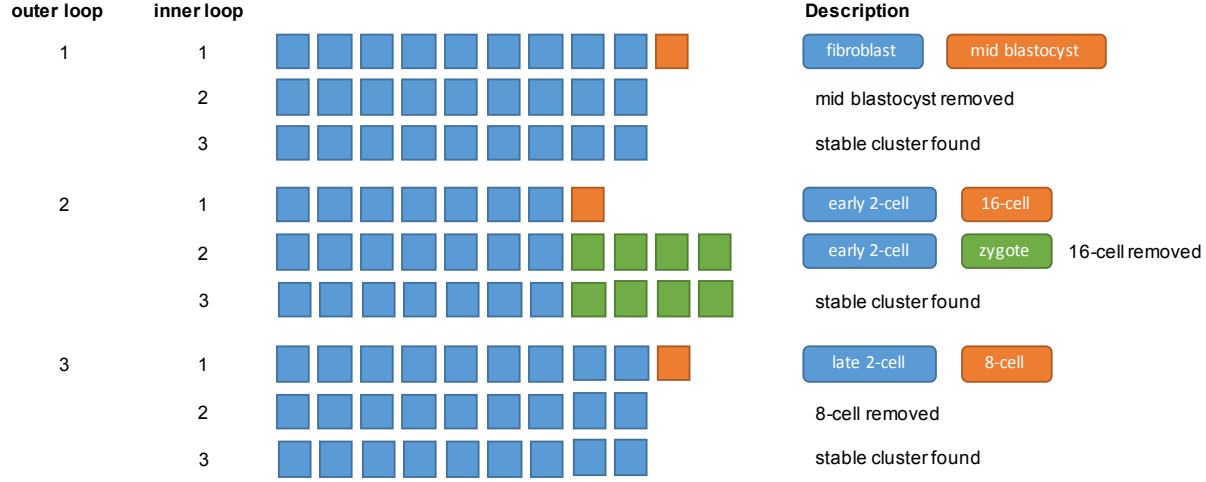


Figure 2

Figure 2.4: *First 3 outer loops of BiSNN-Walk on mouse embryo data set. Outer loop 1 has three inner loops. On the first inner loop, our initial cluster contains nine fibroblast cells and one mid-blastocyst cell. Notice that on the second inner loop, the mid-blastocyst is removed from the cell, and we obtain the cleaned stable cluster on our third inner loop. The second outer loop also contains three inner loops. On the first inner loop, we obtain our initial cluster of seven early 2-cell-stage cells and one 16-cell-stage cell. On the second inner loop, the 16-cell-stage cell was removed, and the zygotes, who are much closer in developmental to early 2-cell stage, are added. On the third inner loop, we obtain the cleaned stable cluster. A similar self-correcting behavior can be seen in the third outer loop.*

## 2.4 Results

### 2.4.1 Validation Data

We use three public datasets to evaluate our algorithm: Mouse Embryonic Cells [14], Human Embryonic Cells [85], and Human Cancer/Somatic Cells [60]. To ensure a level of uniformity of the gene expression, we ran RNA short-reads from each experiment through the standard ENCODE pipeline [19] using STAR for alignment [16] and RSEM for calling differential expression [44]. Gene expressions are normalized using transcript per million (TPM). The three data sets will be referred hereafter as “mouse”, “human embryo”, and “human cancer”, respectively.

The run time of BiSNN-Walk is  $O(krnm^2)$ , where  $k$  = number of clusters or number of outer loop iterations,  $n$  = number of genes,  $m$  = number of cells and  $r$  = number of inner loop iterations. Since  $m$  and  $k \propto m$  are relatively small compared to  $n$ ,  $n$  will dominate the runtime.  $r$  is reflective of the quality of the data—cleaner data will require less iterations. The minimum number of  $r$  is 2: one round to obtain an initial cluster, and another round to verify that it’s stable. Table 2.6 details BiSNN-Walk’s outputs on the evaluation data sets.

*Mouse Embryonic Cells.* The size of the gene expression matrix is  $41,128 \text{ genes} \times 317 \text{ cells}$ . All mouse embryonic cells are crossed between CAST female mated to C57 male cell lines. Embryonic cells were collected during 10 developmental stages from zygote to blastocysts. Somatic cells (liver and fibroblast) are also collected. For consistency’s sake, somatic cells are obtained from either C57 x CAST or CAST x C57 offsprings. See Main Paper: Table 2 for details. Cells are sequence using either Smart-seq or Smart-seq2 technology [14]. This data set will be here on referred to as “mouse”.

*Human Embryonic Cells.* The size of the gene expression matrix is  $60,483 \text{ genes} \times 124 \text{ cells}$ . In this work, Yan et al [85] investigates the genetic markers involved in the derivation of human embryonic stem cells (hESC) by examining the development of human embryo through its developmental stages. 124 embryonic cells were obtained from in-vitro fertilization patients. Patients are aged controlled to be within 25-35 years old with mean age of 30. The cells were categorized into into 8 categories, details show in Main Paper: Table 3. The sequencing technology introduced and used in this study is called “single cell RNA-Seq”. This dataset will here on be referred to as “human embryo”.

*Human Cancer Cells* The size of the gene expression matrix is  $60,483 \text{ genes} \times 86 \text{ cells}$ . In this work the sequencing technique “Smart-seq” was introduced by Ramskold et al. [60] and applied to various low frequency cancer cells such as circulating tumor cells or somatic cells that are difficult to obtain in mass quantities, such as brain cells. A total of 86 human cells are reported by the study<sup>2</sup>, details about cell species are listed in Main Paper: Table 4. Though the study contains both cancer cells and various somatic cells, we will dub this data set “human cancer” for short.

## 2.4.2 Cell Clustering Results

### 2.4.2.1 Performance Comparison vs. SNN-Cliq

Adjusted Rand Index, a recommended metric to quantify agreement between clusters in [65, 49], was used to compare our clustering results against SNN-Cliq’s. Please refer to Appendix E for an overview of ARI. Direct comparison between the two clustering algorithms is not straightforward. First, SNN-Cliq requires the neighborhood parameter  $k$ , and there is little guidance as to how to choose this parameter; we therefore obtained the SNN-Cliq clusters by varying  $k$  from 4 to 12, and considered the clustering result with the highest ARI; in other words, we purposely gave an advantage to SNN-Cliq’s clustering result. In addition, neither BiSNN-Walk nor SNN-Cliq clustered all cells, therefore we also used the number of clustered cells to gauge algorithm performance, with more cells clustered being

---

<sup>2</sup>The study also contained 2 white blood cells, but the sequencing quality was extremely poor, and is confirmed by the author to be unusable through email correspondence. The study also contained reports on mouse cells, but are not used in our data because they cannot be compared directly to human cells.

more preferable.

From the results shown in Table 2.2, BiSNN-Walk is comparable to SNN-Cliq in terms of cell-clustering quality. For mouse data, ARI for SNN-Cliq is higher, but it only clustered about half of the cells. A major difficulty with this data set is distinguishing the three blastocyst stages. SNN-Cliq refused to cluster this stage almost entirely at optimal  $k$  parameter, which is why only 177 out of the 317 cells were clustered. In fact, if we force SNN-Cliq to cluster a similar number of cells (304/317) as BiSNN-Walk, SNN-Cliq’s ARI drops down to 0.465, slightly lower than our result. For human cancer, SNN-Cliq has much lower ARI even though the number of cells clustered are comparable. For human embryo, BiSNN-Walk has a slightly lower ARI while the number of clustered cells are the same. Taking a closer look at the results, we found that BiSNN-Walk was not able to separate *zygote*, *oocyte*, and *2-cell-stage* cells, whereas SNN-Cliq could. This problem does not appear if we had used Euclidean distance as initial similarity matrix; in fact, in that case we actually have a slightly higher ARI than SNN-Cliq (0.798 vs 0.796). This is yet another motivation for exploring the theoretical properties of our entropy-based measure so we can select a more appropriate starting point.

Figure 2.5 shows heatmaps of BiSNN-Walk clusters against ground truths. Both ground truth and BiSNN-Walk clusters are roughly ordered chronologically, and the visible diagonal block structure suggests strong concordance. As mentioned previously, it is difficult to separate 8- and 16-cell cells as well as the early-, mid-, and late-stage blastocysts; however, the diagonal structure of the heatmap indicates that developmental stages that are chronologically close are clustered together. Human embryo cell clusters are also ordered according to developmental stages. Similar to mouse data, the diagonal pattern is clearly visible, indicating developmental stages are clustered by chronological proximity. Human cancer results are not ordered in any particular order, but as indicated by the diagonal structure and the high ARI score, most ground truth cell types are found perfectly. LNCaP cell (prostate cancer cell-line cells) and LNCaP-HTC cells (prostate cancer cells isolated by EPCAM markers in petridish) could not be separated due to their close resemblance. SKMEL5 and UACC are two melanoma cell lines, and could not be separated due to their similarity. The PC3 bladder cancer cell line were not clustered because they were among the last four cells to be clustered, causing Walktrap to returns two clusters of size 2, thus triggering the stopping condition.

#### 2.4.2.2 Performance Comparison vs. Selected Algorithms

BiSNN-Walk is compared with GiniClust [34], a recently published clustering algorithm specifically designed to handle scRNA-Seq data, and four general purpose biclustering algorithms: Plaid [41], Cheng & Church [12], Xmotifs [79], and BiMax [47]. Please refer to Appendix J for brief overviews of the algorithms.

Figure 2.5: *BiSNN-Walk clusters compared to ground truths. x-axis are the BiSNN-Walk clusters, and y-axis ground truth. The value in each grid represents the percentage of ground truth cluster that is in the BiSNN-Walk cluster. For example, in Mouse data, the BiSNN-Walk cluster “Zy & 2e” contains all zygote and early 2-cell stage cells, thus the values in grids (“Zy & 2e”, “zy”) and (“Zy & 2e”, “2.e”) are both 1. The distinct diagonal pattern for all three datasets indicate that the ground truth was well-recovered by BiSNN-Walk clusters.*

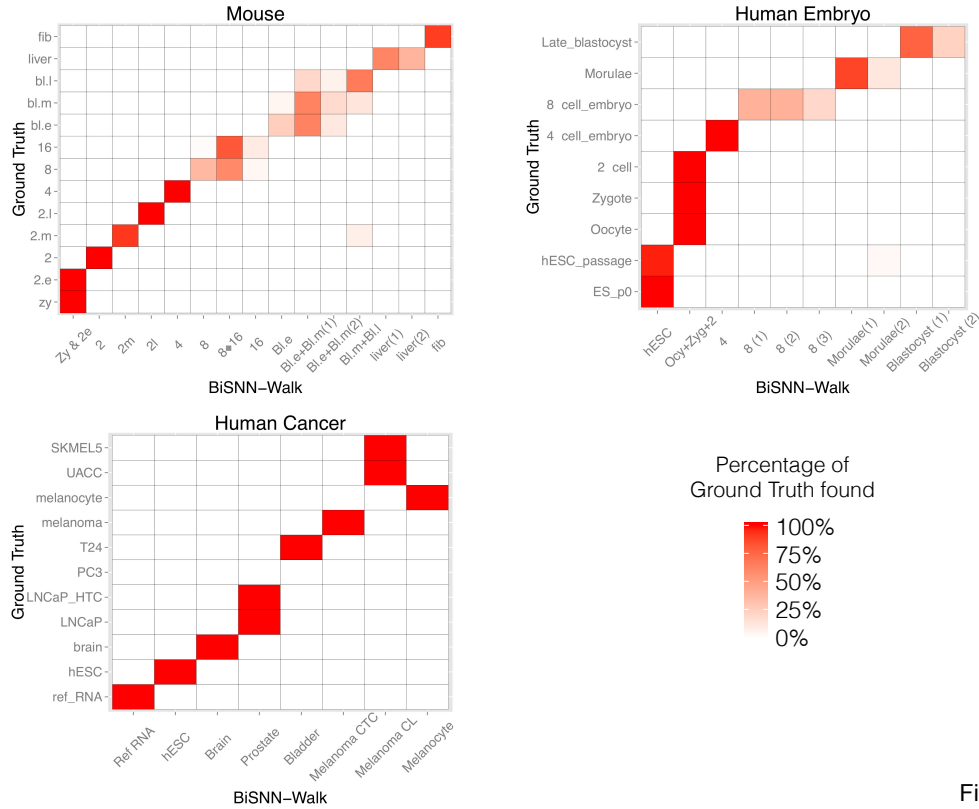


Figure 3

To train each algorithm, we first find impactful tuning parameter(s) and explore a range of values where such parameters returned reasonable answers. We then pick candidate values from that range and perform an exhaustive search to choose the clustering most concordant with the ground truth (measured by Adjusted Rand Index). Again, we purposely gave an advantage to these methods for their parameter selections.

GiniClust returns non-overlapping cell clusters, so we use ARI to measure its cell clustering performance. As the results in Table 2.2 shows, BiSNN-Walk’s clustering results surpass that of GiniClust’s on all three data sets, as measured by ARI. GiniClust’s performance on the developmental data sets, i.e mouse and human embryo, was unspectacular. It was able to cluster together cells who are roughly close in developmental stages, but was not able to find

the finer stages. This may be because GiniClust was designed to isolate small tight-knit rare cell types rather than general purpose biclustering. GiniClust’s human cancer clusters were quite decent, though it only clustered 80% of the cells. Please find more detailed discussion on GiniClust results in Figure 2.9.

Because the other clustering algorithms allow overlapping cell clusters, ARI is not a suitable measure, thus we will visually compare their results to BiSNN-Walk’s. Figure 2.6 shows the cell clustering performance of each algorithm for the mouse data set. Among the four algorithms, only Plaid clusters showed a reasonable diagonal structure, indicating decent alignment with ground truth. A closer examination shows that the Plaid was able to cluster related cell stages together, but did not have the specificity to obtain as fine a resolution as BiSNN-Walk clusters. Cell clustering results for other organism show a similar theme, plots are shown in Appendix J.

The failure of the four biclustering algorithms stems from requiring inputs, i.e a gene exhibits high expression and one exhibiting zero expression is equally informative. For RNAseq data, for a given cell the majority of the genes will exhibit zero expression, the biclustering algorithms in consideration thus were not designed to process this kind of input. Another failure point for these algorithm is the fact that they all consider both overlapping cell and gene clusters, though this formulation is more general, it also works to the disadvantage to the algorithms because they cannot take advantage of the simpler cluster structure of the RNAseq data. We’d be remiss if we did not mention that the algorithm may need expert tuning to achieve maximum performance.

### 2.4.3 Gene Clustering Results

One of the main features of our method is that it simultaneously clusters both cells and genes. We argue that our gene clusters indeed make sense using two methods of evaluation: gene overlap analysis and biological term enrichment analysis.

In gene overlap analysis we examine the overlap of top 100 characteristic genes of each cluster. Clusters who are more biologically similar should share more characteristic genes. For instance, mid-2-cell and late-2-cell stages should share more genetic drivers than, say, mid 2-cell and blastocyst stage. In enrichment analysis we enrich the top 100 characteristic genes of each cluster and see whether the enriched terms makes sense in the context of the cluster. For instance, for a cluster that contains mostly of brain cells, its characteristic genes should return neuron-related enriched terms. In other words, in gene overlap analysis, we check whether gene clusters make sense relative to each other, and in enrichment analysis, we verify whether the gene clusters are representative of their associated cell cluster.

Figure 2.6: Cell clusters found by biclustering algorithm compared to ground truth for the mouse dataset.  $x$ -axis are the cell clusters found by indicated algorithm, ordered roughly by developmental stage.  $y$ -axis is ground truth ordered by developmental stage. The value in each grid represents the percentage of ground truth cluster that is in each cell cluster. The lack of distinct diagonal patterns indicate the cell clusters found by these algorithms weren't homogenous.

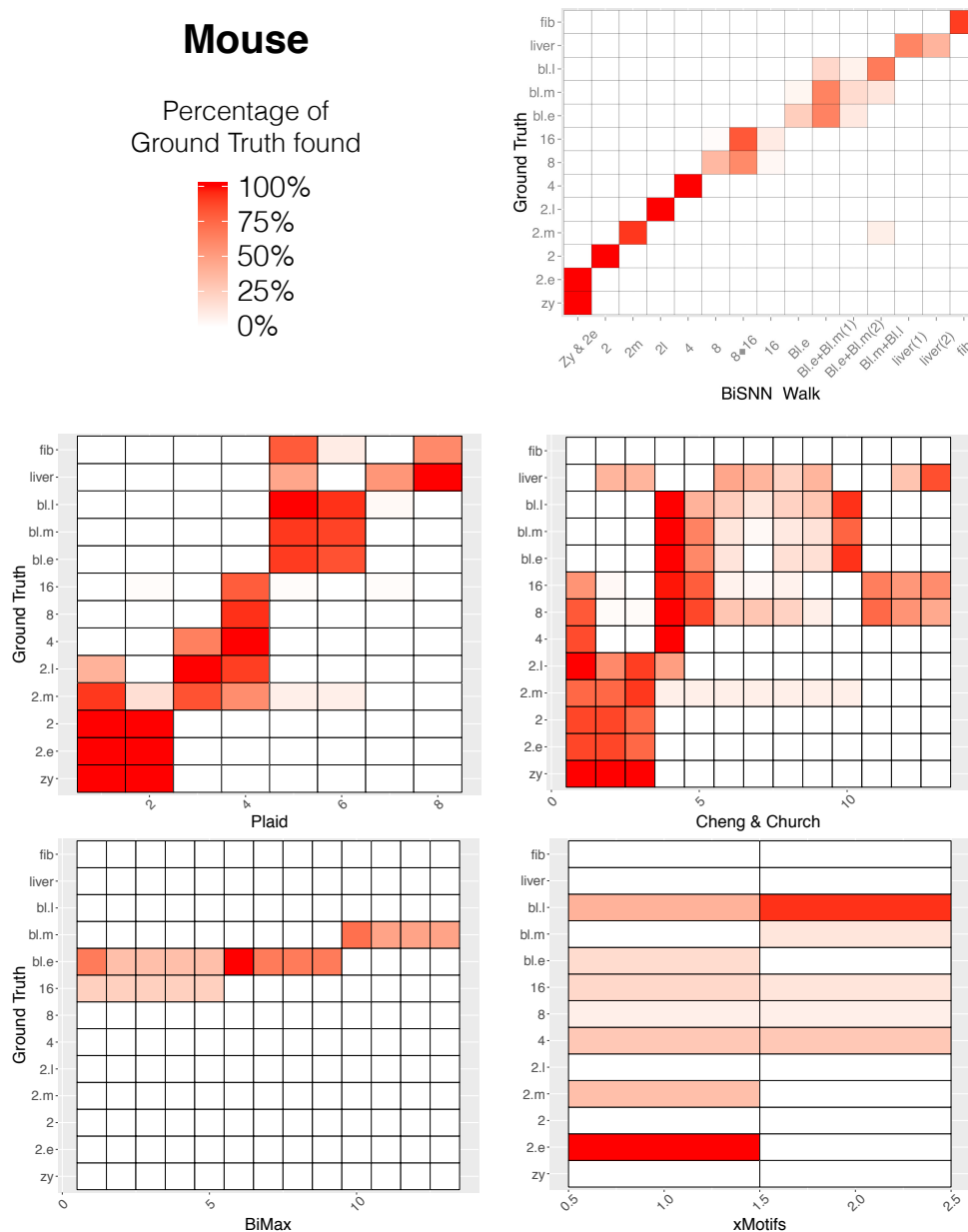


Figure 4



Figure 2.7: Cell-clusters found by selected biclustering algorithm compared to ground truth for the human embryo dataset. *x*-axis are the cell-clusters found by indicated algorithm, ordered roughly by developmental stage. *y*-axis is ground truth ordered by developmental stage. The value in each grid represents the percentage of ground truth cluster that is in each cell cluster.

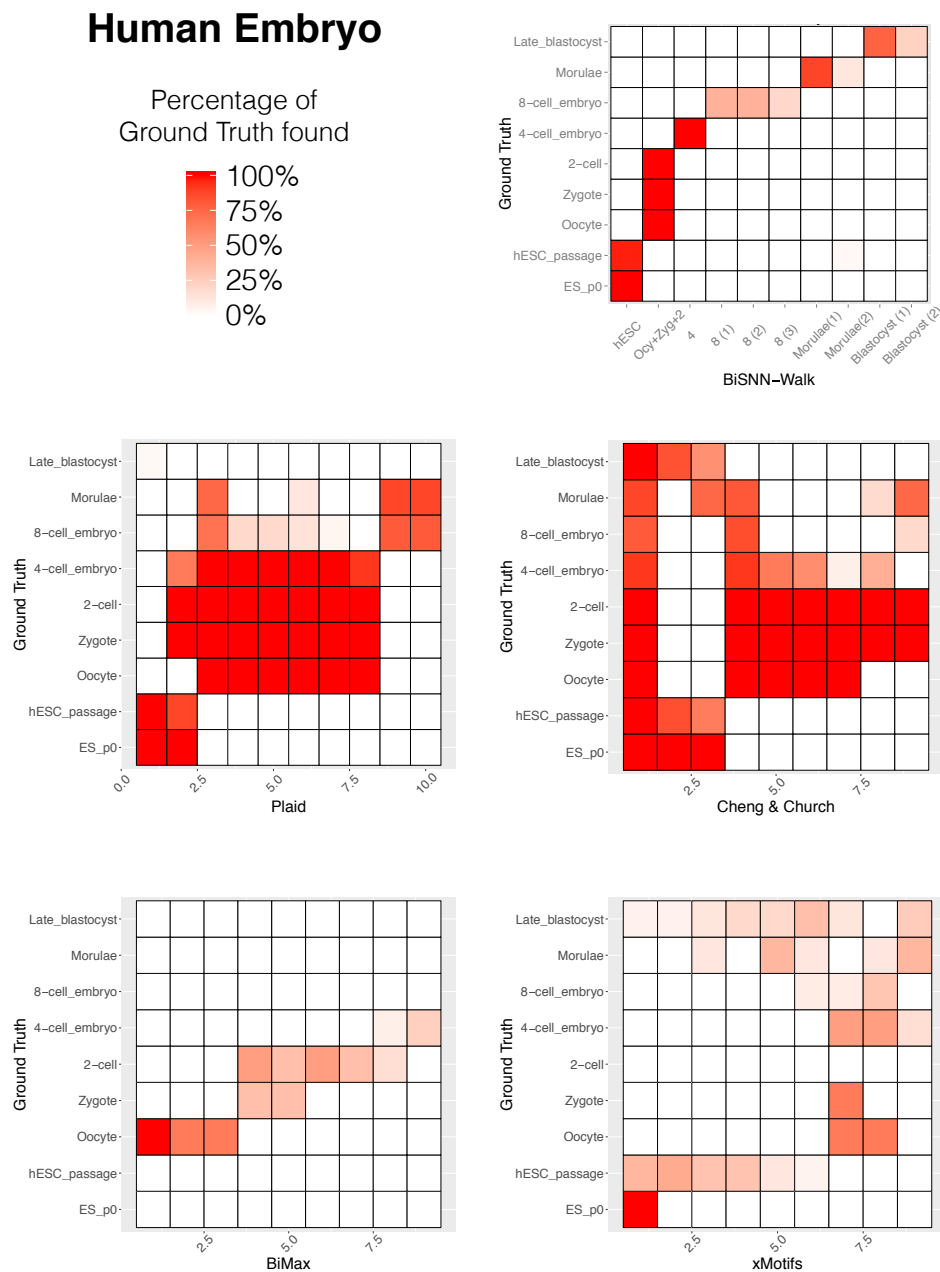


Figure 6

Figure 2.8: Cell-clusters found by selected biclustering algorithm compared to ground truth for the human cancer dataset.  $x$ -axis are the cell-clusters found by indicated algorithm,  $y$ -axis is ground truth. The value in each grid represents the percentage of ground truth cluster that is in each cell cluster.

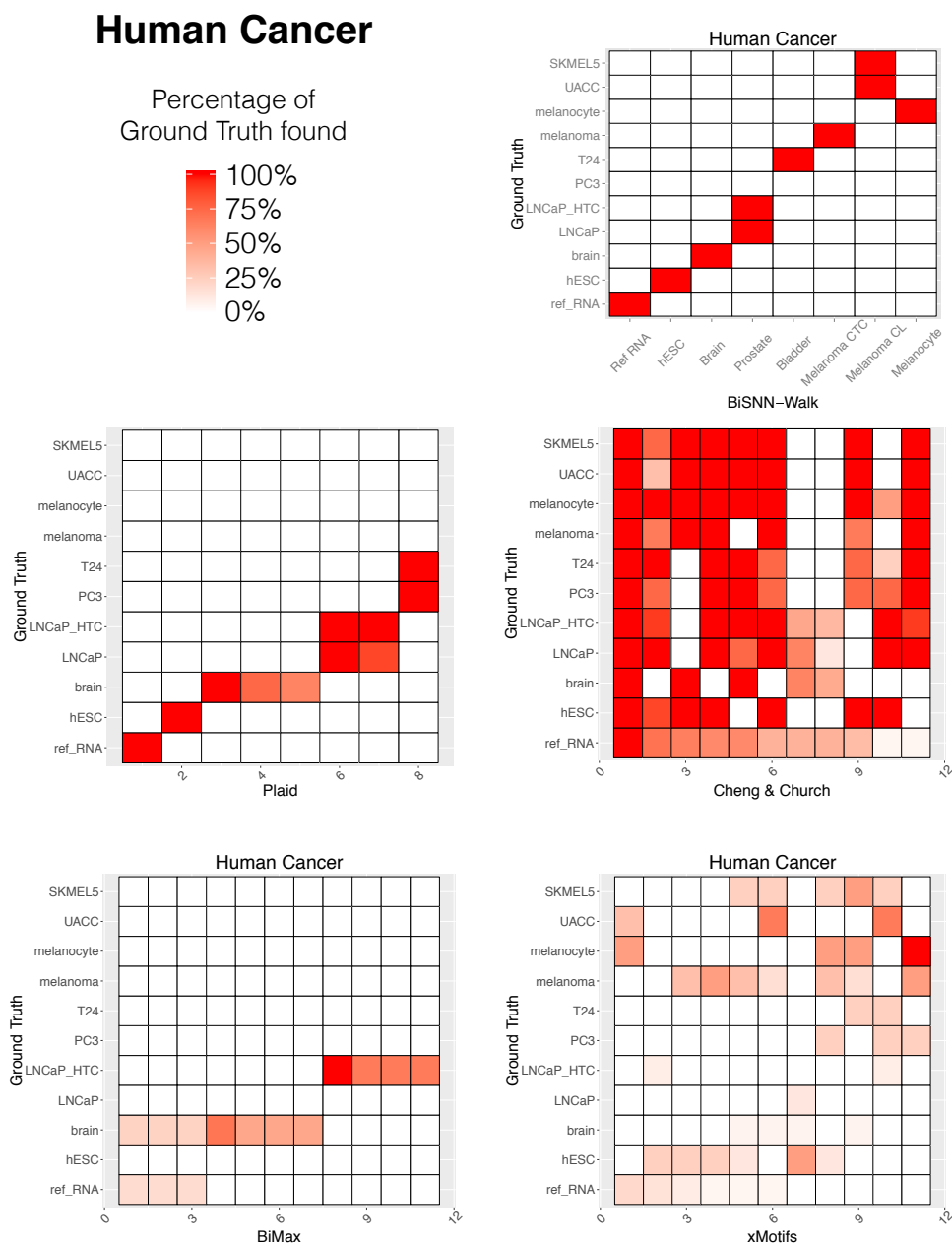


Figure 7

Figure 2.9: Heatmap of *BiSNN-Walk* (left) and *GiniClust* (right) cell clusters plotted against ground truth. x-axis are the cell-clusters found by indicated algorithm, y-axis is ground truth. The value in each grid represents the percentage of ground truth cluster that is in each cell cluster.

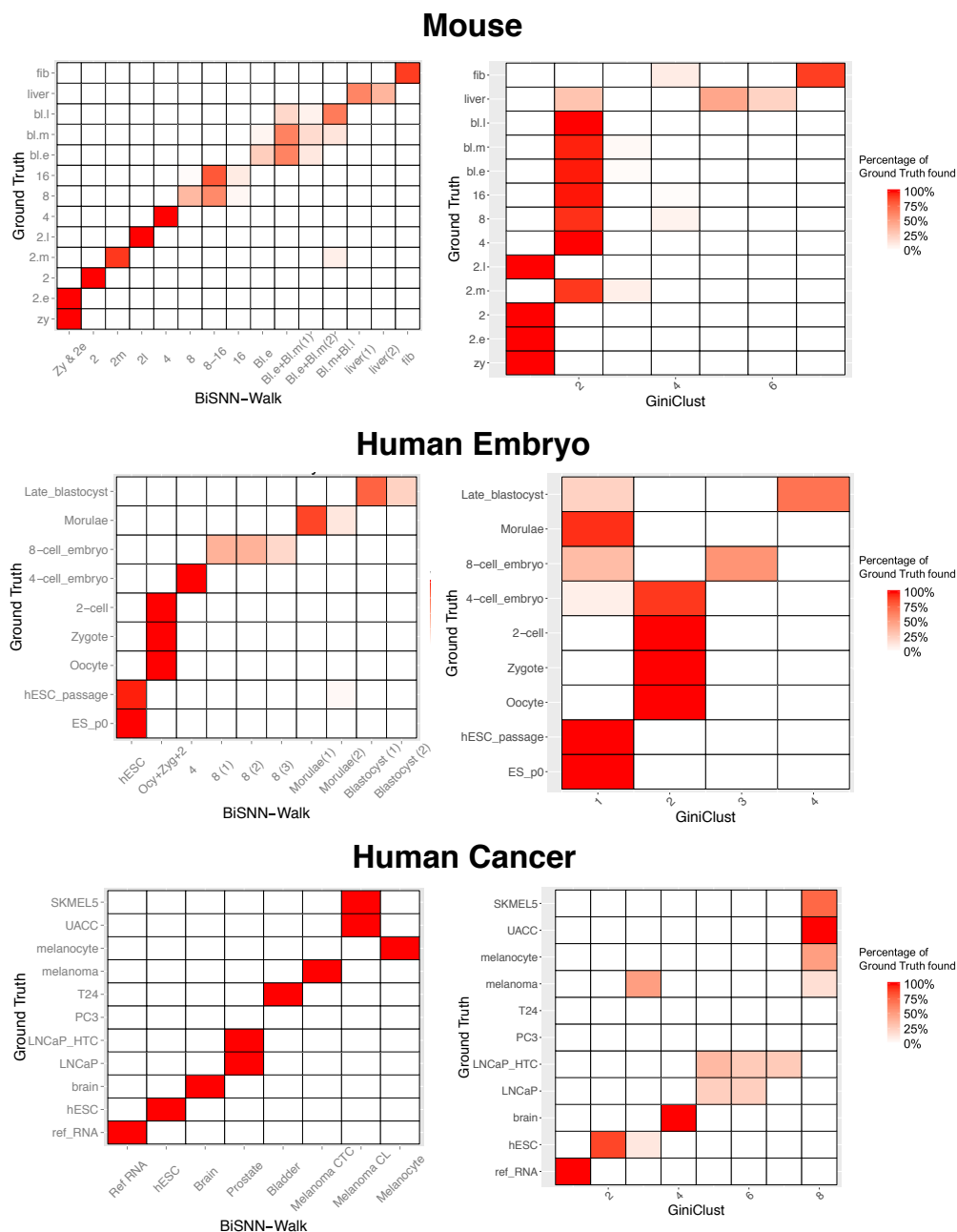


Figure 8

### 2.4.3.1 Gene Overlap Analysis

Figure 2.10 shows heatmaps of the overlap between top 100 characteristic genes of each cluster. For mouse and human embryo data, the apparent block diagonal structure in Figure 2.10 confirms the hypothesis that stages that are chronologically close will share more characteristic genes. For human cancer, as expected, we see very low overlap between clusters of different cell types except a somewhat elevated association between “Melanoma CTC” (circulating melanoma cells) and “Melanoma CL” (cancer line melanoma cells), which is reasonable, since they are of the same cell type. However the reason why only see a overlap of 15 genes may be explained by previous observation that CTC profiles are quite distinct from those of cancer cell lines [58].

### 2.4.3.2 Biological Enrichment Analysis

To perform enrichment analysis, we took the top 100 characteristic genes for each cluster and checked whether enrichment terms make sense with respect to the types of cells in the cluster. GO-term enrichment was used for human cancer data, whereas Anatomy Enrichment was performed on mouse data. Since Anatomy Enrichment is not available for human, no enrichment analysis was performed for human embryo. Selected enrichment results are shown in Table 2.7 and Table 2.8, respectively. Enrichment was performed using InterMine’s Python API [73].

Mouse result shows highly relevant enriched terms for somatic cells (fibroblast, liver) and early embryonic cells (2-cell and 4-cell stage). 8-cell to blastocyst cells were not enriched well because the clusters themselves are quite heterogeneous in the first place. Most of the human cancer cells how highly relevant enriched terms. No enrichment was found for prostate and bladder cancer clusters because none of the 82 prostate genes and 11 bladder related genes were part of the gene expression in the first place<sup>3</sup>. This indicates that these organs are poorly studied. The lack of enrichment in the “Melanoma CTC” cluster, as argued previously, is likely due to the genetic profile of CTCs exhibiting stark departure from Melanocyte and Melanoma cell-line cells, both of which are significantly enriched with the term “melanosome”.

---

<sup>3</sup>Prostate and bladder-related genes were queried from [www.humanmine.org](http://www.humanmine.org)

Figure 2.10: *Overlap between top 100 characteristic genes of each cluster. The color saturation indicates the number overlapping genes between the top 100 characteristic genes of two clusters. Maximum value for each grid is therefore 100.*

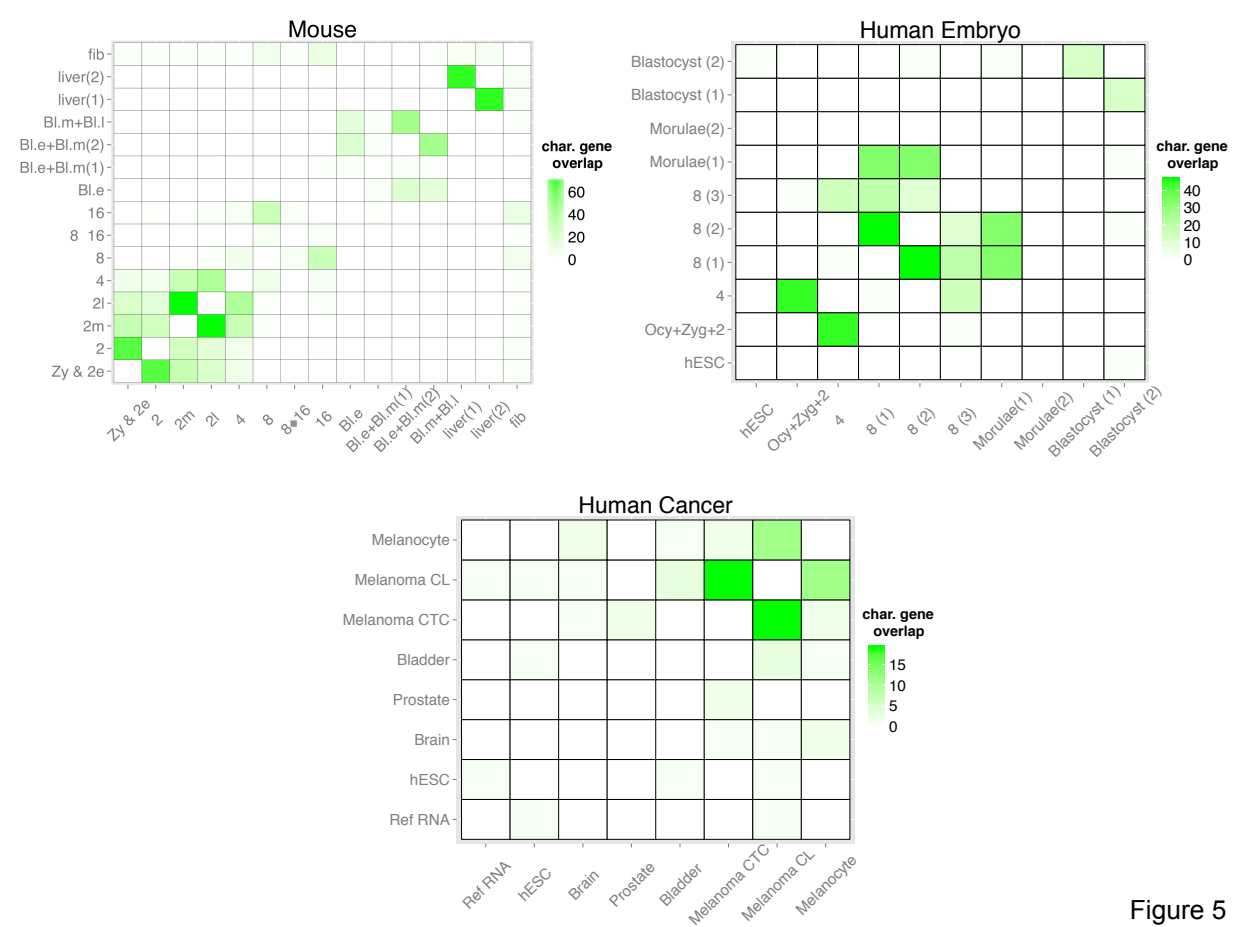


Figure 5

## 2.5 Discussion and Future Work

Clustering is an important tool for genetic analysis; however, finding important genes associated with those clusters is sometimes of more scientific interest. In this work we presented a simple, fast, and self-correcting biclustering algorithm BiSNN-Walk that is based on SNN-Cliq [84]. Results from applying BiSNN-Walk to three large scRNA-Seq studies showed that BiSNN-Walk is able to retain and even improve SNN-Cliq’s clustering performance. Moreover, since BiSNN-Walk extracts clusters one at a time according to their “tightness” (measured by transitivity and conductance), the order in which the clusters are found can be reflectively of their reliability. Being a biclustering algorithm, BiSNN-Walk also returns characteristic genes ranked by their relevance to the associated cell cluster. We have shown from multiple perspectives that BiSNN-Walk returns biologically sensible biclusters.

We used a simple entropy-based measure as a guidance to choose among initial similarity matrices. Using entropy as a surrogate for “clusterability” is a novel idea, and in our case it served well for choosing highly clusterable similarity matrix as initial input. Further investigations on this idea is underway. Several other areas of the algorithm can be improved. The stopping criterion is currently too naive and should be further investigated and improved. The definition of characteristic genes is a bit ad hoc. Though it worked well in the three public datasets, a systematic way of tuning this parameter would greatly improve usability.

Table 2.2: *Performance comparison between BiSNN-Walk, SNN-Cliq, and GiniClust. ARIs are calculated against ground truth.*

	Mouse Embryo	Human Embryo	Human Cancer
BiSNN-Walk	0.472 (311/317)	0.776 (124/124)	0.883 (82/86)
SNN-Cliq	0.574 (177/317)	0.796 (124/124)	0.661 (86/86)
GiniClust	0.098 (317/317)	0.379 (119/124)	0.870 (69/86)

Number in parentheses is (number of cells clustered / total number of cells)

Table 2.3: *Collection details of mouse embryonic cells*

Developmental Stage	Hrs after Ovulation	No. of Samples
Zygote	20-24	4
Early 2 Cell	31-32	8
Mid 2 Cell	39-40	12
Late 2 Cell	46-48	10
4 Cell	54-56	14
8 Cell	68-70	48
16 Cell	76-78	58
Early Blastocyst	86-88	43
Mid Blastocyst	92-94	60
Late Blastocyst	100-102	30
C57 2 Cell	NA*	8
Liver	-	13
Fibroblast	-	10

\*In correpondence with the author of the paper, the “C57 2-Cell” cells have different genetic background than the other 2-cell cells, and are of low sequencing quality. So their exact placement of the C57 2-cell cells in the 2-cell developmental stage is unclear.

Table 2.4: *Collection details of human embryonic cells*

Developmental Stage	Hrs (after fertilization)	No. of Samples
Oocyte	4 (after retrieval)	3
Zygote	19	3
2 Cell	27	6
4 Cell	48	12
8 Cell	72	20
Morulae	96	16
Late.Blastocyst	144	30
hESC	~30 days	32



Table 2.5: *Collection details of human somatic/cancer cells*

Cell Type	No. of Samples
universal human reference RNA	20
brain	16
prostate cancer cell line (PC3)	4
bladder cancer cell line (T24)	4
melanoma derived circulating tumor cells (CTC)	6
melanocytes	2
melanoma cancer	7
embryonic stem cells	8
prostate cancer cells (picked from petri-dish)	7
prostate cancer cells (isolated by EPCAM marker)	8

	Time Elapsed (s)	Avg No. Inner Loops	No. Clusters Found (No. Real Clusters)
Mouse	311	4	14 (13)
Human Embryo	109	3.8	10 (9)
Human Cancer	86	3.7	8 (11)

Table 2.6: *BiSNN-Walk Output Information.* “Avg No. Inner Loops” is average number of inner loops called per round of outer loop, and can be interpreted as either the speed of convergence or quality of the data, as faster convergence is achieved with cleaner data.

Cluster Name	Cell Types	Enriched Terms
Zy & 2.e	Zygote Early 2-cell	germ cell of ovary germ cell of gonad 2-cell stage conceptus
2	2-cell	2-cell stage conceptus 1-cell stage conceptus
2.m	mid 2-cell	2-cell stage conceptus 1-cell stage conceptus 4-cell stage conceptus
2.l	late 2-cell	2-cell stage conceptus 1-cell stage conceptus
4	4-cell	2-cell stage conceptus
8	8-cell	
8-16	8-cell 16-cell	
Bl.e	early blastocyst mid blastocyst	embryo endoderm endoderm
Bl.e+Bl.m(1)	early blastocyst mid blastocyst late blastocyst	primitive endoderm endoderm early conceptus
Bl.e+Bl.m(2)	early blastocyst mid blastocyst late blastocyst	
Bl.m+Bl.l	late blastocyst mid blastocyst mid 2-cell	
liver(1)	liver	liver liver lobe liver and biliary system
liver(2)	liver	liver liver lobe liver and biliary system
fb	fibroblast	tendon mesenchyme bone

Table 2.7: *Enriched terms for mouse embryo data set. As one can see, liver, fibroblast, and early developmental stages were well enriched. Early- and mid-blastocyst also clusters saw relevant enrichment. For Bl.e+Bl.m(2) and Bl.m+Bl.l cluster, significant terms were found, but were not reported since they did not seem relevant to the developmental stage. No significant terms were found for 8-cell and 16-cell clusters. Please refer to Supplementary file for full list of enriched terms. EMAPA mouse development anatomy ontology database was used for enrichment.*

Cluster Name	Cell Types	Enriched Terms
Ref_RNA	reference RNA	
hESC	human embryonic stem cells	stem cell population maintenance somatic stem cell population maintenance embryo development
Brain	brain	neuron part axon part synapse part
Prostate	LNCaP cell line cells LNCaP_HTC petridish extracted	
Bladder	T24 bladder cancer cell line	
Melanoma CTC	circulating melanoma tumor cells	
Melanoma CL	SKMEL5 melanoma cell line UACC melanoma cell line	melanosome membrane
Melanoma CTC	circulating melanoma tutor cells	melanosome membrane

Table 2.8: *Enriched terms for human cancer data set. As one can see, four of the eight clusters saw relevant enrichment. Ref\_RNA saw a wide mix of significant terms, as expected. Significantly enriched terms were returned for all clusters except Melanoma CTC, but were not reported since they did not seem relevant to the particular cell type. For the full list of enriched terms please refer to Supplementary Information.*

## Chapter 3

# A more powerful and flexible method for measuring Amyloid- $\beta$ accumulation through linear regression

### 3.1 Introduction

Alzheimer's Disease (AD) is currently an incurable neuro-degenerative disorder whose symptoms include memory loss, depression, bipolar disorder, irritability etc. Despite decades of medical advances, it is still unclear what causes its onset and progression. Old age is considered the most common trigger for AD, however, innate factors such as genetics as well as acquired factors such as depression were also associated with the disease's onset [6]. Among the many hypothesized causes of AD, amyloid hypothesis and  $\tau$  hypothesis are the most widely accepted [50]. Amyloid and  $\tau$  hypothesis both stipulate that the increased accumulation of the respective protein (Amyloid- $\beta$  or  $A\beta$  protein in the former and  $\tau$  protein the latter) due to lack of clearance or overproduction, leads to a cascade of events ultimately resulting in neuronal death, a process that can begin as early as 1-2 decades prior to onset of any clinical symptoms. In the late 1970's and early 1980's, amyloid and  $\tau$  were identified as among the potential culprits for the onset of AD. Numerous genotype-phenotype and genetic linkage studies on familial AD in the 1990's linked AD onset to mutations in genes associated with the production or deposition of amyloid proteins [26, 66, 76], providing further support for amyloid hypothesis. With copious amount of experimental evidence,  $A\beta$  soon became the main focus of AD research for the next two and half decades, and to this day remains a widely accepted therapeutic targets across the globe [68]. Medical and pharmaceutical effort by targeting amyloid, however, has not seen significant return on investment, as all major amyloid-targeting drug development thus far have ended in failure [36]. In addition, with recent advancement in amyloid imaging technology, several contradicting observations

against the amyloid hypothesis emerged, such as studies where  $A\beta$  accumulation not correlating well with disease progression [17, 46]. At the same time, multiple evidence supporting  $\tau$  also surfaced, such as spatial patterns of  $\tau$  accumulation correlating better patterns of neurodegeneration than amyloid [3, 52].  $\tau$  pathology, therefore, re-emerged as the main alternative to amyloid for medical and pharmaceutical research. The merits and foibles of both hypothesis are still actively debated in the AD community, and both are presently actively pursued. Work presented here applies to amyloid imaging data.

The state of the art technology for imaging  $A\beta$  accumulation is florbetapir-PET (Positron Emission Tomography). Florbetapir is a radioactive tracer specifically engineered to bind to  $A\beta$  proteins [81]. The PET machine will pick up on the positron emitted by the tracer and locate its position, and in turn produce a  $256 \times 256 \times 80$  3D image, with each voxel reflecting the level of  $A\beta$  accumulation in that area. The PET voxel intensities are then translated into numerical representation called Standard Uptake Values (SUV). To avoid working with millions of voxel-wise SUVs of an image, the brain is usually partitioned into different functional anatomical regions (the exact partition differ according to imaging pipeline) called regions of interests (ROIs). The average SUV of all pixels in each region are calculated, and these ROI-level SUVs are the values that people typically work with. Subsequent mentions of SUV will refer to ROI-level SUVs instead of voxel-wise SUVs.

ROIs fall into two categories: target and reference. Target ROIs are regions where  $A\beta$  may accumulate, while reference ROIs are regions that are relatively disease free. Under ideal conditions, tracers would bind exclusively to  $A\beta$  proteins, so that the signal collected by PET scans is directly proportional to the actual accumulation of  $A\beta$  plaque, such binding is called **specific binding**. However, in practice, tracers may be free floating (don't bind to anything) or bind to other materials. This is called **non-displaceable binding**. See Figure 3.1 for a schematic.

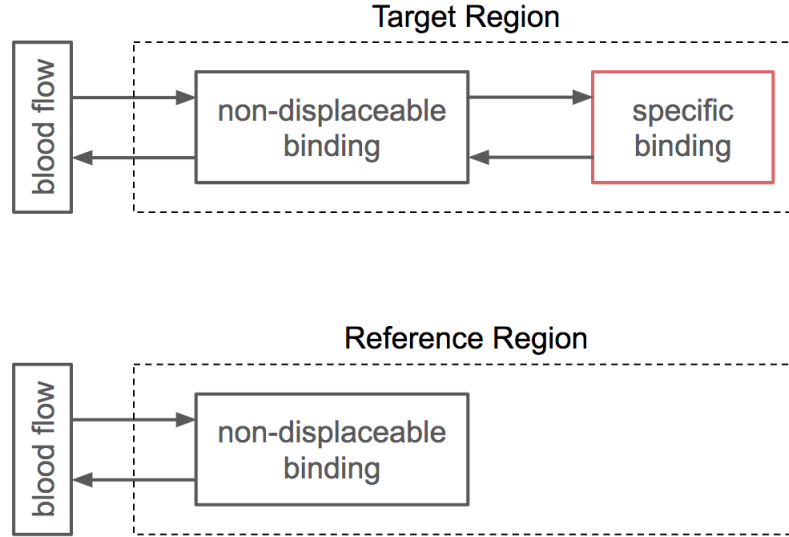


Figure 3.1: *Pharmacokinetic schematic of the sources of PET signals in target vs reference regions. Blood flow carries tracer molecules, which enter and exit the non-displaceable binding compartments, where tracers are bound to non-amyloid material. In both target and reference regions, non-displaceably bound tracer molecules can return to blood flow (unbound state); however, in target regions where  $A\beta$  proteins tend to accumulate, tracers can also become bound to the protein, a process called specific binding. The signal we wish to measure is the specific-binding compartment in red in the target region; however, in practice the actual signal is the aggregate from both specific and non-displaceable compartments, therefore, the signal from the reference region is used as a benchmark for the non-displaceable portion of the target ROI signal.*

Since the signal obtained by PET in the target region is a combination of specific and non-displaceable, the latter must be correctly accounted for in order for the SUV to correctly reflect the state of accumulation. To this end, the field has converged to a measure called Standard Uptake Value Ratio (SUVr), which is a rough estimate of the binding potential, an important constant in particle kinetic theory [40]. SUVr is defined as

$$SUVr = \frac{\text{SUV of target region}}{\text{SUV of reference region}}$$

Despite its ubiquity, SUVr has several drawbacks. First, as a ratio-based statistic with a random numerator and denominator, SUVr's statistical properties can be tricky to understand. Secondly, changes in  $A\beta$  accumulation can be masked by the interaction between the random numerator and the random denominator. Lastly, SUVr does not allow straightforward incorporation of exogenous information such as race, gender, and age, therefore accounting for the interaction effect of these predictors would be even more difficult. Therefore, we propose an alternative measure to SUVr, which we dub the  $\Delta$ -measure, in hope to

side step the aforementioned issues. As of the time of the writing the authors are not aware of any alternative to SUVR.

This chapter is organized as follows: Section 3.2 introduces the two amyloid image data sets we will analyze; Section 3.3 details the exploratory analysis that unveiled a linear relationship between target and reference SUV common to both data sets. Based on this linear relationship, we introduce an alternative measure (dubbed  $\Delta$ -measure) to SUVR for gauging amyloid accumulation in Section 3.4. We propose a generating model for target SUV in Section 3.5.1, which serves as the statistical backdrop for analyzing the statistical behaviors of the two measures in the subsequent Section 3.5.2. Finally, Section 3.6 compares the performance of SUVR and  $\Delta$ -measure on both real and simulated data.

## 3.2 Data

Two datasets were obtained for subsequent analysis.

The first data set is from Genentech’s ABBY/BLAZE Phase 2 trials (Clinical Trial ID: NCT01723826). This dataset consists of SUVs from 30 AD-positive patients at 16 ROIs (12 target, 4 reference ROIs) over 3 scans (entry, 47 weeks, 69 weeks). Only the entry and 47-weeks scans were used for subsequent analysis since 10 patients dropped out of the study at week 69, greatly reducing the fidelity the week-69 analysis results. All patients were diagnosed as AD and  $A\beta+$  at entry scan, and were age-controlled at 55-years or older. Image processing was done by INVICRO (previously Molecular Neuroimaging).

The second dataset was obtained from the ADNI GO and ADNI 2, which we will collectively refer to as “ADNI”<sup>1</sup>. To make the the ADNI dataset comparable to BLAZE, we only used data from 40 AD patients in the study. Each patient was scanned twice (entry scan, two year follow up scan), and data on 9 ROIs (4 target, 5 reference) are provided. The image processing was done by the Jagust Lab at UC Berkeley.

Although the detailed delineation of target and reference regions may differ by processing pipelines, they agree on the general anatomical locations. To keep the analysis tractable, subcortical white matter was used as the reference ROI for both data sets, as recommended by multiple recent studies [67, 70, 39, 9].

---

<sup>1</sup> Data used in preparation of this article was obtained from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) database ([adni.loni.usc.edu](http://adni.loni.usc.edu)). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: [http://adni.loni.usc.edu/wp-content/uploads/how\\_to\\_apply/ADNI\\_Acknowledgement\\_List.pdf](http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf)



### 3.3 Explore the linear relationship between target and reference SUVs

Let  $T_t^i$ ,  $R_t^i$  denote respectively the target and reference region SUV at time  $t$  for patient  $i$ . Unless otherwise mentioned,  $t = 1$  will denote entry scan, and  $t = 2$  the follow up scan.

Figure 3.2 plots  $T_1$  and  $R_1$  for both ADNI and BLAZE where we see  $T$  and  $R$  exhibits a clear linear relationship. This observation suggests a linear relationship of the form

$$T_1^i = \alpha_1 + (\beta_1 \times R_1^i) + \epsilon_1^i \quad (3.1)$$

Where  $\alpha_1$  denotes the contribution to  $T_1$  (target ROI SUV) from specific binding,  $\beta_1 \times R_1$  is the contribution from non-displaceable binding, and  $\epsilon_1$  is the error.

Plotting the data for time 2 shows that the linear relationship between  $T$  and  $R$  carries across time (Figures B.2, B.5). This means the following modification to Eq. 3.1 provides a reasonable description of  $T_2$ :

$$T_2^i = \alpha_2 + (\beta_2 \times R_2^i) + \epsilon_2^i \quad (3.2)$$

Let  $e_1$  and  $e_2$  denote the least squares residuals of Eq. 3.1 and 3.2, respectively. When we plot  $e_1$  vs  $e_2$ , another linear pattern emerges (Figure 3.3), suggesting a strong patient effect, i.e if patient is above the fitted line (from Eq. 3.1) at  $t_1$  then they will likely be above fitted line at  $t_2$ . Let  $Z^i$  denote the patient effect for patient  $i$ . The updated relationship should be of the form

$$T_t^i = \alpha_t + (\beta_t \times R_t^i) + Z^i + \epsilon_t^i$$

Using feasible generalized least squares, one can check that it's reasonable to assume  $\beta$  to be constant across time (details presented in Section B.3). Piecing together the above observations suggests that Eq. 3.3 provides a reasonable description between target and reference SUVs.

$$T_t^i = \alpha_t + (\beta \times R_t^i) + Z^i + \epsilon_t^i \quad (3.3)$$

The linear relationship described by Eq. 3.3 will be used in Section 3.4.2 to derive a new measure for gauging amyloid accumulation. In Section 3.5.1, we will leveraged this relationship again to propose a data generating model to approximate the generation of target SUVs.

It's worth noting that even though ADNI and BLAZE uses completely different image acquisition/processing pipeline, the linear relationship persists in both datasets (A vs B in Figures 3.2, 3.3, and 3.4). In addition, the linear relationship shown in Figures 3.2, 3.3, and 3.4 carries across all target/reference combinations and across time (see Appendix B.1).

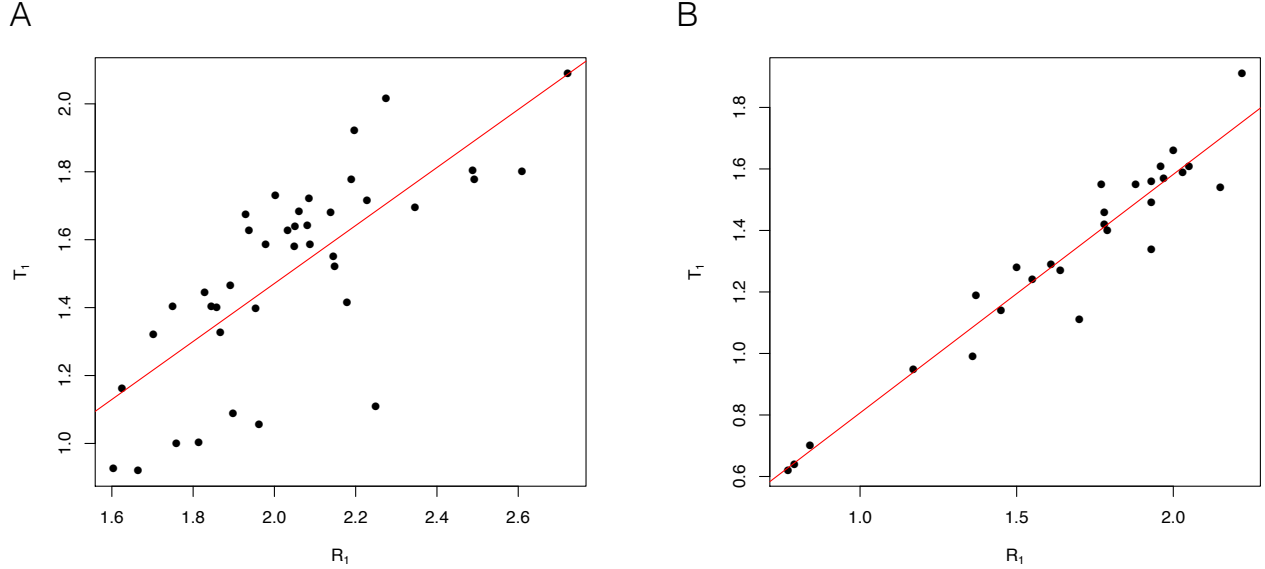


Figure 3.2:  $T_1$  vs  $R_1$ .  $T_1$  and  $R_1$  denote target and reference SUV at  $t_1$  respectively. Target: frontal cortex, Reference: subcortical whitematter. A: ADNI, B: BLAZE.

These observations indicate that the linear relationship is likely biologically meaningful.

### 3.4 Measures for gauging $A\beta$ accumulation

In this section we first give a brief overview of how SUVr is commonly used to measure amyloid accumulation in Section 3.4.1. We then introduce an alternative measure in Section 3.4.2 that was derived from the linear relationship between target and reference SUVs observed in Section 3.3.

#### 3.4.1 SUVr

As mentioned in Introduction, SUVr is the de-facto measure to gauge amyloid accumulation. To quantify the *change* in amyloid accumulation between time points for patient  $i$ , SUVr at two different time points are subtracted, i.e  $\Delta SUVr = SUVr(t_2) - SUVr(t_1)$ . A common method to determine whether a group of  $n$  patient saw significant accumulation is to perform the one-sample T-test on the collected  $\Delta SUVr$  differentials for each patient, i.e

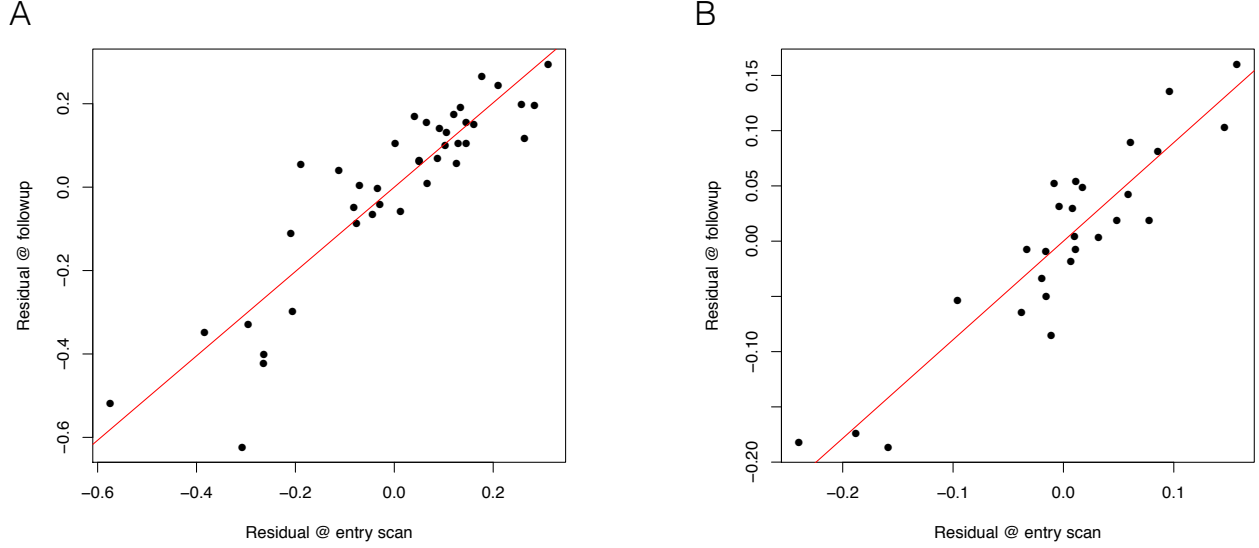


Figure 3.3:  $e_1$  vs  $e_2$ .  $e_t$  is the residual from regression  $T_t$  on  $R_t$ , i.e the realization of  $\epsilon_t$  from Eq. 3.1 and 3.2. The figure shows the scatter plot between  $e_1$  and  $e_2$ . Linear relationship indicates a strong patient effect. Target: frontal cortex, Reference: subcortical white matter. A: ADNI, B: BLAZE.

$\{\Delta SUVr^1, \dots, \Delta SUVr^n\}$ .

### 3.4.2 $\Delta$ -measure: leveraging linear relationships between target and reference SUVs

Eq. 3.3 suggests a linear relationship between the changes in SUVs of the form

$$\Delta T^i = \Delta \alpha + (\beta \times \Delta R^i) + \epsilon^i \quad (3.4)$$

Where  $\Delta T^i = T_2^i - T_1^i$ ,  $\Delta R^i = R_2^i - R_1^i$ ,  $\Delta \alpha = \alpha_2 - \alpha_1$ , and  $\epsilon^i = \epsilon_2^i - \epsilon_1^i$ . Figure 3.4 shows the scatter plot of  $\Delta T$  vs  $\Delta R$ , where we observe a distinct linear relationship. One may notice that the linear relationship between  $\Delta T$  and  $\Delta R$  appear more pronounced than that between  $T$  and  $R$ . This is because the patient effect  $Z^i$  is canceled during subtraction, reducing the error.

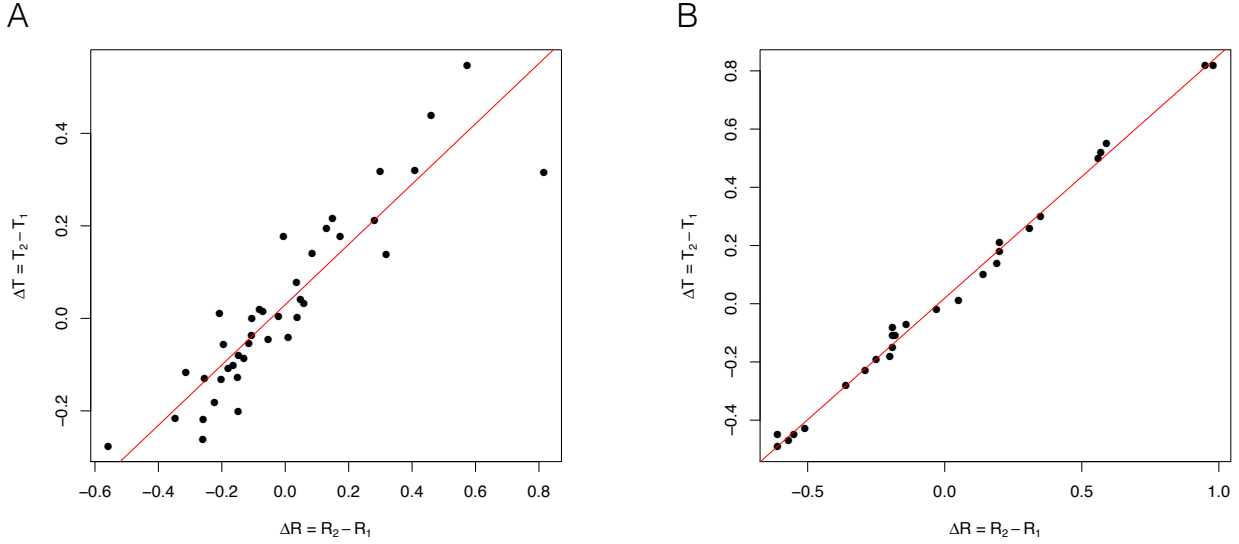


Figure 3.4:  $\Delta T$  vs  $\Delta R$ . Removing the patient effect by subtracting SUVs at  $t_1$  from that at  $t_2$  ( $\Delta T = T_2 - T_1$ ,  $\Delta R = R_2 - R_1$ ) produced a more pronounced linear relationship than in Figure 3.2. Target: frontal cortex, Reference: subcortical whitematter. A: ADNI, B: BLAZE.

This formulation lends itself to an alternative normalization to SUVr. Eq. 3.4 provides a clear relationship between the change in target SUV and reference SUV: if  $\Delta R = 0$ , then  $\Delta T = \Delta\alpha + \epsilon$ , that is, if we see no change in reference SUV (i.e non-displaceable binding remains constant), then the expected change in target SUV (i.e specific binding) is  $\Delta\alpha$ .

To determine group level amyloid accumulation, we need to determine whether  $\Delta\alpha$  is significantly different from zero. Since  $\Delta$ -measure is derived from simple linear regression, Kendall–Theil Sen Siegel non-parametric linear regression can be used to estimate  $\Delta\alpha$ , call this estimator  $\tilde{\Delta\alpha}$ . A bootstrapped confidence interval of  $\tilde{\Delta\alpha}$  can be constructed to determine whether there is statistically significant group-level accumulation.

A parametric alternative is available if we make a fairly standard assumption that the error  $\{\epsilon^i\}$  from Eq. 3.4 are independent and identically distributed  $N(0, \sigma_\epsilon^2)$  for some constant  $\sigma_\epsilon^2$ . Under these assumptions, the least squares estimator for  $\Delta\alpha$ , denoted  $\hat{\Delta\alpha}$ , has the following distribution

$$\hat{\Delta\alpha} \sim N\left(\Delta\alpha, \hat{\sigma}_\epsilon^2 \frac{\sum_i R^i}{n \sum_i (R^i - \bar{R})^2}\right)$$

Where  $\bar{R} = \frac{1}{n} \sum_{i=1}^n R_i$ , and  $\hat{\sigma}_\epsilon = \frac{1}{n-2} \sum_{i=1}^n (T^i - \hat{T}^i)^2$ ,  $\hat{T}^i$  is the fitted value for  $T^i$ . To

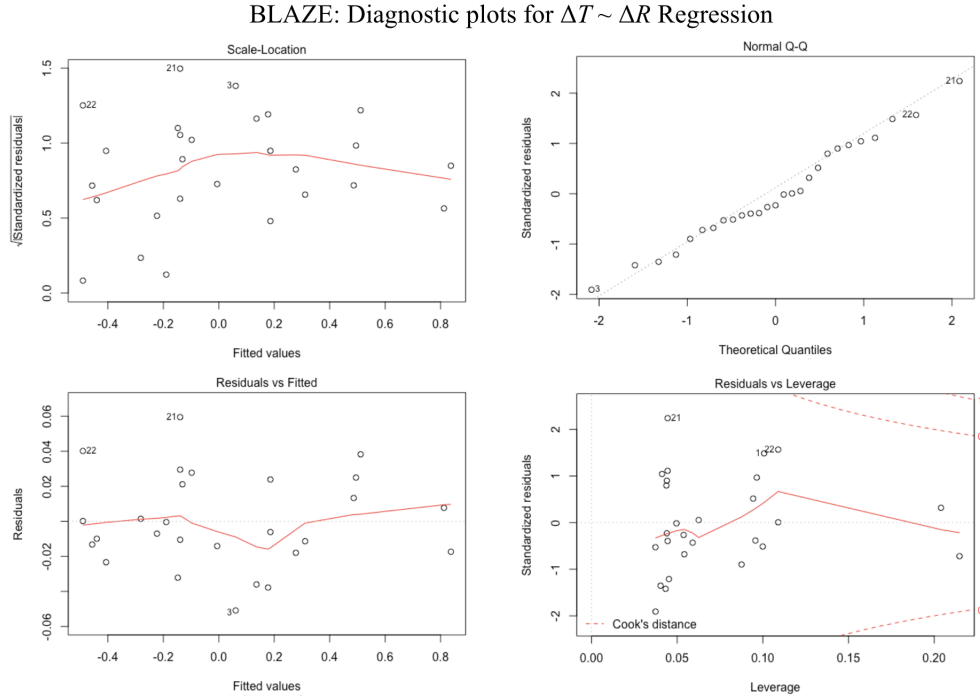


Figure 3.5: *Diagnostic plots for  $\Delta T \sim \Delta R$  regression using BLAZE data. Scale-location and residual-fitted plots show no evidence of heteroskedasticity; QQ plot indicates that normality assumption holds well; and leverage plot identified no influential points.*

determine group level accumulation, one then simply tests  $H_0 : \Delta\alpha = 0$  vs  $H_a : \Delta\alpha \neq 0$  using  $\hat{\Delta\alpha}$  as test statistic.

Figure 3.5 shows the diagnostic plots of least squared fit for Eq. 3.4. Scale-location and residual-fitted plots show no evidence of heteroskedasticity; QQ plot indicates that normality assumption holds well; and leverage plot identified no influential points. Considering each observation correspond to distinct patient, the independence assumption is also sensible. These evidence suggest that i.i.d normal assumption mentioned previously is well supported by the data. Testing of  $\Delta$ -measure in subsequent analysis will be done using the parametric approach.

### 3.5 Compare statistical properties of SUVr and $\Delta$ -measure under assumed parametric generating model

Since SUVr and  $\Delta$ -measure are alternate measures, it would be illuminating to study the statistical behavior of the two measures under a common set of modeling assumptions. First, in Section 3.5.1, we propose an data generating model that approximates the generation of

the target SUVs, and offer empirical support for our proposed model. Then, in Section 3.5.2, we study the statistical behavior of the two measures under the proposed generating model.

### 3.5.1 Approximating the generation of target SUVs

We have seen in Section 3.3 that there is clear linear relationship between target and reference SUVs (Eq. 3.3). However, to transform Eq. 3.3 into a data-generating model for  $T_t^i$ , we will need to make distributional assumptions about  $\epsilon_t^i$ ,  $Z^i$ , and  $R_t^i$ .

In accordance to classical linear mixed effect model, we assume

1. Errors are i.i.d Gaussian, i.e  $\epsilon_t^i \underset{i.i.d}{\sim} N(0, \sigma_\epsilon^2), \forall t, i$
2. Patient effects are i.i.d Gaussian, i.e  $Z^i \underset{i.i.d}{\sim} N(0, \sigma_Z^2) \forall i$  for some constant  $\sigma_Z^2$ .

To assess the plausibility of the assumption 1, we first note that, from the QQ plot in Figure 3.5, it's fair to assume  $\epsilon^i$  from Eq. 3.4 is normally distributed. Also note that  $\epsilon^i = \epsilon_2^i - \epsilon_1^i$  ( $\epsilon_t^i$  from Eq. 3.3). While these observations do not imply  $\epsilon_1^i, \epsilon_2^i$  are individually Gaussian, but it does make such assumption plausible.

The similar argument applies to assumption 2. The diagnostic QQ plots for the regression stated by Eq. 3.1 for both BLAZE (Figure 3.6 and ADNI (Figure B.7) indicate that  $Z^i + \epsilon_1^i$  being Gaussian is reasonable, making  $Z^i$  being Gaussian a fair assumption as well.

We make the additional assumption that

3.  $R_t^i \sim N(\mu_R, \sigma_R^2) \forall i, t$  for some constants  $\mu_R$  and  $\sigma_R^2$

We claim plausibility of assumption 3 using the following evidence. We first assess normality using QQ plots. Figure 3.7 shows the QQ plots of  $R_1$  for ADNI and BLAZE. Despite ADNI having slightly heavy tails and BLAZE exhibiting left skew, most of the points track the QQ line well, giving credibility to the Gaussian assumption.

To assess whether  $R_2^i$  have the same mean, variance, and distribution as  $R_1^i$ , we used the two sided two sample t-test for different means, F-test for different variances, and Kolmogorov–Smirnov for different distributions. Results are recorded in Table 3.1, where it clearly shows that there is not enough evidence to reject that  $R_1^i \stackrel{D}{=} R_2^i$  in distribution, as the p-values for all tests are far from the thresholds usually considered statistically significant.

BLAZE: Diagnostic plots for  $T_1 \sim R_1$  Regression

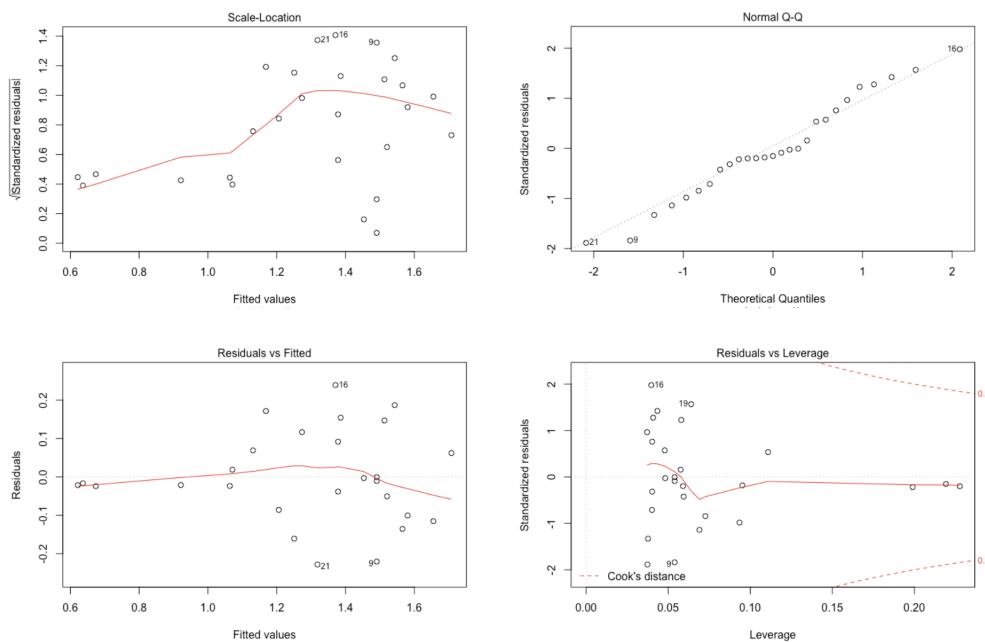


Figure 3.6: Diagnostic plots for  $T_1 \sim R_1$  regression using BLAZE data.

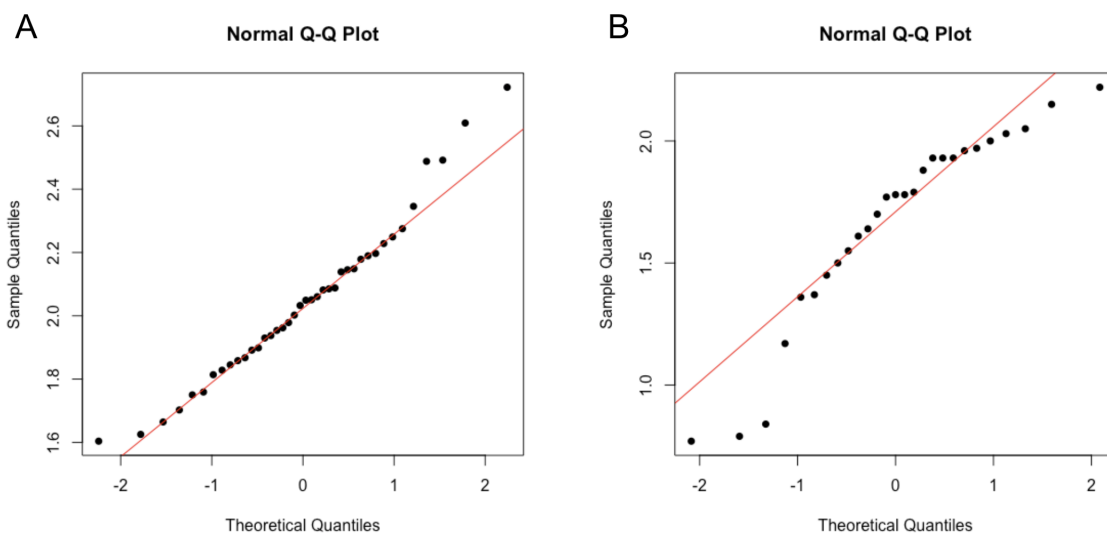


Figure 3.7: QQ plots of reference SUV (subcortical white matter) for entry scan. A: ADNI, B: BLAZE.

Table 3.1: P-values from testing whether  $R_1 \stackrel{D}{=} R_2$ 

	ADNI	BLAZE
T-test	0.87	0.94
F-test	0.23	0.93
KS-test	0.57	0.32

Table 3.1: *P-values of two sided T-test (for different means), F-test (for different variance), and Kolmogorov–Smirnov test (for difference in distribution) to ascertain whether  $R_1 \stackrel{D}{=} R_2$* 

In addition, we make the following assumptions that are difficult to verify using data at hand, but we believe are fairly standard and non-controversial.

4. Constant correlation of reference SUV across patient/time:  $Cor(R_t^i, R_\tau^i) = \rho, \forall i, t \neq \tau$
5. Variables are pairwise independent:  $R_t^i \perp \epsilon_t^i \perp Z^i, \forall i, t$

In summary, we believe the following is a reasonable generating model for target SUVs

$$T_t^i = \alpha_t + (\beta \times R_t^i) + Z^i + \epsilon_t^i \quad (3.5)$$

where  $R_t^i \sim N(\mu)_R, \sigma_R^2), \forall i, t$

$$Z^i \sim N(0, \sigma_Z^2), \forall i$$

$$\epsilon_t^i \sim N(0, \sigma_\epsilon^2), \forall i, t$$

$$Cor(R_t^i, R_\tau^i) = \rho, \forall i, t \neq \tau$$

$$R_t^i \perp \epsilon_t^i \perp Z^i, \forall i, t$$

### 3.5.2 Statistical properties of SUVR and $\Delta$ -measure under the assumed generating model

In this section we investigate the statistical behavior of SUVR and  $\Delta$ -measure under the generating model proposed in Section 3.5.1.

**Proposition 3.5.1.** Given the generating model (Eq. 3.5), the expected value and variance of  $\Delta SUVr = \frac{T_2}{R_2} - \frac{T_1}{R_1}$  are

$$E[\Delta SUVr] = \frac{\Delta \alpha}{\mu_R} (1 + CV_R^2) \quad (3.6)$$

where

$$CV_R = \frac{\sigma_R}{\mu_R} \text{ is the coefficient of variation of } R$$



and a second-order approximation of the variance is

$$\begin{aligned} Var(\Delta SUVr) \approx & \left( \frac{1}{\mu_R^2(1 + CV_R^2)} + \frac{2CV_R^2(2 + CV_R^2)}{\mu_R^2(1 + CV_R^2)^3} \right) (2\sigma_Z^2 + 2\sigma_\epsilon^2 + \alpha_1^2 + \alpha_2^2) - \\ & \frac{(\alpha_1 + \alpha_2)^2}{\mu_R^2} [1 + CV_R^2]^2 - \\ & 2 \times \frac{1 + 2CV_R^2 + 10\rho CV_R^2}{(1 + 2\rho CV_R^2)^3} \frac{\alpha_1\alpha_2 + \sigma_Z^2}{\mu_R^2} \end{aligned} \quad (3.7)$$

And  $\hat{\Delta}\alpha$  is normally distributed with mean and variance stated below:

$$E(\hat{\Delta}\alpha) = \Delta\alpha \quad (3.8)$$

$$Var(\hat{\Delta}\alpha) \approx \frac{2\sigma_\epsilon^2}{n} \left[ \frac{n-2}{n-1} + \frac{n}{(n-1)^2} \right] \quad (3.9)$$

*Proof.* The derivation of Proposition 3.5.1 involves repeated use of Taylor's theorem. Details of the proof are relegated to Section B.4.  $\square$

There are four parameters that control  $Var(\Delta SUVr)$ :  $\sigma_\epsilon$ ,  $\sigma_Z$ ,  $\rho$ , and  $CV_R$ .

$CV_R$  is the coefficient of variation of the reference ROI, which measures the group-level variation of the reference ROI: high  $CV_R$  indicates high patient-to-patient reference variation. For SUVr, the ideal scenario would be all patients have identical reference uptake (perfect reference), i.e  $CV_R = 0$ . From Proposition 3.5.1, under  $CV_R = 0$ ,  $Var(\Delta SUVr) = \frac{2\sigma_\epsilon^2}{\mu_R^2}$ , and it is apparent that  $\Delta SUVr$  and  $\hat{\Delta}\alpha$  have the same power, since they have identical test statistic. As  $CV_R$  increases, however,  $\Delta SUVr$ 's power decreases as noise starts to add up. This is potentially hard to control, since uptake of  $A\beta$  tracers in the reference is a natural process and cannot be selectively controlled without simultaneously jeopardizing the fidelity of the target ROIs. The empirical  $CV_R$  is about 0.37 for BLAZE, and 0.15 for ADNI, which means we expect  $\Delta$ -measure to work better for BLAZE, and both empirical and simulated results support this insight (Section 3.6.1, 3.6.2).

$\sigma_Z^2$  is the amount of patient-level variability, i.e how different patients are from one-another. This is accounted for explicitly in the  $\Delta$ -measure; SUVr, however, requires external control such as careful patient screening and experimental design to mitigate the potential impact  $\sigma_Z^2$ . According to Eq. 3.7,  $\sigma_Z^2$  has complicated interaction with  $CV_R$  and  $\rho$ . If we make the simplifying assumption that  $\rho = 0.5$ , which is close to the empirical estimates (Table 3.2), then  $\sigma_Z$  contributes to  $Var(\Delta SUVr)$  according to the following relationship:

$$Var(\Delta SUVr) \approx \dots + \sigma_Z^2 \frac{4CV_R^2}{\mu_R^2(1 + CV_R^2)}$$

Empirically  $\sigma_Z$  is 0.25 for BLAZE and 0.1 for ADNI.

$\rho = \text{Cor}(R_1, R_2)$  is the correlation of reference SUV between the first and second scan for a patient. The effect of  $\rho$  on  $\text{Var}(\Delta\text{SUVr})$  takes the following form:

$$\text{Var}(\Delta\text{SUVr}) \approx \dots - 2 \times \underbrace{\frac{1 + 2CV_R^2 + 10\rho CV_R^2}{(1 + 2\rho CV_R^2)^3}}_{\text{multiplier}} \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{\mu_R^2}$$

Where the “multiplier” part is always positive if  $\rho \geq 0$ , thus will always decrease  $\text{Var}(\Delta\text{SUVr})$ , but its magnitude is highly non-linear with respect to  $\rho$  and  $CV_R$ . Empirically  $\rho = 0.55$  for ADNI and  $\rho = 0.32$  for BLAZE.

$\sigma_\epsilon^2$  is statistical error from various other sources such as measurement, registration, and alignment error...etc. This can potentially be improved via better data collection and processing pipeline, such as more accurate instruments, and more sophisticated registration software to achieve more precise alignment.

Among the parameters,  $CV_R$ ,  $\rho$ , and  $\epsilon_Z^2$  do not affect  $\Delta$ -measure because it is specifically designed to side-step these issues, and  $\epsilon^2$  will naturally affects both methods. Table 3.2 lists the empirical estimates for the four crucial parameters mentioned above for ADNI and BLAZE data.

Table 3.2: Estimates of model parameters for BLAZE and ADNI data sets

	$CV_R$	$\sigma_Z$	$\rho$	$\sigma_\epsilon$
ADNI	0.15	0.09	0.55	0.045
BLAZE	0.37	0.27	0.32	0.065

## 3.6 Results

### 3.6.1 Performance comparison on real data

Table 3.3 and 3.4 compares the p-values of detecting  $A\beta$  accumulation in various target regions using SUVr and  $\Delta$ -measure, using subcortical white matter as reference. Testing of  $\Delta$ -measure is done using the parametric approach mentioned in Section 3.4.2. Since both data sets consists of AD positive patients, we assume  $A\beta$  accumulation, as suggested by the Amyloid Hypothesis, although this cannot be verified. One can see that for BLAZE data, both methods are in agreement in state of accumulation.  $\Delta$ -measure is generally be more sensitive to detecting an increase in the target signal from baseline compared to  $\Delta\text{SUVr}$  by having smaller p-values in 9 out of the 11 target ROIS (Table 3.3, disregarding the counter-intuitive result from target ROI “caudate”). However, for the ADNI cohort, this

observation is not recapitulated (Table 3.4). This likely due to BLAZE having higher group level variability ( $CV_R^2$ ) and patient effect ( $\sigma_Z^2$ ) per Eq. 3.7, both increases the variance (thus decreases the power) of  $\Delta SUVr$ .

Table 3.3: Performance comparison between  $\Delta$ -measure and  $\Delta SUVr$  using BLAZE data

Tar ROI	$\Delta\alpha$	$\Delta\bar{S}$	$\Delta\alpha$ pval	$\Delta\bar{S}$ pval
frontal	0.0184	0.0091	0.0006	0.0091
post cingulum	0.0188	0.0134	0.0245	0.0530
parietal	0.0139	0.0084	0.1950	0.1871
lateral temporal	0.0358	0.0193	0.0009	0.0019
medial temporal	0.0248	0.0119	0.0030	0.0104
orbitofrontal	0.0128	0.0040	0.2913	0.6180
occipital	0.0551	0.0292	0.0000	0.0002
anterior cingulum	0.0155	0.0070	0.0518	0.0871
rectus	0.0185	0.0066	0.2796	0.6058
caudate	-0.0180	-0.0123	0.1304	0.1369
putamen	0.0499	0.0272	0.0001	0.0002
thalamus	0.0090	0.0093	0.3211	0.1317

Table 3.3: Comparing the  $p$ -values for testing progression for BLAZE placebo arm using  $\Delta$ -measure vs  $SUVr$  on various target ROIs. Subcortical white matter was used as reference ROI. The columns  $\Delta\alpha$  and  $\Delta\bar{S}$  are the effect size (estimates of progression), and columns “ $\Delta\alpha$  pval” and “ $\Delta\bar{S}$  pval” are the corresponding  $p$ -values.

Table 3.4: Performance comparison between  $\Delta$ -measure and  $\Delta SUVr$  using ADNI data

	$\Delta\alpha$	$\Delta\bar{S}$	$\Delta\alpha$ pval	$\Delta\bar{S}$ pval
frontal	0.0221	0.0163	0.0914	0.0270
cingulate	0.0154	0.0117	0.1933	0.1078
parietal	0.0181	0.0165	0.0826	0.0052
temporal	-0.0048	0.0005	0.6860	0.9377

Table 3.4: Comparing the  $p$ -values for testing progression for ADNI using  $\Delta$ -measure vs that using  $SUVr$  on various target regions using subcortical white matter as reference region. The columns  $\Delta\alpha$  and  $\Delta\bar{S}$  are the estimates of progression, and columns “ $\Delta\alpha$  pval” and “ $\Delta\bar{S}$  pval” are the corresponding  $p$ -values.  $\Delta$ -measure yielded larger  $p$ -values in more target regions, indicating less sensitivity in detecting progression, assuming progression is true. This is likely the result of higher variation in the reference ROI (i.e higher  $CV_R^2$  and  $\sigma_Z^2$ ) in the ADNI data set compared to BLAZE.

### 3.6.2 Power analysis on simulated data

We compared the power of the two methods in detecting treatment effect using parametric bootstrapped simulations by using Eq. 3.5 as the base generating model and adding an additional treatment effect parameter. The generating equation following Eq. 3.10.

$$T_t^i = \alpha_t + \text{txEff}_t \times 1_{\{\text{patient } i \text{ is treated}\}} + \beta R_t^i + Z^i + \epsilon_t^i \quad (3.10)$$

We generate data for 15 placebo and 30 treatment patients, in accordance with our sample size in BLAZE Phase II trial. We set  $\alpha_{t_1} = 0.02$  and  $\alpha_{t_2} = 0.05$ , and impose a treatment effect of -0.02.  $\beta$  at 0.8, similar to the empirical value of the same parameter. We draw  $z_i$  independently from  $N(0, \sigma_z^2)$  distribution, and similarly,  $\epsilon_t^i$  are drawn independently from  $N(0, \sigma_\epsilon^2)$  distribution, where  $\sigma_\epsilon^2/\sigma_z^2$  are set at various values to ascertain their effect on the power of the two methods (see Results). Target SUVs are simulated by supplying bootstrapped reference SUVs into Eq. 3.10. Two simulations are produced, one using ADNI, and the other using BLAZE.

Figure 3.8 shows the power curves when the reference SUVs are sampled from BLAZE and ADNI respectively. There are drastic differences in the behaviors of the power curves.

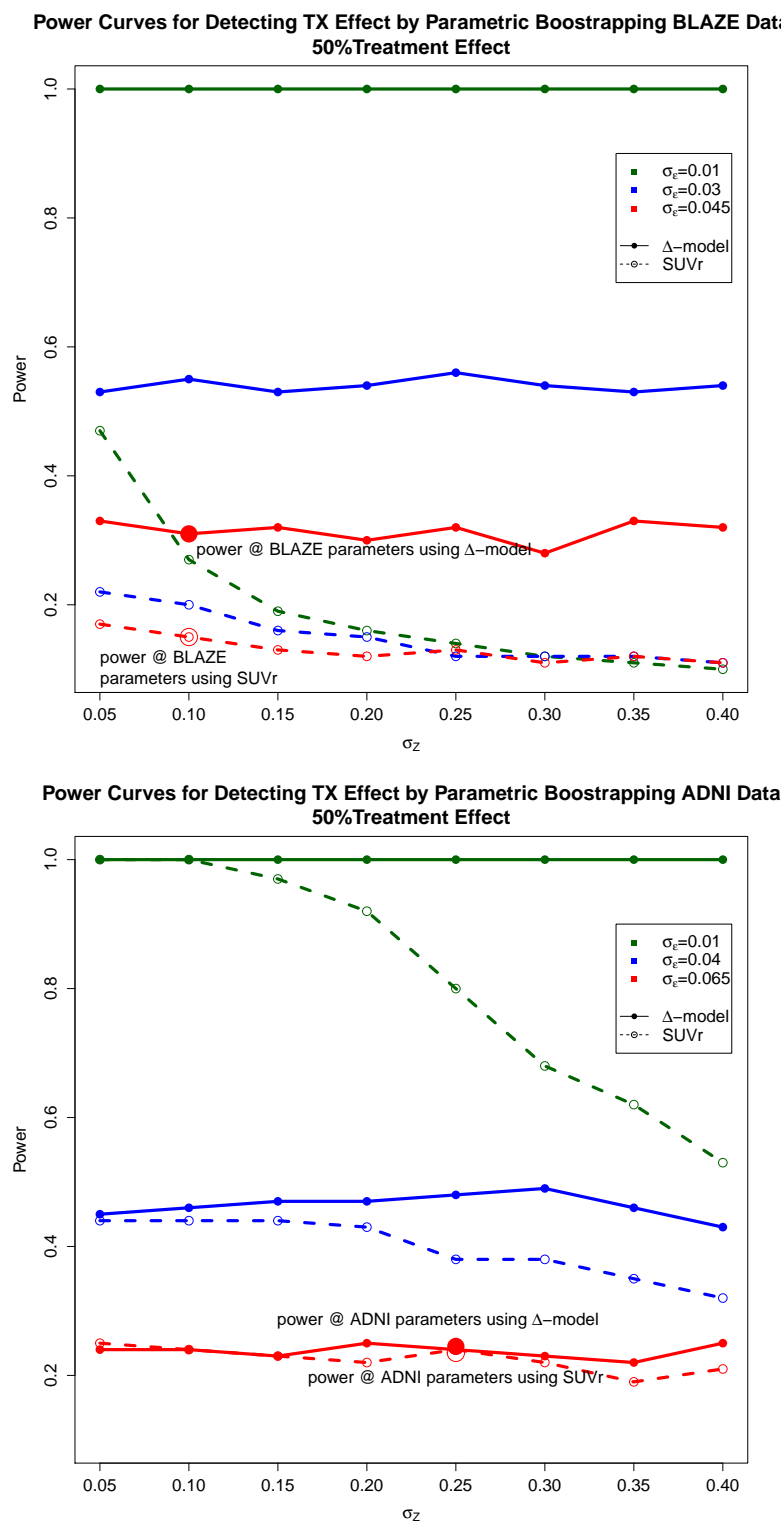
In BLAZE, the advantage of  $\Delta$ -measure is clear: the power curves of  $\Delta$ -measure is markedly above that of SUVr at all configurations. Most notably, under the empirical setting of the BLAZE data, there is a substantial power difference between the two approaches.

$\Delta$ -measure's advantage seen in BLAZE simulation, however, does not appear in ADNI simulation. The power curve diverges only when  $\sigma_\epsilon$  is small and  $\sigma_z$  is high; and when  $\sigma_\epsilon$  is relative large at 0.065, as is seen in the ADNI data, the power curves do not diverge. This echoes our theoretical understanding (Section 3.5.2) as well as empirical observations, where  $\Delta$ -achieves consistently achieves lower p-values in BLAZE, but not in ADNI (Table 3.3 vs Table 3.4).

## 3.7 Conclusion

In our exploratory analysis (Section 3.3), we discovered that target and reference in SUVs exhibited clear linear relationship that persists across time, target-reference-ROI combinations, as well as processing pipelines, which is evidence that this relationship is likely to be biologically meaningful. Based on this linear relationship, we proposed an alternate measure to SUVr, dubbed the  $\Delta$ -measure, for gauging changes in specific binding signal (Section 3.4.2).

To study the statistical behaviors of the two measures, we first proposed a reasonable linear-mixed-effect-based generating model for generating target SUVs, and offered empirical justification for some modeling assumptions (Eq. 3.5, Section 3.5.1). We then used the

Figure 3.8: Power curve comparison between  $\Delta$ -measure and  $\Delta SUVr$  using simulated data

proposed model as the statistical backdrop and studied and commented on the effect of the model parameters on the statistical behavior of the two measures (Section 3.5.2).

To compare the performance of the two measures we used both real data as well as simulation. Comparison of the two measures on ADNI and BLAZE data showed  $\Delta$ -measure to be more sensitive for detecting accumulation under BLAZE data, but not in ADNI. We attribute this performance discrepancy by using the proposed generating model as basis of analysis and identifying two model parameters ( $\sigma_Z$  and  $CV_R$ ) that had markedly different behaviors between the two data sets (Section 3.6).

We then provided simulation-based power comparisons using parametric bootstrapped data (Section 3.6.2). In the power comparison, we see that  $\Delta$ -measure outperforms SUVR by a considerable margin under BLAZE-inspired data, but does not offer significant advantage under ADNI-inspired data. This observation recapitulates the performance discrepancy in the real data comparison.

From both theoretical and empirical investigations, we see that  $\Delta$ -measure serves as a suitable alternative to SUVR. Because  $\Delta$ -measure is rooted in linear regression, it offers a much more flexible framework to 1) incorporate predictors such as age, gender, cognitive scores, and 2) simultaneously evaluate treatment and progression at multiple time points. A similar exercise would be considerably more difficult using the SUVR approach. This particular aspect of the model was not explored in this work due to the lack sufficient patient-level meta data, it is nonetheless a promising investigative direction and a natural application for the  $\Delta$ -measure.

## Chapter 4

# Data-driven discovery of organogenesis master regulator candidates for *D. melanogaster* and *C. elegans*

### 4.1 Introduction

Organogenesis, or the formation of organs, is an exciting field in developmental biology as its advancements requires bringing together advancements in a variety of fields of studies such as cell biology, molecular biology, genetics, and biostatistics. Recent investigations into organogenesis of many model organisms suggest that cell fate could be highly influenced by a dynamic network of small but potent “master regulators”, the absent of which may lead to ectopic- or non-development. For instance, the transcription factor eyeless (*ey*) was named the master regulator of *Drosophila* eye[7], and further evidence suggested that dachshund (*dac*), eye absent (*eya*), and sine oculis (*so*) work with *ey* to form a regulatory network that determines the development of the organ [10, 56]. Worm pharynx is another such organ system whose genesis and maintenance is organized by a handful of transcription factors [48]. Assuming these findings generalize, i.e master regulators exists for all organisms for all organs, then it would be very useful to our understanding of system biology to identify these master regulators.

Data that directly link TFs to organ systems are rare, because TFs influence organ development *indirectly* by regulating genes that drive organ development via cis-regulatory modules (CRMs). Impact of TFs on organ system, however, can still be studied through intermediate datasets that are often prevalent and accessible. For instance, TF binding site data can be used to infer regulatory activity between TF and genes; gene-organ relationship, in turn, can be obtained via large scale projects such as in-situ database curated by the

Celniker lab [78]. Since these studies are often well-established and comprehensive, we can leverage information provided by these data sources to mine potential tissue-specific master regulators. This work outlines a data-driven mining approach to uncovering high-fidelity organ-specific candidate regulators by taking advantage of such public datasets. The immediate value of the work is providing dry-lab guidance to wet-lab experimentation, which is expensive and labor intensive yet remains the definitive way of validating a master regulator.

Machine learning techniques have been applied to master regulator analysis with some success. For instance, Lefebvre et al. [43] introduced MARINa that performs master regulator discovery using a inferred regulatory map between TFs and regulons (activated or repressed targets) as well as the relevant gene expressions profile between contrasting cell states as inputs. Gevaert and Plevritis [25] introduced a three step discovery algorithm in which a generalized linear model is used to identify potential TF regulators, and a elastic-net based algorithm is used to construct TF-gene regulatory network, and finally master regulators and their downstream targets are identified using module network analysis. A more recent work by Sikdar and Datta [71] introduced a two-step hypothesis-testing-based algorithm in which the inferred regulatory modules of a TF between treatment and control groups are compared and the TF with the highest Kendall statistic is crowned the master regulator. An issue these algorithms share in common is that the conclusions drawn depend heavily on the dependability of the regulatory network inferred or taken as input by the algorithms. This is potential problematic, especially when regulatory network itself is still a field of active research, and often lack concrete experimental evidence supporting their reliability. The hypothesis-testing-based algorithm [71] also has makes a debatable assumption that the master regulator is at the top of the regulatory hierarchy, especially when TFs often achieve gene regulation by collaborating in a regulatory network [8]. The algorithm we present is agnostic to the TF network structure, thus side-stepping aforementioned issues.

The text is organized as follows. In Section 4.2 we present data acquisition and processing for *D. melanogaster* and *C. elegans*. Section 4.3 will detail the mining algorithm. Specifically, section 4.3.4 covers how to extract and select candidate TFs for the two organisms. Section 4.4 will comment on on the quality of the TFs obtained.

## 4.2 Data

As mentioned previously, TFs affect tissue development indirectly by influencing genetic drivers of tissue development, and our objective is to elicit the TFs that drive the development process of a particular tissue. The framing of the problem is reminiscent of the problem of feature selection in machine learning, where the goal is to ascertain the few features from a collection that are the top drivers of some underlying process. Our objective can be translated into such a machine learning problem by leveraging the following three categories of information:



1. Genomic location of genes. For model organisms this is generally curated by labs across the world and are publicly available. The datasets we use in this work are obtained from Wormbase and Flybase for *C. elegans* and *D. melanogaster*, respectively. We hereafter refer to this category of information as **Gene-Location** data.
2. Data related to activation of genes in specific organs of interest. There are a growing number of large scaled efforts trying to curate tissue-specific gene profiles for many model organisms, e.g *Drosophila* [78], human [74], multi-organism [54]. We will use Celniker lab’s (LBL) in-situ database [78] for *D. melanogaster* and Waterston lab’s FACS RNA-Seq data for *C. elegans* (Warner and Waterston, in preparation). We hereafter refer to this category of information as **Gene-Organ** data.
3. Genomic locations of TF binding site. Numerous large scale databases are available (e.g Chip-base [86]), however, the TF binding information we use in this work are obtained from experiments of collaborator laboratories. We hereafter refer to this category of information as **TF-Location** data.

**Gene-Organ** data will be used to construct the response  $Y_{n \times 1}$ , where  $n$  = number of genes. Suppose our organ of interest is the nervous system, then we will assign  $Y_i = 1$  if gene  $i$  is associated with nervous system according to the gene-organ database and 0 otherwise. The **Gene-Location** and **TF-location** data will be used to construct the feature matrix  $X_{n \times m}$ , where  $X_{i,j}$  denotes pseudo-measures of the interaction/regulation strength between gene  $i$  and TF  $j$ . Section 4.3 will describe construction details.

We will now detail the descriptions of databases used for *D. melanogaster* and *C. elegans* in Section 4.2.1 and 4.2.2, respectively.

### 4.2.1 *D. melanogaster*

The **Gene-Organ** dataset we use is the in-situ database curated by the Celniker lab in Lawrence Berkeley National Laboratory [78, 77, 29, 59]. As the name suggests, in-situ hybridization [22] is the main technique applied. In this data, gene specific anti-sense florescent riboprobes are manufactured and hybridize into fixed *Drosophila* embryo such that florescence microscopy can be used to determine gene’s physical location in a developing embryo. Figure 4.1 shows an example where the gene *Ptx1* lights up in areas that are associated with development of Malpighian tubules. Based on the physical location of gene expression, embryologists use expert knowledge about embryonic cell fate and annotate the gene-tissue association using controlled vocabulary. At the time of writing, expert annotation of 7,921 genes are available in the in-situ database [59]; however, because not all genes conform with labeling in the reference genome, some genes were forced to be left out, resulting in only 6,056 being used in the final construction. In this work we will look at gut and nervous system.



Figure 4.1: *Example of Ptx1 gene imaged during Stage 11 of D. melanogaster embryonic development*

The **TF-Location** data is obtained from ChIP-chip experiments conducted in [72]. In this study, two replicate experiments are done for each TF. For a particular TF, replicates are passed through MACS2 separately [87] with a p-value threshold of  $10^{-3}$  to obtain two sets of potential peaks. Then replicates are then pooled and MACS2 is called again with p-value threshold of  $10^{-3}$  on the pooled experiments. Peaks from individual experiments are only retained if they also show up in the pooled peak set. The retained peaks from individual experiments are then ranked according to their p-value, producing two sets of ranked peaks, which are input into the IDR framework [45] to assess reproducibility of the peaks. Peaks are retained if they meet the IDR threshold of 0.05. All TF peaks in the resulting data are accompanied by a local IDR value, which will be used in subsequent analysis as a surrogate for regulation strength between TF and genes near the genomic region.

The **Gene-Location** data used is the UCSC fly genome v19 (GRCh37.p13) [38, 64].

### 4.2.2 C. elegans

The **Gene-Organ** data is the fluorescence-activated cell sorting (FACS) RNA-seq data provided by the Waterston Lab as the gene-organ data (Warner and Waterston, in preparation). In each experiment, a specific tissue cell type (e.g muscle) from the FACS process will go through RNA-Seq, which returns tissue-specific expression levels of genes. Expression levels are measured in depth coverage per million reads (DCPM), and the cells from muscle,

hypodermis, nervous system, pharynx, intestine are analyzed.

**TF-Location** data we use is the uniform IDR thresholded peak calls for *C.elegans* mod-ENCODE TF ChIP-seq data [4]. In short, the TF peaks from ChIP-seq experiments are called by SPP using an FDR of 0.9. The artificially high FDR threshold allows both signal and noise to pass through, where by peaks from replicates are then passed through the IDR protocol (threshold of 0.05 was used) and sorted into signal and noise components to determine the irreducible discovery rate of each peak location. The resulting peaks that passes the IDR step are pooled across replicates and SPP is called again with the relaxed FDR threshold of 0.9. A final output consists of peak location and the FDR of each location outputted by SPP. See [4] for detailed description of the data generation and peak calling procedures.

**Gene-Location** data is obtained from WormBase release WS248 [82].

### 4.3 Methods: constructing a learning problem

We will combine the three data types outlined in Section 4.2 into a classification problem. Classification problems are often of the form

$$Y = f(X) + \epsilon$$

Where  $Y$  denotes a  $n \times 1$  response vector, often binary.  $X$  is the  $n \times m$  predictor matrix,  $f$  is some predictor function that we want to fit, and  $\epsilon$  captures the error. Sections 4.3.1 and 4.3.2 will detail the construction of  $Y$  and  $X$ , respectively; Section 4.3.3 will outline the model ( $f$ ) we are going to use, and Section 4.3.4 will tie the ideas together and present a method of selecting the master regulator candidates.

#### 4.3.1 Construction of the response vector

The learning problem will be tissue specific, i.e, we need separate setups for each tissue type. The response vector will be binary of length  $n$ , with 1 indicating the gene is associated with the tissue type, i.e it is present in the embryonic regions associated with the tissue.

For *D. melanogaster*, first collect all controlled vocabulary related to the tissue of interest. Then, for each gene, search through its annotations for the tissue-specific vocabulary. For instance, all terms associated with nervous system will contain “nerv” (e.g nervous, nerve), “cns” (central nervous system), so for each gene, we will look at all its expert-annotated terms and pattern match for “nerv” and “cns”. If pattern match exists, then we will mark 1 (i.e associated) in the response vector corresponding to this gene, and 0 other wise.

For *C. elegans*, a gene is deemed to be associated with a particular tissue if its DCPM for a tissue is 10 folds higher than the average of other tissues. The threshold of 10 was chosen to be conservative, but analysis show that this particular threshold produces reasonable results

(Section 4.4).

### 4.3.2 Construction of the predictor matrix

We use the **Gene-Location** and **TF-Location** data to construct feature matrix  $X_{n \times m}$ , with  $X_{i,j}$  being a pseudo-measure of the intensity in which TF  $j$  regulates gene  $i$ . Peak calls from both *D. melanogaster* and *C. elegans* data come with significance measure for the peaks, e.g for *D. melanogaster* it's the local IDR, and for *C. elegans* the FDR. We use negative log of the significance measure as a pseudo-measure for the strength of regulation. For gene  $i$  and TF  $j$ , we first find the genomic locations of all of gene  $i$ 's transcripts, then record all TF  $j$ 's peaks that fall within 1kb up or down stream from TSS's of gene  $i$ 's transcripts. The 2kbps region around TSS site was chosen because it was observed that many factors display upstream or downstream activity within a 2kbp span covering the TSS [2]. We then sum up the negative log value of unique peaks, and the resulting value serves as a pseudo-measure gauging the strength in which TF  $j$  regulates gene  $i$ . Figure 4.2 offers a schematic. The resulting predictor matrix will be referred to as  $X$ . For *Drosophila*,  $X$  is  $6,056 \times 199$  and for worm  $20,426 \times 252$ .

With this construction, we make the important simplifying assumption that proximity of a transcription factor binding site is indicative of regulation. This, however, does not hold true due to the presence of *trans* regulation. Therefore, the  $X$  matrix in the learning problem can be refined using this information. Another way to refine the input data is to consider cross species TF binding site information, as it was shown that cross-species comparison greatly improves TF occupancy prediction [37]. More over, currently only static binding information considered, however it can was shown that TF residence time as opposed to steady state binding information is a better indicator of regulation [75].

### 4.3.3 Balanced Random Forest

There are myriad of approaches for solving binary classification problem, we chose random forest because it has proven to be successful in handling cases where relationship between predictors and response may not be linear, as in our case. However, numerous studies have shown that class size imbalance adversely affect tree-based methods like random forest [42], thus some form of resampling is needed to improve predictive power. We down sample the larger population so that the final response vector contains the same number of 0's and 1's, producing a balanced learning problem. Random forest will assign each feature a rank according to variable importance (measured by % increase in MSE<sup>1</sup>). We run 20 independent trials of the learning problem so that each feature will have 20 ranks from each of the 20

---

<sup>1</sup>Produced by `importance(..., type=1)` from the `randomForest` R package

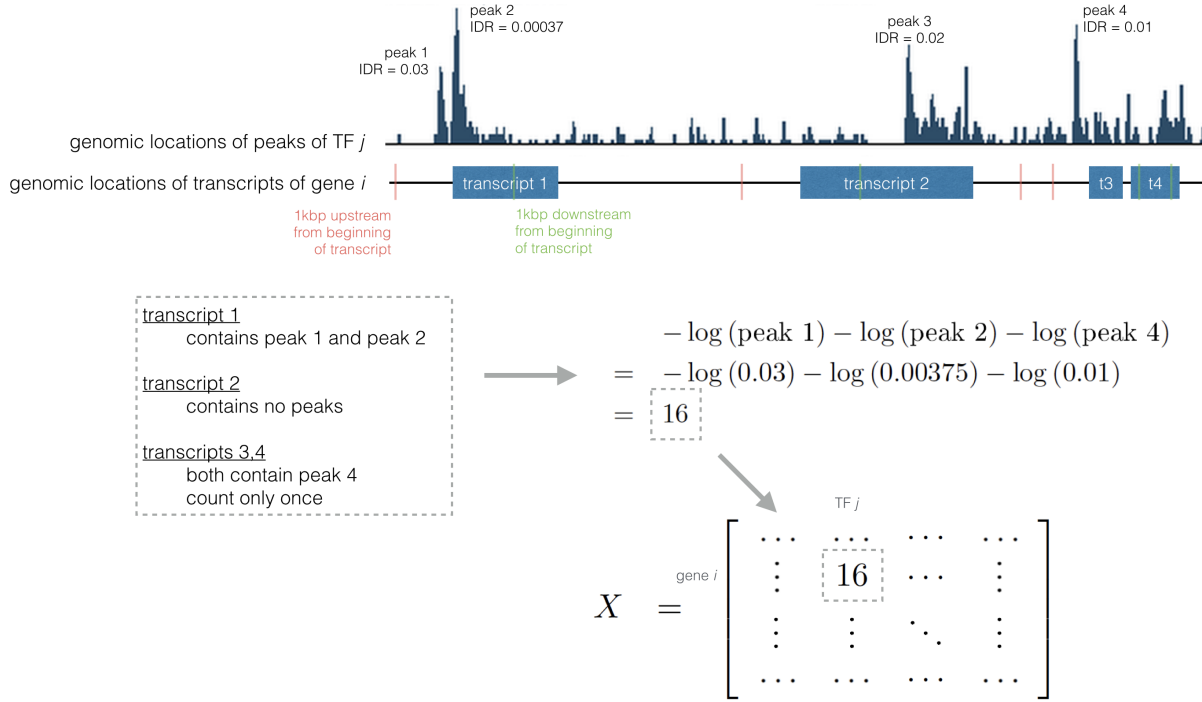


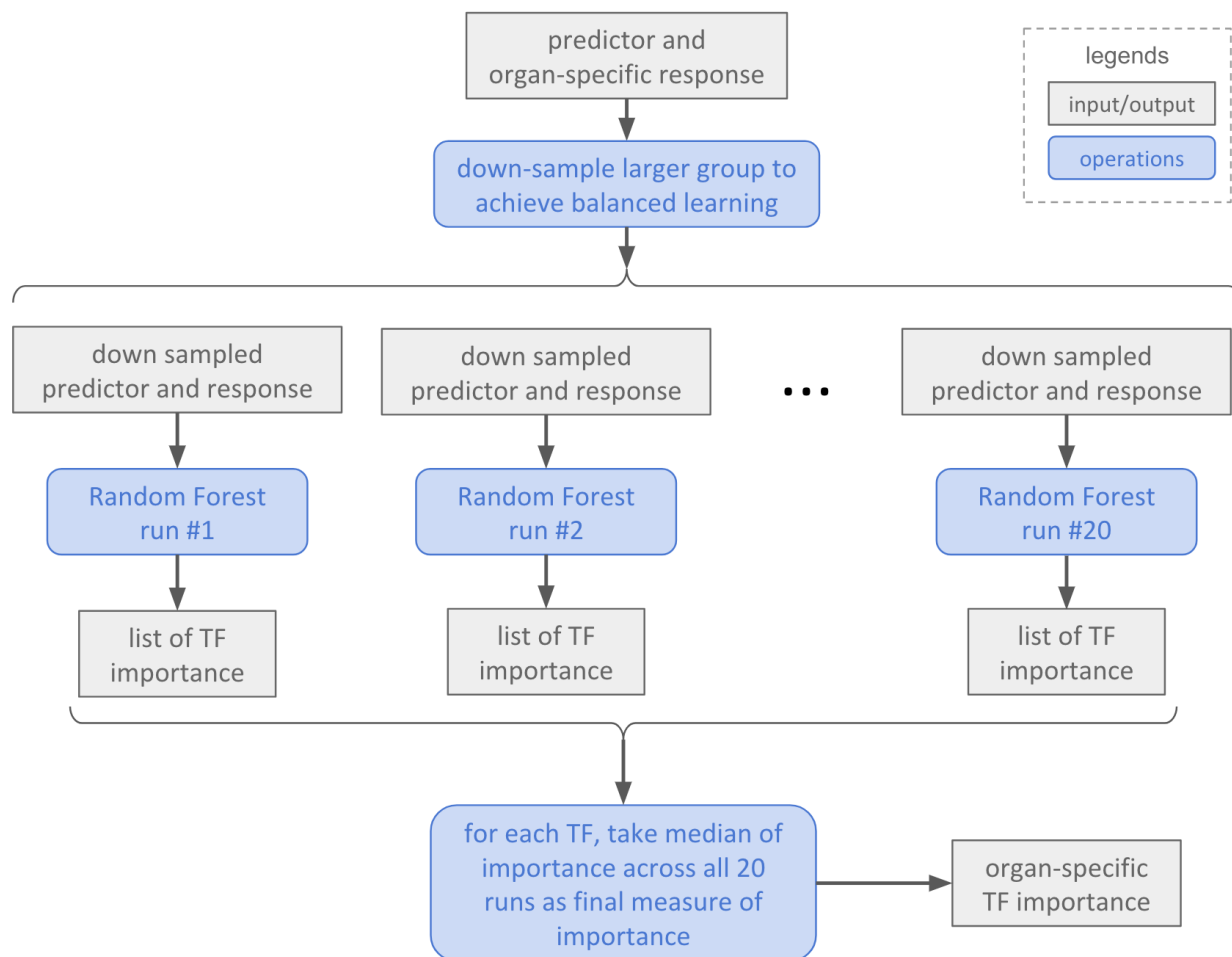
Figure 4.2: Example calculation for filling in one cell of the predictor matrix.

trials, and the median of the 20 ranks is taken as the overall measure of importance. Figure 4.3 demonstrates the flow of the algorithm.

#### 4.3.4 Selecting significant TFs

From previous step, we obtain a ranked list of TFs for each tissue type. A TF may be important for multiple organ systems, such as TFs that regulate housekeeping genes; our objectives, however, is to detect *organ-specific* TFs. To solve this problem, we measure the importance of a TF to a specific organ by its relative importance to other tissue types.

For *D. melanogaster*, we have two organ systems of interest—nervous system and gut, we use the difference in logged rank difference to score the TF's relative importance between organ systems. The advantage of logged rank difference is that a same gap higher in the list has a much bigger effect of the same rank difference lower on the list. For instance, if TF  $i$  is ranked 3 in gut, and ranked 23 in nervous system, their ranked difference is  $\log 3 - \log 23 = -2.03$ ; however, the rank difference between 23 and 43, also a rank difference of 20 nominal ranks, is only  $\log 23 - \log 43 = -0.63$ ; in fact, if TF  $i$  is ranked 23rd, TF  $j$  would need to be ranked as low as 175th to achieve the same score of -2.03.

Figure 4.3: *Flowchart for repeated balanced random forest runs.*

This approach can be generalized to comparing more than two organs. The *C. Elegans* data, for instance, has five tissue types. In this case, we split the tissue types into two groups, say {pharynx, nervous system} and {muscle, hypodermis, and intestine}, and calculate the logged ranked difference for the mean of the two groups. We do this for all  $5 + \binom{5}{2} = 15$  different splits, and take the split with the highest differential. We then assign importance according to the absolute values of the rank differentials.

This approach only reports the differentials, and does not give a cut-off to select the specific TFs. The choice of threshold is left to the user's discretion, and for this

TF	Muscles	Hypodermis	Nervous System	Intestine	Pharynx	Differential
hlh-1 <sub>EM</sub>	<b>1</b>	38	26	30	47.5	3.57
pqm-1 <sub>L4</sub>	25.5	17	37	16.5	<b>1</b>	3.18
RPC-15 <sub>YA</sub>	44	<b>15.5</b>	42	<b>17.5</b>	39	0.93

Table 4.1: Median importance factor returned by balanced random forest on each of the five organ systems. For each TF (row), the number indicates the median importance score for each tissue, and color indicate the grouping that generated the highest differential.

## 4.4 Results

The most reliable way of validating whether a TF is a organogenesis regulator is to perform RNAi knockout experiments. This option was not available at the time of the writing, so experimentally verified GO terms will be used as the primary evidence of TF’s involvement in a specific organ. Unless mentioned specifically, GO terms are derived from experimental evidence<sup>2</sup>. AUC are approximately 0.65 for fly and 0.7 for worm, which are on the low side in terms of predictive power, but this should not invalidate the TFs selected, since at its heart this is a feature selection problem, and features can still be important without the overall model having high predicative power. The reason for the low predicative power of the model is likely due to the dependence structure between the TFs not sufficiently captured since

1. Not all TFs binding information is available (e.g only 199 of the supposed 708 TFs [30] were available for fly data), and
2. Regulation of gene expression often involves interaction between TFs to ensure robust expression [75], and this inter-dependence is not fully captured in the design matrix.

Despite these caveats, one can see that the TFs recovered are sensible.

### 4.4.1 D. melanogaster

One can see that the majority of the TFs selected as master regulators for the nervous system are associated with nervous GO terms. It is reassuring that *Lola*, a well known regulator for fly nervous system development, shows up as the top nervous system TF.

The gut organ is a bit more ambiguous. None of the enriched GO terms are directly related to gut. There are various TFs with terms dealing with organs that may be tangential to gut development, such as muscle, epidermis, pharynx, and trachea (*grh*, *h*, *vri*, respectively). There are a few top gut TFs such as *Max*, *dsx* that have very few GO terms defined. It should also be noted that among the 2056 out of the 6506 genes labeled to be gut associated in the in-situ dataset, only 44 had experimentally verified gut GO terms, yet among the 1230 (out of 6506) genes labeled nerve-related, 254 have experimentally verified

<sup>2</sup>experimental evidence with codes IMP, IEP, EXP, IGI, IDA, or IPI

Table 4.2: Top Discovered TFs that may be driving gut development in *D. melanogaster*

TF	Gut Rank	NS Rank	Differential	Relevant GO Terms
dm	3	20	1.90	
Max	2	12	1.79	
grn	1	3	1.10	organ morphogenesis tissue development
br	8	18	0.81	muscle fiber development
Neu2	57	133	0.68	
grh	33	60	0.60	epithelial cell morphogenesis
CG7045	22	39	0.57	
dsx	77	129	0.52	
h	14	23	0.50	cell morphogenesis
cnc	46	74	0.48	pharynx development
EcR	5	8	0.47	Malpighian tubule morphogenesis epidermis development
vri	7	11	0.45	open tracheal system development

Table 4.3: Top Discovered TFs that may be driving nervous system development

TF	NS Rank	Gut Rank	Differential	Relevant GO Terms
Lola	1	11	2.40	neurogenesis dendrite morphogenesis
bab1	5	26	1.65	
Dif	7	18	0.94	peripheral nervous system neuron development
dac	43	80	0.62	neuron differentiation
HmgD	27	47	0.55	dendrite morphogenesis
ftz	10	16	0.47	central nervous system development <sup>3</sup>
eyg	48	76	0.46	
CG12155	41	26	0.46	
CG13624	15	23	0.43	
chn	46	74	0.40	peripheral nervous system development dendrite morphogenesis nervous system development

nervous-system-related GO terms. This shows that either the nervous-system is much better studied and/or there are discrepancies in expert labeling in the in-situ data, both of which would have negatively impacted the efficacy of the random forest approach. The average AUC across 20 balanced random forecast run is 0.65 for both nervous system and gut.



Tissues	Significant TFs	AUC
Muscle	hlh-1, unc-62 , nhr-11	0.76
Hypodermis	blmp-1, elt-1, nhr-23, nhr-25	0.64
Nervous System	ham-1, lys-2	0.81
Pharynx	pha-4	0.60
Intestine	pqm-1, nhr-28	0.72

Table 4.4: *Significant TFs for each organ system for C. elegans among 20 groupings with highest importance differential (12 are shown because repeats are combined). The AUC-ROC is the averaged across 20 runs of balanced random forecast.*

#### 4.4.2 C. elegans

There are abundant experimental evidence associating the top transcription factors to their respective organ systems, except the intestine.

*hlh-1* was found to be a myogenic regulatory factor family transcription factor that converts almost all cells to a muscle-like fate, regardless of their lineage of origin [21]. *unc-62* was also found to be involved in muscle cell differentiation [35].

There are abundant experimental evidence suggesting *pha-4* being one of the central regulator of pharynx development [48]. Gaudet and Mango showed that *pha-4* “specifies organ identity for *C. elegans* pharyngeal cells” [23].

*blmp-1* and *elt-1* have shown to play important roles in epidermis development. *blmp-1* in epidermal cell fate specification [53] and *elt-1* in positive regulation of epidermis development, [32]. Knocking down of *nhr-25* resulted in induction of epidermal infection genes, showing that the transcription factor is an crucial player in epidermis maintenance [80].

Experimental evidence supports *ham-1* being a potential master regulator for nervous system development, being involved in “cell fate determination” [28] and “neuron migration” [15].

However, we could not find relevant literature on the top intestine TFs, a result mirroring that of *D. melanogaster* (Section 4.4.1). We have found that there are only 15 genes with GO terms containing “gut” or “intestine”<sup>4</sup>, compared to 504 for nervous system (“nerv”), 105 for pharynx (“pharyn”), 607 for muscle (“muscle” or “muscular”), and 100 for hypodermis (“hypoderm”). Thus, unless the gut is a much less complex organ than the pharynx, this may be evidence that worm intestine’s regulatory/genetic activity may not be well-studied.

<sup>4</sup>Data from WormMine: <http://intermine.wormbase.org/tools/wormmine/results.do?trail=%257Cquery%257Cresults.0&queryBuilder=true>

## 4.5 Conclusion

In this work we presented a methodology for uncovering organ-specific candidate master regulators by mining large public datasets and formulating the synergistic information therein into a feature selection problem.

We used down-sampled balanced random forest to perform feature selection due to its ability to handle non-linear relationships between features and response, but users are free to use any feature selection method of their choice. The methodology was applied to two model organisms, *D. melanogaster* and *C. elegans*, where nervous system and gut were examined in the former, and hypodermis, muscle, nervous system, pharynx, and intestine in the latter. We uncovered biologically plausible candidates for both organisms, but could not find evidence in literature supporting the discovered regulator candidates for *D. melanogaster* gut or *C. elegans* intestine. Using FlyMine and WormMine services [73], however, we have found that there are very few genes that have GO-terms associated with intestine or gut, indicating that the organ may not be studied as extensively as others.

In constructing the learning problem we have made myriad of assumptions, which should be verified and relaxed when possible. For instance, a crucial assumption we made is that proximity implies regulation, but this does not hold true due to the presence of *trans* regulation. Thus one way to improve the construction of the feature matrix is to incorporate *trans* regulation information.

This work is still preliminary and can be extended in many directions.

As mentioned in the Results section (Section 4.4), experimental validation such as RNAi knock-down would provide more concrete evidence to the discovered master regulator candidates. For instance, *dm* and *br* have shown high potential to be master regulators for fly gut, which can only be verified via wetlab experiments.

There are other downstream fundamental questions that is not answered by this framework, but can be further explored via the results. For instance, co-occurrence of master regulator candidates in random forest’s trees may reveal key organ-specific regulatory networks.

One of the major drawbacks of the proposed method is its inability to handle extremely unbalanced data. For instance, we could not obtain reasonable results for *Drosophila* eye because there were only 20 genes (out of 6,056) associated with the eye, and such imbalanced presented a major hurdle for random forest feature selection. A natural follow-up, then, would be using techniques such as gradient boosted trees, where a popular implementation XGBoost [11] has a parameter<sup>5</sup> specifically designed for imbalanced data.

Data-driven discovery of organ-specific master regulator can serve as useful guidance for expensive wet lab discovery of master regulators by providing high fidelity candidates. We

---

<sup>5</sup>Specifically, `scale_pos_weight`

were able to uncover plausible candidate despite the many simplifying assumptions in the methodology. Therefore we think this approach has the potential to yield more precise and definitive biological insights upon further tuning and refinement.

# Bibliography

- [1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. *On the surprising behavior of distance metrics in high dimensional space*. Springer, 2001.
- [2] Carlos L Araya et al. “Regulatory analysis of the *C. elegans* genome with spatiotemporal resolution”. In: *Nature* 512.7515 (2014), pp. 400–405.
- [3] Alexandre Bejanin et al. “Tau pathology and neurodegeneration contribute to cognitive impairment in Alzheimer’s disease”. In: *Brain* 140.12 (2017), pp. 3286–3300.
- [4] Alan P Boyle et al. “Comparative analysis of regulatory information and circuits across distant species”. In: *Nature* 512.7515 (2014), pp. 453–456.
- [5] Florian Buettner et al. “Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells”. In: *Nature biotechnology* 33.2 (2015), pp. 155–160.
- [6] Alistair Burns and Steve Iliffe. “Alzheimer’s disease”. In: *BMJ* 338 (2009). ISSN: 0959-8138. DOI: 10.1136/bmj.b158.
- [7] Patrick Callaerts, Georg Halder, and Walter J Gehring. “PAX-6 in development and evolution”. In: *Annual review of neuroscience* 20.1 (1997), pp. 483–532.
- [8] Sunny Sun-Kin Chan and Michael Kyba. “What is a master regulator?” In: *Journal of stem cell research & therapy* 3 (2013).
- [9] Kewei Chen et al. “Improved Power for Characterizing Longitudinal Amyloid- $\beta$  PET Changes and Evaluating Amyloid-Modifying Treatments with a Cerebral White Matter Reference Region”. In: *Journal of Nuclear Medicine* 56.4 (2015), pp. 560–566.
- [10] Rui Chen et al. “Dachshund and eyes absent proteins form a complex and function synergistically to induce ectopic eye development in *Drosophila*”. In: *Cell* 91.7 (1997), pp. 893–903.
- [11] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM. 2016, pp. 785–794.
- [12] Yizong Cheng and George M Church. “Biclustering of expression data.” In: *Ismb*. Vol. 8. 2000. 2000, pp. 93–103.

- [13] Ronald Christensen. *Plane answers to complex questions: the theory of linear models*. Springer Science & Business Media, 2011.
- [14] Qiaolin Deng et al. “Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells”. In: *Science* 343.6167 (2014), pp. 193–196.
- [15] Chand Desai et al. “A genetic pathway for the development of the *Caenorhabditis elegans* HSN motor neurons.” In: *Nature* 336.6200 (1988), pp. 638–646.
- [16] Alexander Dobin et al. “STAR: ultrafast universal RNA-seq aligner”. In: *Bioinformatics* 29.1 (2013), pp. 15–21.
- [17] P Edison et al. “Amyloid, hypometabolism, and cognition in Alzheimer disease An [11C] PIB and [18F] FDG PET study”. In: *Neurology* 68.7 (2007), pp. 501–508.
- [18] Eli Eisenberg and Erez Y Levanon. “Human housekeeping genes, revisited”. In: *Trends in Genetics* 29.10 (2013), pp. 569–574.
- [19] UC Santa Cruz ENCODE Consortium Stadford University. *RNA-seq pipeline for long RNAs*. 2016. URL: <https://www.encodeproject.org/rna-seq/long-rnas/> (visited on 04/17/2016).
- [20] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.
- [21] Tetsunari Fukushige and Michael Krause. “The myogenic potency of HLH-1 reveals wide-spread developmental plasticity in early *C. elegans* embryos”. In: *Development* 132.8 (2005), pp. 1795–1805.
- [22] Joseph G Gall and Mary Lou Pardue. “Formation and detection of RNA-DNA hybrid molecules in cytological preparations”. In: *Proceedings of the National Academy of Sciences* 63.2 (1969), pp. 378–383.
- [23] J Gaudet and SE Mango. “Regulation of organogenesis by the *Caenorhabditis elegans* FoxA protein PHA-4”. In: *Science* 295.5556 (2002), pp. 821–825.
- [24] Gad Getz, Erel Levine, and Eytan Domany. “Coupled two-way clustering analysis of gene microarray data”. In: *Proceedings of the National Academy of Sciences* 97.22 (2000), pp. 12079–12084.
- [25] Olivier Gevaert and Sylvia Plevritis. “Identifying master regulators of cancer and their downstream targets by integrating genomic and epigenomic features”. In: *Biocomputing 2013*. World Scientific, 2013, pp. 123–134.
- [26] Alison Goate et al. “Segregation of a missense mutation in the amyloid precursor protein gene with familial Alzheimer’s disease”. In: *Nature* 349.6311 (1991), p. 704.
- [27] Gérard Govaert and Mohamed Nadif. “Block clustering with bernoulli mixture models: Comparison of different approaches”. In: *Computational Statistics & Data Analysis* 52.6 (2008), pp. 3233–3245.

- [28] Catherine Guenther and Gian Garriga. “Asymmetric distribution of the *C. elegans* HAM-1 protein in neuroblasts enables daughter cells to adopt distinct fates”. In: *Development* 122.11 (1996), pp. 3509–3518.
- [29] Ann S Hammonds et al. “Spatial expression of transcription factors in *Drosophila* embryonic organ development”. In: *Genome Biol* 14.12 (2013), R140.
- [30] Ann S Hammonds et al. “Spatial expression of transcription factors in *Drosophila* embryonic organ development”. In: *Genome biology* 14.12 (2013), R140.
- [31] Derek LG Hill et al. “Medical image registration”. In: *Physics in medicine & biology* 46.3 (2001), R1.
- [32] Moritz Horn et al. “DRE-1/FBXO11-dependent degradation of BLMP-1/BLIMP-1 governs *C. elegans* developmental timing and maturation”. In: *Developmental cell* 28.6 (2014), pp. 697–710.
- [33] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of classification* 2.1 (1985), pp. 193–218.
- [34] Lan Jiang et al. “GiniClust: detecting rare cell types from single-cell gene expression data with Gini index”. In: *Genome Biology* 17.1 (2016), p. 144.
- [35] Yuan Jiang, Herong Shi, and Jun Liu. “Two Hox cofactors, the Meis/Hth homolog UNC-62 and the Pbx/Exd homolog CEH-20, function together during *C. elegans* postembryonic mesodermal development”. In: *Developmental biology* 334.2 (2009), pp. 535–546.
- [36] Fuyuki Kametani and Masato Hasegawa. “Reconsideration of Amyloid Hypothesis and Tau Hypothesis in Alzheimer’s Disease”. In: *Frontiers in neuroscience* 12 (2018), p. 25.
- [37] Majid Kazemian et al. “Quantitative analysis of the *Drosophila* segmentation regulatory network using pattern generating potentials”. In: *PLoS biology* 8.8 (2010), e1000456.
- [38] W James Kent et al. “The human genome browser at UCSC”. In: *Genome research* 12.6 (2002), pp. 996–1006.
- [39] Susan M Landau et al. “Measurement of Longitudinal  $\beta$ -Amyloid Change with 18F-Florbetapir PET and Standardized Uptake Value Ratios”. In: *Journal of Nuclear Medicine* 56.4 (2015), pp. 567–574.
- [40] Marc Laruelle, Mark Slifstein, and Yiyun Huang. “Positron emission tomography: imaging and quantification of neurotransmitter availability”. In: *Methods* 27.3 (2002), pp. 287–299.
- [41] Laura Lazzeroni and Art Owen. “Plaid models for gene expression data”. In: *Statistica sinica* (2002), pp. 61–86.
- [42] Paul H Lee. “Resampling methods improve the predictive power of modeling in class-imbalanced datasets”. In: *International journal of environmental research and public health* 11.9 (2014), pp. 9776–9789.

- [43] Celine Lefebvre et al. “A human B-cell interactome identifies MYB and FOXM1 as master regulators of proliferation in germinal centers”. In: *Molecular systems biology* 6.1 (2010), p. 377.
- [44] Bo Li and Colin N Dewey. “RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome”. In: *BMC bioinformatics* 12.1 (2011), p. 1.
- [45] Qunhua Li et al. “Measuring reproducibility of high-throughput experiments”. In: *The annals of applied statistics* (2011), pp. 1752–1779.
- [46] Yi Li et al. “Regional analysis of FDG and PIB-PET images in normal aging, mild cognitive impairment, and Alzheimer’s disease”. In: *European journal of nuclear medicine and molecular imaging* 35.12 (2008), pp. 2169–2181.
- [47] Sara C Madeira and Arlindo L Oliveira. “Biclustering algorithms for biological data analysis: a survey”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 1.1 (2004), pp. 24–45.
- [48] Susan E Mango. “The *C. elegans* pharynx: a model for organogenesis”. In: (2007).
- [49] Glenn W Milligan and Martha C Cooper. “A study of the comparability of external criteria for hierarchical cluster analysis”. In: *Multivariate Behavioral Research* 21.4 (1986), pp. 441–458.
- [50] E Mohandas, V Rajmohan, and B Raghunath. “Neurobiology of Alzheimer’s disease”. In: *Indian journal of psychiatry* 51.1 (2009), p. 55.
- [51] Tal Nawy. “Single-cell sequencing”. In: *Nature methods* 11.1 (2014), pp. 18–18.
- [52] Nobuyuki Okamura and Kazuhiko Yanai. “Brain imaging: applications of tau PET imaging”. In: *Nature Reviews Neurology* 13.4 (2017), p. 197.
- [53] Barbara D Page et al. “ELT-1, a GATA-like transcription factor, is required for epidermal cell fates in *Caenorhabditis elegans* embryos.” In: *Genes & development* 11.13 (1997), pp. 1651–1661.
- [54] Jian-Bo Pan et al. “PaGenBase: a pattern gene database for the global and dynamic understanding of gene function”. In: *PloS one* 8.12 (2013), e80747.
- [55] Simone Picelli et al. “Smart-seq2 for sensitive full-length transcriptome profiling in single cells”. In: *Nature methods* 10.11 (2013), p. 1096.
- [56] Francesca Pignoni et al. “The eye-specification proteins So and Eya form a complex and regulate multiple steps in *Drosophila* eye development”. In: *Cell* 91.7 (1997), pp. 881–891.
- [57] Pascal Pons and Matthieu Latapy. “Computing communities in large networks using random walks”. In: *Computer and Information Sciences-ISCIS 2005*. Springer, 2005, pp. 284–293.
- [58] Ashley A Powell et al. “Single cell profiling of circulating tumor cells: transcriptional heterogeneity and diversity from breast cancer cell lines”. In: *PloS one* 7.5 (2012), e33788.

- [59] Berkeley Drosophila Genome Project. *Patterns of gene expression in Drosophila embryogenesis*. <http://insitu.fruitfly.org/cgi-bin/ex/insitu.pl>. Online; accessed 29 January 2014. 2013.
- [60] Daniel Ramsköld et al. “Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells”. In: *Nature biotechnology* 30.8 (2012), pp. 777–782.
- [61] William M Rand. “Objective criteria for the evaluation of clustering methods”. In: *Journal of the American Statistical association* 66.336 (1971), pp. 846–850.
- [62] Jorge S Reis-Filho. “Next-generation sequencing”. In: *Breast Cancer Research* 11.3 (2009), S12.
- [63] Jason A Reuter, Damek V Spacek, and Michael P Snyder. “High-throughput sequencing technologies”. In: *Molecular cell* 58.4 (2015), pp. 586–597.
- [64] Kate R Rosenbloom et al. “The UCSC genome browser database: 2015 update”. In: *Nucleic acids research* 43.D1 (2015), pp. D670–D681.
- [65] Jorge M Santos and Mark Embrechts. “On the use of the adjusted rand index as a metric for evaluating supervised classification”. In: *Artificial neural networks–ICANN 2009*. Springer, 2009, pp. 175–184.
- [66] Gerard D Schellenberg et al. “Genetic linkage evidence for a familial Alzheimer’s disease locus on chromosome 14”. In: *Science* 258.5082 (1992), pp. 668–671.
- [67] Christopher G Schwarz et al. “Optimizing PiB-PET SUVR change-over-time measurement by a large-scale analysis of longitudinal reliability, plausibility, separability, and correlation with MMSE”. In: *Neuroimage* 144 (2017), pp. 113–127.
- [68] Dennis J Selkoe and John Hardy. “The amyloid hypothesis of Alzheimer’s disease at 25 years”. In: *EMBO molecular medicine* 8.6 (2016), pp. 595–608.
- [69] Funan Shi and Haiyan Huang. “Identifying Cell Subpopulations and Their Genetic Drivers from Single-Cell RNA-Seq Data Using a Biclustering Approach”. In: *Journal of Computational Biology* 24.7 (2017), pp. 663–674.
- [70] Sepideh Shokouhi et al. “Reference tissue normalization in longitudinal 18 F-florbetapir positron emission tomography of late mild cognitive impairment”. In: *Alzheimer’s research & therapy* 8.1 (2016), p. 2.
- [71] Sinjini Sikdar and Susmita Datta. “A novel statistical approach for identification of the master regulator transcription factor”. In: *BMC bioinformatics* 18.1 (2017), p. 79.
- [72] Matthew Slattery et al. “Diverse patterns of genomic targeting by transcriptional regulators in *Drosophila melanogaster*”. In: *Genome research* 24.7 (2014), pp. 1224–1235.
- [73] Richard N Smith et al. “InterMine: a flexible data warehouse system for the integration and analysis of heterogeneous biological data”. In: *Bioinformatics* 28.23 (2012), pp. 3163–3165.



- [74] Chang Gue Son et al. “Database of mRNA gene expression profiles of multiple human organs”. In: *Genome research* 15.3 (2005), pp. 443–450.
- [75] François Spitz and Eileen EM Furlong. “Transcription factors: from enhancer binding to developmental control”. In: *Nature reviews. Genetics* 13.9 (2012), p. 613.
- [76] Warren J Strittmatter et al. “Apolipoprotein E: high-avidity binding to beta-amyloid and increased frequency of type 4 allele in late-onset familial Alzheimer disease.” In: *Proceedings of the National Academy of Sciences* 90.5 (1993), pp. 1977–1981.
- [77] Pavel Tomancak et al. “Global analysis of patterns of gene expression during Drosophila embryogenesis”. In: *Genome biology* 8.7 (2007), R145.
- [78] Pavel Tomancak et al. “Systematic determination of patterns of gene expression during Drosophila embryogenesis”. In: *Genome Biol* 3.12 (2002), pp. 0081–0088.
- [79] Oliver Voggenreiter, Stefan Bleuler, Wilhelm Gruissem, et al. “Exact biclustering algorithm for the analysis of large gene expression data sets.” In: *BMC Bioinformatics* 13.S-18 (2012), A10.
- [80] Jordan D Ward et al. “Defects in the C. elegans acyl-CoA synthase, acs-3, and nuclear hormone receptor, nhr-25, cause sensitivity to distinct, but overlapping stresses”. In: *PloS one* 9.3 (2014), e92552.
- [81] Dean F. Wong et al. “In Vivo Imaging of Amyloid Deposition in Alzheimer Disease Using the Radioligand 18F-AV-45 (Flobetapir F 18)”. In: *Journal of Nuclear Medicine* 51.6 (2010), pp. 913–920. DOI: 10 . 2967 / jnumed . 109 . 069088. eprint: <http://jnm.snmjournals.org/content/51/6/913.full.pdf+html>. URL: <http://jnm.snmjournals.org/content/51/6/913.abstract>.
- [82] *WormBase web site, release WS248*. URL: <http://www.wormbase.org> (visited on 06/08/2015).
- [83] Angela R Wu et al. “Quantitative assessment of single-cell RNA-sequencing methods”. In: *Nature methods* 11.1 (2014), pp. 41–46.
- [84] Chen Xu and Zhengchang Su. “Identification of cell types from single-cell transcriptomes using a novel clustering method”. In: *Bioinformatics* 31.12 (2015), pp. 1974–1980.
- [85] Liying Yan et al. “Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells”. In: *Nature structural & molecular biology* 20.9 (2013), pp. 1131–1139.
- [86] Jian-Hua Yang et al. “ChIPBase: a database for decoding the transcriptional regulation of long non-coding RNA and microRNA genes from ChIP-Seq data”. In: *Nucleic acids research* 41.D1 (2012), pp. D177–D187.
- [87] Yong Zhang et al. “Model-based analysis of ChIP-Seq (MACS)”. In: *Genome biology* 9.9 (2008), R137.

# Appendix A

## Supporting material for “Identifying cell subpopulations and their genetic drivers from single cell RNA-Seq data using a biclustering approach”

### A.1 Construction of SNN Network

Let’s demonstrate the edge-weight calculation using a simple example. Let  $X$  be a  $2 \times 8$  matrix, i.e we have 2 genes and 8 cells with fabricated gene expressions, shown in Figure A.1.

$$X = \begin{bmatrix} 0 & 0 & 0.25 & 0.25 & 1.5 & 1.75 & 1.5 & 1.75 \\ 0 & 0.25 & 0.25 & 0 & 1.5 & 1.5 & 1.75 & 1.75 \end{bmatrix}$$

Appendix Figure 1

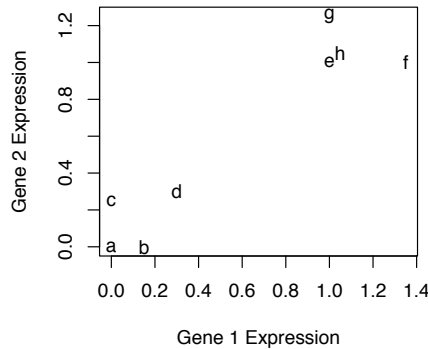


Figure A.1: *Positions of cells in gene expression space. This shows the relative positions of the cells in the gene expression space in our example. It is apparent that our cells should be grouped into two clusters  $\{a, b, c, d\}$  and  $\{e, f, g, h\}$ .*

Let’s calculate the edge weight between  $a$  and  $b$ . First find the list of neighbors ranked in order of proximity (using Euclidean distance), in this case  $a$ ’s neighborhood is  $\{a, b, c, d, e, h, g, f\}$  and that of  $b$  is  $\{b, a, c, d, e, h, g, f\}$ . Then define an integer  $k$  so that we only look at the

top  $k$  neighbors in each list (this is why sometimes shared-nearest-neighbor is also called the  $k$ -nearest-neighbor). Let  $k = 3$ , then the neighbor list we actually use are  $\{a, b, c\}$  for  $a$  and  $\{b, a, c\}$  for  $b$ . Searching through the pair of listings, we find the highest positions of their common neighbors, in this case,  $a$ , who is ranked 0 in  $a$ 's neighborhood and 1 in  $b$ 's (or  $b$ , who is ranked 1 in  $a$ 's neighborhood and 0 in  $b$ 's). Note that even though  $c$  is a common neighbor, it is ranked lower than the other common neighbors (i.e  $a$  and  $b$ ) in the list, so it is not used to calculate proximity. The average rank of the highest common neighbor in this case is  $\frac{0+1}{2} = 0.5$ , and the edge weight is therefore  $k - 0.5 = 3 - 0.5 = 2.5$ . Take another example, suppose we want to create an edge between  $a$  and  $e$ , with  $k = 3$ , then the corresponding neighborhood lists are  $\{a, b, c\}$  and  $\{e, h, g\}$ . Since no common neighbor exists, the an edge will not be drawn between  $a$  and  $e$  in the final graph. Using the procedure described above, Figure A.2 shows the networks create using SNN constructor with  $k = 2$  and  $k = 3$ , respectively.

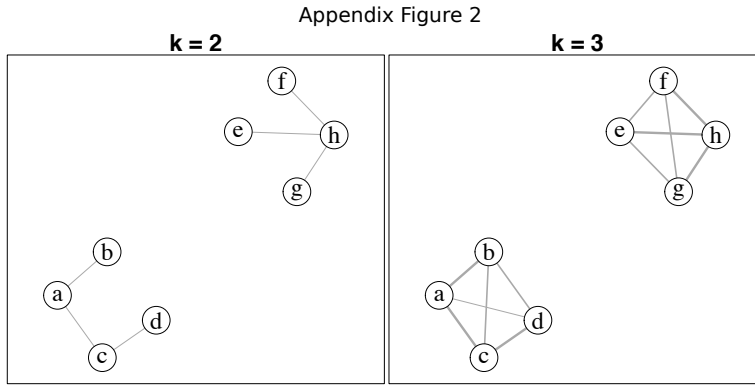


Figure A.2: *Constructed SNN network using example gene expression matrix with  $k = 2$  (left) and  $k = 3$  (right). Notice that the  $k = 3$  network contains more edges, because lower  $k$  will yield a more sparse network by construction.*

In general, let  $n$  be the number of genes and  $m$  be the number of cells, then the  $n \times m$  gene expression represents  $m$  points in  $\mathbb{R}^n$ , the algorithm constructs a network with nodes being cells and edge weights between two cells being pseudo-measure of their proximity in  $\mathbb{R}^n$ . Let  $c$  denote a cell, define a positive integer  $k$  to be the neighborhood size such that  $V_k(c)$  is an ordered list of  $k$  cells who are  $c$ 's closest neighbor measured in Euclidean distance, with the first element of  $V_k(c)$  being the closest to  $c$ . Define  $rank_k(a, c)$  to be the position of cell  $a$  in  $V_k(c)$ . Then the weight of the edge between cells  $a$  and  $c$  is defined as

$$w(a, c) \triangleq \begin{cases} \max \left\{ k - \frac{rank_k(b, a) + rank_k(b, c)}{2} \right\} & \text{if } c \in V(a) \cup V(c) \\ 0 & \text{o.w} \end{cases}$$

In short, the closer two nodes are to their shared neighbor, the more weight will be assigned to the edge that connects them. In BiSNN-Walk  $k$  is hard-coded to be  $\lceil \log(n) \rceil$ . This is because BiSNN-Walk contains a self-correcting scheme, so it does not require a refined selection of  $k$ , and we found that  $\lceil \log(n) \rceil$  is reasonable under most scenarios.

We chose to construct SNN instead of directly using similarity matrices for two reasons. First, in SNN networks, the notion of “distance” between two nodes is established in the context of a local neighborhood instead of quantified by an absolute measure, such as the Euclidean distance. This localization of distance is especially desirable for high dimensional data, where the absolute distance measure like Euclidean distance becomes less and less useful for gauging proximity with higher dimensions [1]. Secondly, the edge weights between two nodes in an SNN network implicitly carries information about the similarity between the two neighborhoods of the two nodes, whereas in a similarity matrix, the similarity score between two cells only carries information about the two nodes themselves. Through its edge construction, SNN creates a filter that condenses informative neighborhood characteristics, which are otherwise lost in similarity matrices.

## A.2 Overview of irreducible discovery rate

One of the most pertinent question in high throughput sequencing is whether the signal we see in the data are real, or true positives. For instance, suppose we were to conduct a Chip-seq experiment to find binding sites (peaks) of a transcription factor, suppose we were to repeat the experiment under identical settings many times, the peaks that show up as significant across experiments would be considered “reproducible”. In practice, however, we usually an experiment is only replicated twice due to budget and time constraints, and irreproducible discovery rate was introduced to quantify the “reproducibility” of the signals in the replicate experiments. Other measures of reproducibility also exist before the introduction of IDR, the most prominent of which include Spearman’s correlation and rank correlation; however, the idea that set IDR apart from its predecessors is that it makes a lot more sense to measure reproducibility using the signals that are actually reproducible; in other words, one should not use the entire experiment to measure reproducibility of replicate experiments. Also, there was a lack of measure that quantify local reproducibility, i.e using the previous Chip-seq example, it is also worthwhile to know the reproducibility of individual peaks.

The main idea of IDR is that it separates the pairs into two groups (remember, we have two replicates of every experiment, thus the data is a  $n \times 2$  matrix, i.e  $n$  pairs), a reproducible group, and a non-reproducible group. Let  $(x_{i,1}, x_{i,2})$  denote the pair of observations, assume  $(x_{i,1}, x_{i,2}) \sim F^1(\cdot, \cdot)$  if the pair belongs to the reproducible group,  $\sim F^0(\cdot, \cdot)$  otherwise. Suppose the proportion of genuine signals is  $\pi_1$  and that of spurious signals is  $\pi_0 = 1 - \pi_1$ , then  $(x_{i,1}, x_{i,2}) \sim F(\cdot, \cdot) = \pi_1 F^1(\cdot, \cdot) + \pi_0 F^0(\cdot, \cdot)$ . Let  $F_1(\cdot) \equiv$  marginal distribution of the first coordinate, and  $F_2(\cdot)$  similarly defined.

Now let's define the dependence structure within each group. Let

$$(z_{i,1}, z_{i,2}) \sim BN \left( \begin{pmatrix} \mu \\ \mu \end{pmatrix}, \begin{pmatrix} \rho\sigma^2 & \sigma^2 \\ \sigma^2 & \rho\sigma^2 \end{pmatrix} \right)$$

$\mu_1 > 0$ ,  $\rho > 0$  if the pair are drawn from the genuine group, otherwise  $(z_{i,1}, z_{i,2}) \sim SBN$ . Here  $\rho$  gauges the overall reproducibility between two experiments, and is the notion of IDR we use in our study. Let  $G$  denote the marginal distributions of  $z_{i,j}$ , then

$$G(\cdot) = \frac{\pi_1}{\sigma} \Phi \left( \frac{\cdot - \mu}{\sigma} \right) + \pi_0 \Phi(\cdot)$$

In our model,  $(z_{i,1}, z_{i,2})$  are the unobserved latent variables that induces  $(x_{i,1}, x_{i,2})$  according to the following relationship

$$x_{i,1} = F_1^{-1}(G(z_{i,1})) \quad (\text{A.1})$$

with  $x_{i,2}$  similarly defined. In other words, the drawing of  $(z_{i,1}, z_{i,2})$  (thus  $G$ ) gives  $(x_{i,1}, x_{i,2})$  their dependence structure, while  $F$  dictates the actual value they will take. This is referred to by the paper as the copula mixture model.

According to Eq. A.1,  $(z_{i,1}, z_{i,2}) = (G^{-1}(F_1(x_{i,1})), G^{-1}(F_2(x_{i,2})))$ . Assume all pairs are independent and identically distributed, i.e they are all induced by their respective i.i.d  $(z_{i,1}, z_{i,2})$ 's, then the semi-parameterized likelihood function is parameterized by  $\theta = (\pi_1, \mu, \rho, \sigma)$  and  $(F_1, F_2)$  and can be written as

$$L(\theta) = \prod_{i=1}^n [\pi_0 h_0(G^{-1}(F_1(x_{i,1})), G^{-1}(F_2(x_{i,2}))) + \quad (\text{A.2})$$

$$\pi_1 h_1(G^{-1}(F_1(x_{i,1})), G^{-1}(F_2(x_{i,2})))] \quad (\text{A.3})$$

Where  $h_1$  is the density of  $BN \left( \begin{pmatrix} \mu_1 \\ \mu_1 \end{pmatrix}, \begin{pmatrix} \rho\sigma^2 & \sigma^2 \\ \sigma^2 & \rho\sigma^2 \end{pmatrix} \right)$  and  $h_0$  is the density of SBN.

EM algorithm is used to fit  $L(\theta)$  using the following steps:

1. First compute the marginal empirical distribution  $\hat{F}_1(x_{i,1}) = \frac{r_{i,1}}{n}$ , where  $r_{i,1}$  = rank of  $x_{i,1}$  in experiment 1.  $\hat{F}_2(x_{i,2})$  similarly defined.
2. Let  $u_{i,1} \equiv \frac{n-1}{n} \hat{F}_1(x_{i,1})$  be the empirical quantile of  $x_{i,1}$ . The factor  $\frac{n-1}{n}$  is applied to avoid unboundedness of  $G^{-1}$  at 1. Obtain  $u_{i,2}$  with similar fashion
3. Initialize  $\theta$ , denote it  $\theta^{(0)} = (\pi_1^{(0)}, \rho^{(0)}, \mu^{(0)}, \sigma^{(0)})$
4. Compute pseudo data  $z_{i,1} = G^{-1}(u_{i,1})$ , and  $z_{i,2}$ .

5. Apply EM algorithm on the likelihood function of the augmented dataset  $Y_i = (\mathbf{z}_i, K_i)$ , where  $\mathbf{z}_i = (z_{i,1}, z_{i,2})$  and the latent variable

$$K_i = \begin{cases} 1 & \mathbf{z}_i \in \text{reproducible group} \\ 0 & \text{o.w} \end{cases}$$

The corresponding likelihood is

$$l(\theta) \equiv \sum_{i=1}^n \{K_i (\log \pi_1 + \log h_1(\mathbf{z}_i)) + (1 - K_i) (\log \pi_0 + \log h_0(\mathbf{z}_i))\}$$

For the Expectation step, we need to find the expectation of  $l(\theta)$ :

$$\begin{aligned} Q(\theta|\theta^{(0)}) &= E_{K|Z, \theta^{(0)}} l(\theta) \\ &= \sum_{i=1}^n \{E_{K|Z, \theta^{(0)}} [K_i] (\log \pi_1 + \log h_1(\mathbf{z}_i)) + [1 - E_{K|Z, \theta^{(0)}} (K_i)] (\log \pi_0 + \log h_0(\mathbf{z}_i))\} \end{aligned}$$

where

$$\begin{aligned} E_{K|Z, \theta^{(0)}} [K_i] &= P(K_i = 1 | \mathbf{z}_i, \theta^{(0)}) \\ &= \frac{P(K_i = 1, \mathbf{z}_i | \theta^{(0)})}{P(\mathbf{z}_i | \theta^{(0)})} \\ &= \frac{\pi_1^{(0)} h_1(\mathbf{z}_i)}{\pi_1^{(0)} h_1(\mathbf{z}_i) + (1 - \pi_1^{(0)}) h_0(\mathbf{z}_i)} \end{aligned}$$

For the Maximization step, we need to maximize  $Q(\theta|\theta^{(0)})$  which involves fairly straightforward calculus steps. After convergence, set the resulting  $\theta$  as  $\theta^{(1)}$ .

6. If convergence criterion is not met, e.g.  $\|\theta^{(0)} - \theta^{(1)}\| < \epsilon$  for some predefined  $\epsilon$ , set  $\theta^{(1)} \mapsto \theta^{(0)}$  and return to step 4.

### A.3 Overview of Walktrap Clustering

Walktrap Clustering was proposed by Pascal Pons and Matthieu Latapy in [57]. The method uses an agglomerative hierarchical clustering to cluster the the nodes, and the paper also suggests a method of cutting the resulting tree. Notation here will follow the paper as closely as possible.

Let  $G(V, E)$  be an undirected graph with vertices  $V$  and edges  $E$ , where  $|V| = n$ , and  $|E| = m$ . Let  $A$  be the adjacency or weight matrix, and  $D = \text{diag}(d_1, \dots, d_n)$  is the  $n \times n$

diagonal matrix containing the corresponding degree (sum of weights if graph is weighted) of each node.  $P = [P_{ij}] = \left[ \frac{A_{ij}}{d_i} \right]$  be the corresponding  $n \times n$  transition matrix.

The most important piece of any hierarchical clustering algorithm is the definition of distance between nodes. Here the distance between nodes is defined as

**Definition.** The distance between node  $i$  and  $j$  is

$$r_{ij}(t) = \sqrt{\sum_{k=1}^n \frac{([P^t]_{ik} - [P^t]_{jk})^2}{d_k}} \quad (\text{A.4})$$

$$= \|D^{-\frac{1}{2}} [P^t]_{i\cdot} - D^{-\frac{1}{2}} [P^t]_{j\cdot}\| \quad (\text{A.5})$$

where  $t$  is some predefined time.

Since

$$[P^t]_{ij} = \mathbb{P}(\text{a walk starting at node } i \text{ will end up at } j \text{ at time } t)$$

The vector  $[P^t]_{i\cdot}$  can be thought of a visiting “profile” of a walk starting at  $i$ , at time  $t$ , then if  $i$  and  $j$  share many neighbors, then their visiting profile should be similar, thus the corresponding distance  $r_{ij}(t)$  should be small. The  $D^{-\frac{1}{2}}$  factor is just a normalizing factor that down weights the effect of nodes with large degrees, whom, by the nature of the transition matrix, will be visited more no matter the starting position.

$r_{ij}(t)$  is closely associated with the spectral properties of the transition matrix  $P$ . Let  $\{\lambda_\alpha : 1 \leq \alpha \leq n\}$  and  $\{v_\alpha : 1 \leq \alpha \leq n\}$  be the eigenvalues and eigenvectors of  $P$ , then

$$r_{ij}^2(t) = \sum_{\alpha=1}^n \lambda_\alpha^{2t} (v_\alpha(i) - v_\alpha(j))^2$$

where  $v_\alpha(i)$  is the  $i^{\text{th}}$  element of the vector  $v_\alpha$ .

Since  $r_{ij}(t)$  is the “distance” between node  $i$  and  $j$  only at time  $t$ , it’s a better idea to examine at the entire history of the walk over all  $t$ , which leads to the generalized distance  $\hat{r}_{ij}$ , which defined as

**Definition.** Let  $\{c_k : k = 1, \dots, \infty, c_k \geq 0 \forall k, \sum c_k = 1\}$  be a set of predefined weights. Let  $\hat{P}_{i\cdot} = \sum_{k=1}^{\infty} c_k P_{i\cdot}^k$ , The generalized distance

$$\hat{r}_{ij}^2 = \sum_{\alpha=1}^n f^2(\lambda_\alpha) (v_\alpha(i) - v_\alpha(j))^2 \quad (\text{A.6})$$

$$= \|D^{-\frac{1}{2}} \hat{P}_{i\cdot} - D^{-\frac{1}{2}} \hat{P}_{j\cdot}\| \quad (\text{A.7})$$

where  $f(x) = \sum_{k=1}^{\infty} c_k x^k$  is a power series function dictated by  $\{c_k\}$ .

**Example.** If we consider the continuous parallel of the random walk defined by  $P$ , i.e in the continuous random walk, the probability of a walk starting in node  $i$  and ending up in node  $j$  after time  $t$  is

$$\left[ e^{(P-I)t} \right]_{ij}$$

Then the associated generalized distance with this transition matrix is

$$\hat{r}_{ij}^2 = \sum_{\alpha=1}^n e^{2t(\lambda_{\alpha}-1)} (v_{\alpha}(i) - v_{\alpha}(j))^2 \quad (\text{A.8})$$

with  $c_k = \frac{t^k}{k!} e^{-t}$ .

Since computing  $\hat{r}_{ij}^2$  exactly require us to know all the eigenvectors, it is entirely possible when  $P$  is small, but becomes quite expensive when  $P$  is large ( $O(n^3)$ ), so in most cases we will use the form A.8 and approximate  $\hat{P}_i$ . To approximate  $\sum_{k=1}^{\infty} c_k P_i^k$  notice that since  $\sum_j [P^k]_{ij} = 1$  and  $[P^k]_{ij} \geq 0 \forall i, j$ , and  $\sum_k c_k = \sum_k \frac{t^k}{k!} e^{-t} = 1$ , then for any  $\epsilon > 0$ , there exists an integer  $r$  such that  $\|\sum_{k=r+1}^{\infty} c_k P_i^k\| < \epsilon$  by Cauchy-Schwartz. We can approximate  $\hat{P}_i$  with  $\sum_{k=1}^r c_k P_i^k$  with some predefined  $r$ .

So far we only talked about node-to-node distance, in order for the distance to be used in a heirarchical setting, we'll need to extend this notion to cluster-cluster and cluster-node setting. Let  $C$  be a cluster, the average probability of a walk starting at any of the members in  $C$  to reach node  $j$  is

$$\hat{P}_{Cj} = \frac{1}{|C|} \sum_{i \in C} \hat{P}_{ij}$$

then the corresponding generalized distance between two clusters is

$$\hat{r}_{C_1 C_2} = \| D^{-\frac{1}{2}} \hat{P}_{C_1} - D^{-\frac{1}{2}} \hat{P}_{C_2} \|$$

Therefore, to build the tree, we will start with every node being its own cluster, call this clustering  $\mathcal{P}_1$ . And in the next step, like the regular hierarchical clustering, we will merge two of the clusters (nodes) in  $\mathcal{P}_1$  to obtain the next clustering  $\mathcal{P}_2$ . For each step  $k$  clusters from the clustering  $\mathcal{P}_{k-1}$ , and all nodes will be merged into a single cluster by step  $n - 1$ , which will be the root of the tree.

At each step we will merge clusters by minimizing the following quantity

$$\sigma_k = \frac{1}{n} \sum_{C \in \mathcal{P}_k} \sum_{i \in C} \hat{r}_{iC}^2$$

Which is the average squared distance of a node to the cluster it belongs to. Minimizing this quantity directly at each step is computationally intensive, and requiring  $O(|\mathcal{P}_k|^2)$



computation time for each  $k$ , instead we try to find, let  $C_1, C_2 \in \mathcal{P}_k$ , and  $C_3 = C_1 \cup C_2$ , then

$$\Delta\sigma(C_1, C_2) = \frac{1}{n} \left( \sum_{i \in C_3} \hat{r}_{iC_3} - \sum_{i \in C_1} \hat{r}_{iC_1} - \sum_{i \in C_2} \hat{r}_{iC_2} \right)$$

Which relates to  $\hat{r}_{C_1 C_2}^2$  like

$$\Delta\sigma(C_1, C_2) = \frac{1}{n} \frac{|C_1| |C_2|}{|C_1| + |C_2|} \hat{r}_{C_1 C_2}$$

So as long as we know  $\hat{r}_{C_1 C_2}$ ,  $\Delta\sigma(C_1, C_2)$  can be calculated in linear time.

To cut the tree, we use the quantity

$$\eta_k = \frac{\Delta\sigma_k}{\Delta\sigma_{k-1}} = \frac{\sigma_{k+1} - \sigma_k}{\sigma_k - \sigma_{k-1}}$$

Intuitively, the idea is that when two very distant communities are merged, we would see a large  $\Delta\sigma$ , so the preferable clusering  $\mathcal{P}_k$  should contain distant clusters so that further merging of  $\mathcal{P}_{k+1}$  would greatly increase  $\sigma_k$ , but previous clustering  $\mathcal{P}_{k-1}$  still contain similar clusters such that  $\mathcal{P}_{k-1}$  to  $\mathcal{P}_k$  does not increase  $\sigma_k$  significantly, that is, we cut the tree at  $k = \text{argmax}_k \eta_k$ .

## A.4 Overview of Adjusted Rand Index

The Rand index is developed by William M. Rand for the purpose of quantifying the agreement between two clustering results in his seminal paper “Objective criteria for the evaluation of clustering methods” [61]. The method assumes that the clusters do not overlap, i.e each item belongs to only one cluster. In our case, let  $U = \{U_1, \dots, U_M\}$  and  $V = \{V_1, \dots, V_N\}$  be two sets of clusters on cells  $1, \dots, n$ . Define the following quantities, as mentioned in the main text,

$$a = \text{pairs belong to the same cluster in } U \text{ as well as } V \quad (\text{A.9})$$

$$b = \text{pairs belong to the same cluster in } U \text{ but different clusters in } V \quad (\text{A.10})$$

$$c = \text{pairs belong to different clusters in } U \text{ but the same cluster in } V \quad (\text{A.11})$$

$$d = \text{pairs belong to different clusters in } U \text{ as well as } V \quad (\text{A.12})$$

$a$  is a set of nodes, but if there is no confusion we will also use  $a$  to denote its cardinality. The Rand index of the clustering  $U, V$  is

$$RI(U, V) = \frac{a + d}{a + b + c + d} = \frac{a + d}{\binom{n}{2}}$$

The Rand Index is bounded between  $[0, 1]$ . Suppose we break down the cluster memberships in a different way as shown in Table A.1.

	$V_1$	$V_2$	$\cdots$	$V_N$	total
$U_1$	$n_{11}$	$n_{12}$	$\cdots$	$n_{1N}$	$n_{1\cdot}$
$U_2$	$n_{21}$	$n_{22}$	$\cdots$	$n_{2N}$	$n_{2\cdot}$
$\vdots$	$\vdots$		$\ddots$	$\vdots$	$\vdots$
$U_M$	$n_{M1}$	$n_{M2}$	$\cdots$	$n_{MN}$	$n_{M\cdot}$
total	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot N}$	$n$

Table A.1: Contingency table showing the break-down of membership assignment of node-pairs. Here  $n_{ij}$  = number of nodes that are simultaneously assigned to clusters  $U_i$  and  $V_j$ .

Table A.1 allows us to easily calculate a few important quantities, e.g

$$\begin{aligned} \binom{n_{ij}}{2} &= \text{total number of possible node pairs that are assigned to } U_i \text{ and } V_j \\ \binom{n_{i\cdot}}{2} &= \text{total number of possible node pairs that are assigned to } U_i \\ \binom{n}{2} &= \text{total number of possible node pairs} \end{aligned}$$

Using these quantities we can calculate the probability of a node pair belonging to  $a + d$ :

$$\begin{aligned} a + d &= \underbrace{\left\{ \underbrace{\binom{n}{2}}_{\text{total \# pairs}} - \underbrace{\left[ \sum_{i=1}^M \binom{n_{i\cdot}}{2} + \sum_{j=1}^N \binom{n_{\cdot j}}{2} - \sum_{i=1}^M \sum_{j=1}^N \binom{n_{ij}}{2} \right]}_{\text{total \# pairs clustered together in at least one of the partitions}} \right\}}_{\text{total \# pairs that are clustered into different clusters in both partitions}} + \\ &\quad \underbrace{\sum_{i=1}^M \sum_{j=1}^N \binom{n_{ij}}{2}}_{\text{total \# pairs that were clustered to the same cluster in both partitions}} \\ &= \binom{n}{2} + 2 \sum_{i=1}^M \sum_{j=1}^N \binom{n_{ij}}{2} - \left[ \sum_{i=1}^M \binom{n_{i\cdot}}{2} + \sum_{j=1}^N \binom{n_{\cdot j}}{2} \right] \end{aligned}$$

One issue with the Rand Index is that, suppose  $T = \{T_l, l = 1 \dots L\}$  is the ground truth partition, then  $RI(U, T)$  and  $RI(V, T)$  are not comparable, that is, even if  $RI(U, T) >$

$RI(V, T)$  it is not necessarily the case that  $U$  is a better partition than  $V$  with respect to  $T$  because there is no consistent baseline measure. In other words, comparing  $RI(U, T)$  and  $RI(V, T)$  is akin to comparing realizations of  $X \sim N(\mu_x, \sigma^2)$  and  $Y \sim N(\mu_y, \sigma^2)$  without actually knowing what  $\mu_x$  and  $\mu_y$  are. This fact severely limits the usefulness of the Rand Index; therefore, Hubert and Arabie proposed to frame the problem in terms of a hypergeometric model [33], in which we assume that the number of elements in  $n_i$ ’s and  $n_j$ ’s are fixed, and  $N_{ij}$ ’s are random variables. Then, the probability of a node pair to belong to  $U_i$  and  $V_j$  is

$$E \left[ \frac{\binom{N_{ij}}{2}}{\binom{n}{2}} \right] = \frac{\binom{n_{i.}}{2} \binom{n_{.j}}{2}}{\binom{n}{2} \binom{n}{2}} \quad (\text{A.13})$$

Thus

$$E \left[ \binom{N_{ij}}{2} \right] = \frac{\binom{n_{i.}}{2} \binom{n_{.j}}{2}}{\binom{n}{2}}$$

Then, with some simple algebra

$$E[RI(U, V)] = \frac{E \left[ \binom{n}{2} + 2 \sum_{i=1}^M \sum_{j=1}^N \binom{N_{ij}}{2} - \left[ \sum_{i=1}^M \binom{n_{i.}}{2} + \sum_{j=1}^N \binom{n_{.j}}{2} \right] \right]}{\binom{n}{2}} \quad (\text{A.14})$$

$$= 1 + 2 \sum_{i=1}^M \sum_{j=1}^N \frac{\binom{n_{i.}}{2} \binom{n_{.j}}{2}}{\binom{n}{2}^2} - \left[ \sum_{i=1}^M \frac{\binom{n_{i.}}{2}}{\binom{n}{2}} + \sum_{j=1}^N \frac{\binom{n_{.j}}{2}}{\binom{n}{2}} \right] \quad (\text{A.15})$$

Using the chance-corrected form of an index:  $\frac{index - E[index]}{max[index] - E[index]}$  and noting Rand Index is

boudned above by 1, then

$$ARI(U, V) = \frac{RI(U, V) - E[RI(U, V)]}{1 - E[RI(U, V)]} \quad (\text{A.16})$$

$$= \frac{\sum_{i=1}^M \sum_{j=1}^N \binom{n_{ij}}{2} - \frac{\sum_{i=1}^M \binom{n_{i\cdot}}{2} \sum_{j=1}^N \binom{n_{\cdot j}}{2}}{\binom{n}{2}}}{\frac{1}{2} \left[ \sum_{i=1}^M \binom{n_{i\cdot}}{2} + \sum_{j=1}^N \binom{n_{\cdot j}}{2} \right] - \frac{\sum_{i=1}^M \binom{n_{i\cdot}}{2} \sum_{j=1}^N \binom{n_{\cdot j}}{2}}{\binom{n}{2}}} \quad (\text{A.17})$$

## Appendix B

# Supporting material for “A more powerful and flexible method of measuring Amyloid- $\beta$ accumulation using linear model”

### B.1 Linear relationship between target vs reference SUV persists across time

This section contains scatter plots of  $T_1$  vs  $R_1$ ,  $T_2$  vs  $R_2$ , and  $T_2 - T_1$  vs  $R_2 - R_1$  across all reported target vs reference ROIs for both ADNI and BLAZE data sets. From these plots one can see that the linear relationship between target and reference SUVs persists through time, target/reference combinations, as well as processing pipelines, indicating that the linear relationship is likely biologically meaningful.

Due to their large dimensions, plots may not display properly within the document, so the corresponding URLs are provided in their respective captions for reference.

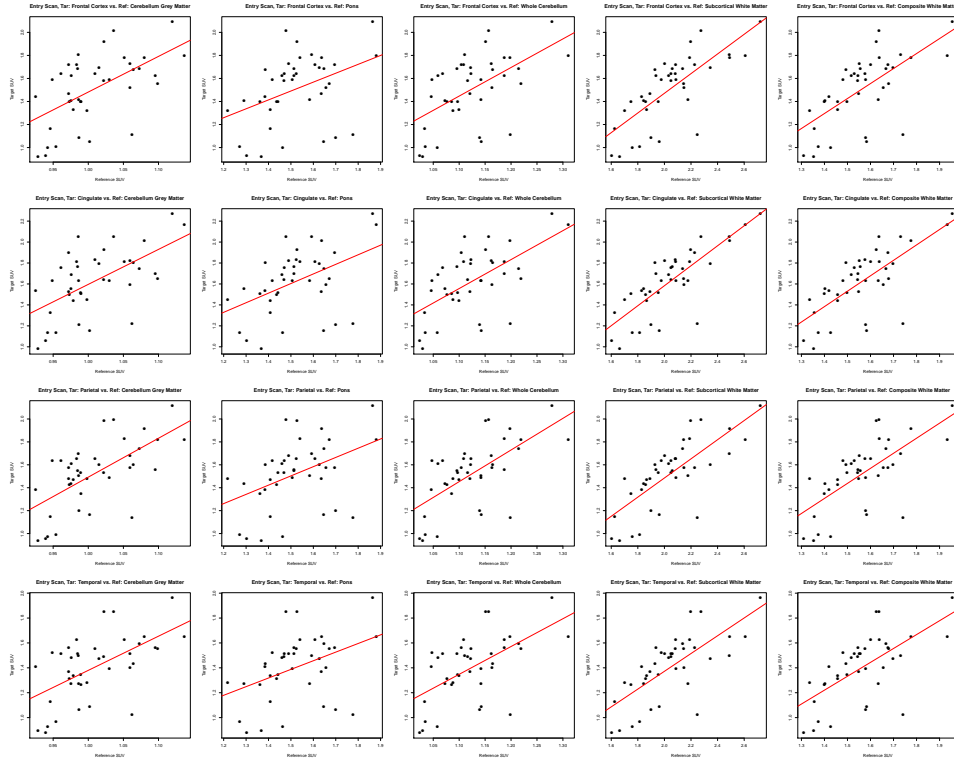


Figure B.1:  $T_1$  vs  $R_1$  across all target and reference ROIs for ADNI. The linear relationship is particularly strong with Subcortical White Matter and Composite White Matter as reference ROIs. For full size image see: [https://www.dropbox.com/s/qgu2ofwa8y9t769/adni\\_slopeVSinterceptALLpoints-t1.pdf?dl=0](https://www.dropbox.com/s/qgu2ofwa8y9t769/adni_slopeVSinterceptALLpoints-t1.pdf?dl=0)

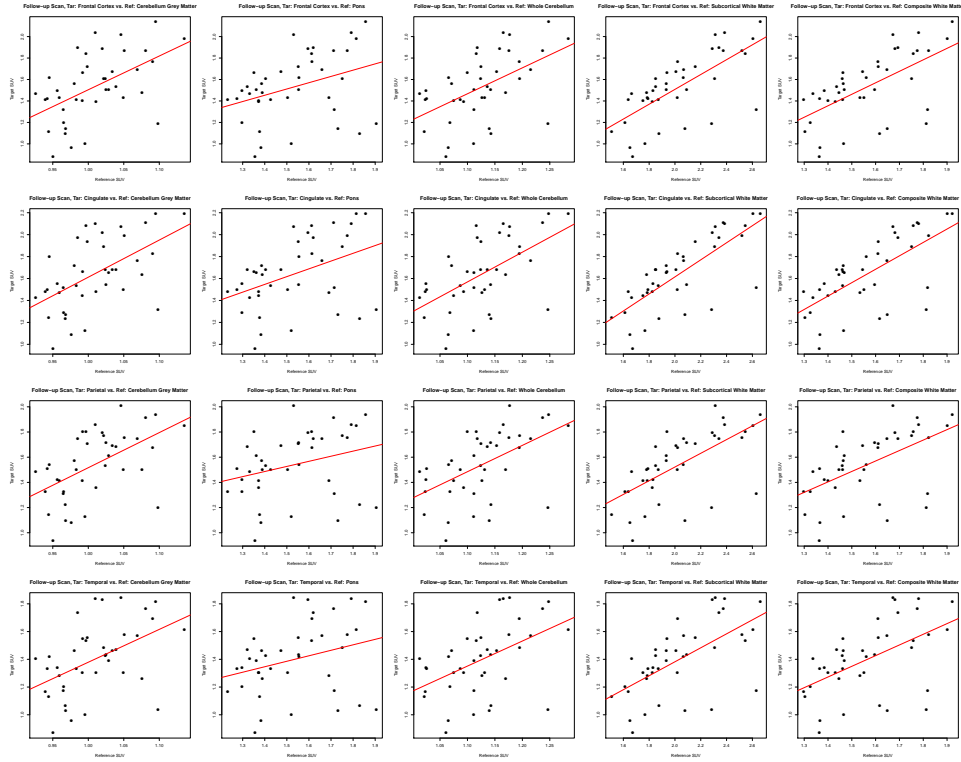


Figure B.2:  $T_2$  vs  $R_2$  across all target and reference ROIs for ADNI. As with Figure B.1, the linear relationship is strongest with Subcortical White Matter and Composite White Matter as reference ROIs. For full size image see: [https://www.dropbox.com/s/aig3uefm0ud9kcx/adni\\_slopeVSinterceptALLpoints-t2.pdf?dl=0](https://www.dropbox.com/s/aig3uefm0ud9kcx/adni_slopeVSinterceptALLpoints-t2.pdf?dl=0)

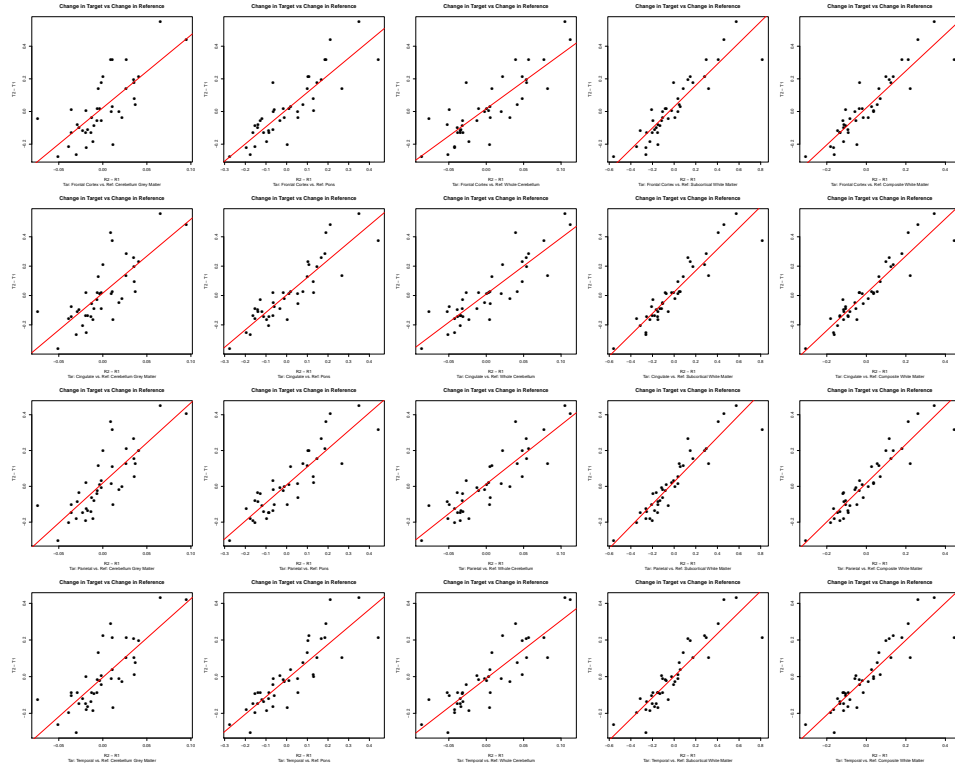


Figure B.3:  $T_2 - T_1$  vs  $R_2 - R_1$  across all target and reference ROIs for ADNI. The linear relationship is very strong across combinations of target and reference ROIs. For full size image see: [https://www.dropbox.com/s/evsrar7coa2bajt/adni\\_slopeVSinterceptALLpoints-t2-t1.pdf?dl=0](https://www.dropbox.com/s/evsrar7coa2bajt/adni_slopeVSinterceptALLpoints-t2-t1.pdf?dl=0)



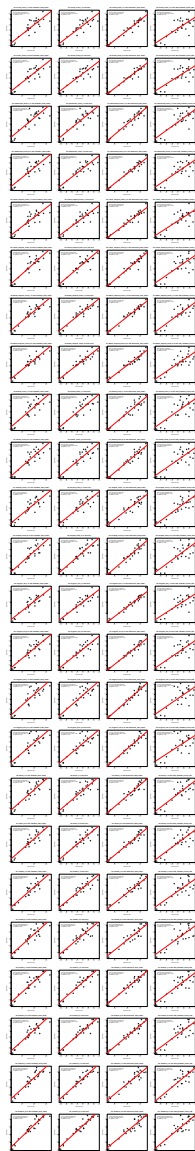


Figure B.4:  $T_1$  vs  $R_1$  across all target and reference ROIs for BLAZE. Linear relationship carries across all combinations of target and reference ROIs. For full size image see: [https://www.dropbox.com/s/vff0m791m07smkk/blaze\\_slopeVSinterceptALLpoints-t1.pdf?dl=0](https://www.dropbox.com/s/vff0m791m07smkk/blaze_slopeVSinterceptALLpoints-t1.pdf?dl=0)

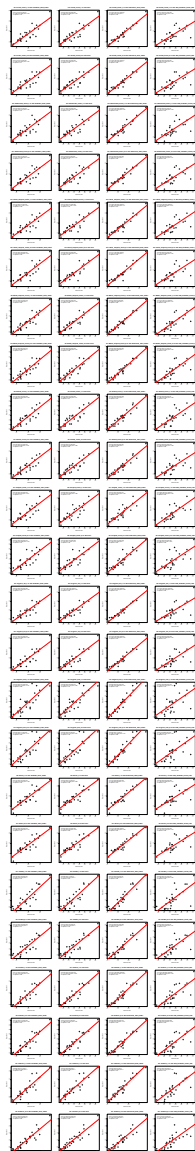


Figure B.5:  $T_2$  vs  $R_2$  across all target and reference ROIs for BLAZE. Linear relationship carries across all combinations of target and reference ROIs. For full size image see: [https://www.dropbox.com/s/2q86bgqwcqtufuo/blaze\\_slopeVSinterceptALLpoints-t2.pdf?dl=0](https://www.dropbox.com/s/2q86bgqwcqtufuo/blaze_slopeVSinterceptALLpoints-t2.pdf?dl=0)

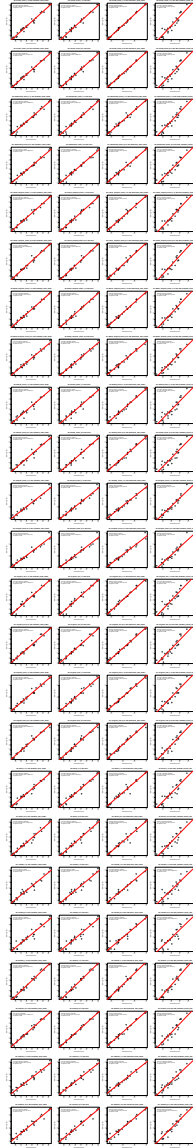


Figure B.6:  $T_2 - T_1$  vs  $R_2 - T_1$  across all target and reference ROIs for BLAZE. Linear relationship carries across all combinations of target and reference ROIs. For full size image see: [https://www.dropbox.com/s/3fq981zwn9tbr68/blaze\\_slopeVSinterceptALLpoints-t2-t1.pdf?dl=0](https://www.dropbox.com/s/3fq981zwn9tbr68/blaze_slopeVSinterceptALLpoints-t2-t1.pdf?dl=0)

## B.2 ADNI linear regression diagnostic plots

ADNI: Diagnostic plots for  $T_l \sim R_l$  Regression

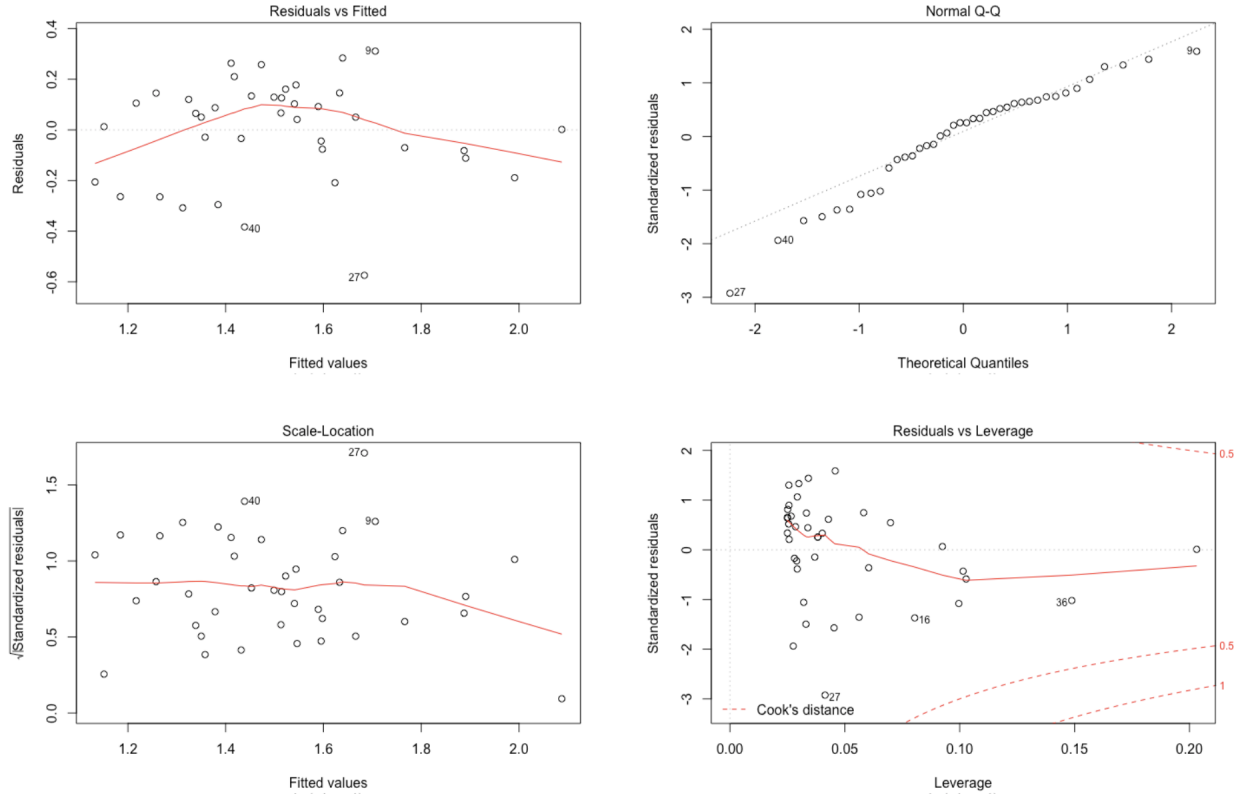


Figure B.7: Diagnostic plots for linear models regressing  $T_1$  on  $R_1$  for ADNI data set. Scale-location and residual-fitted plots indicated no heteroskedasticity. QQ plot showed that normality assumption holds reasonably well. Leverage plot identified no influential points. See Figure 3.6 for corresponding diagnostic plot for BLAZE data set. Target: Frontal Cortex, Reference: Subcortical White Matter.

## B.3 Using feasible generalized least squares to verify that $\beta$ is constant across time

This is an aside from the main text showing that the slope parameter  $\beta$  can be assumed to be constant across time. One can safely skip this section without losing context.

We use Feasible Generalized Least Squares (FGLS) [20]. The regression framework takes

the following form

$$\begin{bmatrix} T_1^1 \\ T_1^2 \\ \vdots \\ T_1^{30} \\ T_2^1 \\ T_2^2 \\ \vdots \\ T_2^{30} \end{bmatrix} = \begin{bmatrix} 1 & 0 & R_1^1 & 0 \\ 1 & 0 & R_1^2 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & R_1^{30} & 0 \\ 0 & 1 & 0 & R_2^1 \\ 0 & 1 & 0 & R_2^2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & R_2^{30} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1^1 \\ \epsilon_1^2 \\ \vdots \\ \epsilon_1^{30} \\ \epsilon_2^1 \\ \epsilon_2^2 \\ \vdots \\ \epsilon_2^{30} \end{bmatrix}$$

Let  $A_{30 \times 30} = \begin{bmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{bmatrix}$ ,  $B_{30 \times 30} = \begin{bmatrix} \rho & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \rho \end{bmatrix}$ , then the covariance of the error term  $[\epsilon_1^1, \dots, \epsilon_1^{30}, \epsilon_2^1, \dots, \epsilon_2^{30}]$  is  $G_{60 \times 60} = \begin{bmatrix} A & B \\ B & A \end{bmatrix}$ . The pseudo-code for the FGLS procedure is as follows:

1. Initialize  $\sigma^2 = 1$  and  $\rho = 0$ , i.e  $G = I_{60 \times 60}$ . Set **converged** = **False**
2. **While** (not converged)
  - a) Let  $Q$  be a matrix such that  $G = QQ^T$
  - b) Let  $\tilde{X} = Q^{-1}X$ ,  $\tilde{Y} = Q^{-1}Y$ , fit the regression model  $\tilde{Y} \sim \tilde{X}$  and obtain the coefficient estimates. Obtain the residual vector  $[e_1^1, \dots, e_1^{30}, e_2^1, \dots, e_2^{30}]$ .
  - c) Set  $\sigma_{new}^2 = \frac{1}{60}[(e_1^1)^2 + \dots + (e_1^{30})^2 + (e_2^1)^2 + \dots + (e_2^{30})^2]$ ,  $\rho_{new} = \frac{1}{30} \sum_{i=1}^{30} e_1^i e_2^i$ ,  
 and set  $A_{new} = \begin{bmatrix} \sigma_{new}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{new}^2 \end{bmatrix}$ ,  $B_{new} = \begin{bmatrix} \rho_{new} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \rho_{new} \end{bmatrix}$ ,  $G_{new} = \begin{bmatrix} A_{new} & B_{new} \\ B_{new} & A_{new} \end{bmatrix}$
  - d) If  $\|G - G_{new}\|_F < \text{some predefined threshold}$ , where  $\|\cdot\|_F$  is the Frobenius norm, set **converged** = **True**. Otherwise let  $G = G_{new}$  and repeat the **while** loop.

3. Return the regression estimates  $\hat{\beta} = \begin{bmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix}$  and covariance matrix  $[X^T G^{-1} X]^{-1}$ .

We apply FGLS procedure to BLAZE data with  $t_1 = \text{entry scan}$  and  $t_2 = \text{week 47}$ . To test whether  $\hat{\beta}_1 = \hat{\beta}_2$ , i.e a contrast function of  $\Lambda = [0, 0, 1, -1]^T$ , we use the common F-test for contrasts [13], which states that  $\frac{(\Lambda^T \hat{\beta})^2}{\Lambda^T G \Lambda} \sim F(1, 56)$ , which returns a effect size of

0.064 and the corresponding p-value of 0.80, thus the conclusion of accepting the null that  $\hat{\beta}_1 - \hat{\beta}_2 = 0$ . When the same procedure is carried out on ADNI data with 40 patients, we obtain a F-statistic of 0.012 on 76 degrees of freedom, which translates to a p-value of 0.91, thus for ADNI data, we also accept the null that  $\beta_1 - \beta_0 = 0$ .

## B.4 Deviation details of Proposition 3.5.1

This section contains derivation details of We will use Formulae B.1 and B.2 repeatedly: if  $E(X) = \mu_X, E(Y) = \mu_Y, Var(X) = \sigma_X^2, Var(Y) = \sigma_Y^2$ , and  $cor(X, Y) = \rho$ , then

$$E\left(\frac{X}{Y}\right) \approx \frac{\mu_X}{\mu_Y} + \frac{1}{\mu_Y^2}[\sigma_Y^2\left(\frac{\mu_X}{\mu_Y}\right) - 2\rho\sigma_Y\sigma_X] \quad (B.1)$$

$$E(XY) \approx \mu_X\mu_Y + 2\rho\sigma_X\sigma_Y \quad (B.2)$$

Notice also that  $(\frac{R-\mu}{\sigma_R})^2 \sim \chi_1^2$ , another fact that we will exploit repeatedly.

From here on out, patient ID superscript will be omitted, i.e  $T_t^i = T_t, R_t^i = R_t$

### B.4.1 SUVR

For some single person, his/her SUVR,  $SUVR_t$ , according to the the generation model, Eq 3.3 are:

$$\begin{aligned} SUVR_2 &= \frac{\alpha_2}{R_2} + \beta + \frac{Z}{R_2} + \frac{\epsilon_2}{R_2} \\ SUVR_1 &= \frac{\alpha_1}{R_1} + \beta + \frac{Z}{R_1} + \frac{\epsilon_1}{R_1} \end{aligned}$$

Thus the expected value of  $\Delta SUVR$  is calculated as

$$\begin{aligned} \Delta SUVR &= SUVR(t_2) - SUVR(t_1) \\ &= \left(\frac{\alpha_2}{R_2} - \frac{\alpha_1}{R_1}\right) + Z\left(\frac{1}{R_2} - \frac{1}{R_1}\right) + \left(\frac{\epsilon_2}{R_2} - \frac{\epsilon_1}{R_1}\right) \\ E[\Delta SUVR] &= E\left[\left(\frac{\alpha_2}{R_2} - \frac{\alpha_1}{R_1}\right) + Z\left(\frac{1}{R_2} - \frac{1}{R_1}\right) + \left(\frac{\epsilon_2}{R_2} - \frac{\epsilon_1}{R_1}\right)\right] \\ &\approx \frac{\alpha_2 - \alpha_1}{\mu_R} \left(1 + \left(\frac{\sigma_R}{\mu_R}\right)^2\right) = \frac{\alpha_2 - \alpha_1}{\mu_R} (1 + CV_R^2), \end{aligned}$$

Because

$$E(Z) = 0 \text{ and } \frac{\epsilon_2}{R_2} =_D \frac{\epsilon_1}{R_1}$$

$CV_R = \frac{\sigma_R}{\mu_R}$  is the coefficient of variation of  $R$

The variance calculation is a bit more complicated

$$\begin{aligned}
 Var(\Delta SUVr) &= Var\left[\frac{\alpha_2 + Z + \epsilon_2}{R_2} - \frac{\alpha_1 + Z + \epsilon_1}{R_1}\right] \\
 &= \underbrace{Var\left(\frac{\alpha_2 + Z + \epsilon_2}{R_2}\right)}_{(*)} + \underbrace{Var\left(\frac{\alpha_1 + Z + \epsilon_1}{R_1}\right)}_{(**)} \\
 &\quad - \underbrace{2 Cov\left(\frac{\alpha_2 + Z + \epsilon_2}{R_2}, \frac{\alpha_1 + Z + \epsilon_1}{R_1}\right)}_{(\#)}
 \end{aligned}$$

We will use the following side calculations for  $(*)$  and  $(\#)$  respectively. Calculation for  $(**)$  will be omitted since the calculation of the two are very similar.

Side calculation for  $(*)$ . A note on notation,  $E(\mathbf{expr})^2 \equiv E(\mathbf{expr}^2)$  and  $E^2(\mathbf{expr}) \equiv [E(\mathbf{expr})]^2$ .

$$\begin{aligned}
 (*) &= E\left(\frac{1}{R_2^2}\right)E(\alpha_2 + Z + \epsilon_2)^2 - E^2\left[\frac{1}{R_2}\right]\underbrace{E^2(\alpha_2 + Z + \epsilon_2)}_{=\alpha_2^2} \\
 &\approx \left(\frac{1}{\mu^2 + \sigma_R^2} + \frac{2\sigma_R^2(\sigma_R^2 + 2\mu^2)}{(\mu^2 + \sigma_R^2)^3}\right)(\sigma_Z^2 + \sigma_\epsilon^2 + \alpha_2^2) - \frac{\alpha_2^2}{\mu^2}\left[1 + \left(\frac{\sigma_R}{\mu}\right)^2\right]^2 \\
 &\text{owing to the fact that } \left(\frac{R_2}{\sigma_R}\right)^2 \sim \chi_1^2\left(\left(\frac{\mu}{\sigma_R}\right)^2\right) \\
 &\text{so } Var(R_2^2) = \sigma_R^4 Var\left(\left(\frac{R_2}{\sigma_R}\right)^2\right) = \sigma_R^4 2\left(1 + 2\left(\frac{\mu}{\sigma_R}\right)^2\right) = 2\sigma_R^2(\sigma_R^2 + 2\mu^2)
 \end{aligned}$$

Side calculation for (#)

$$\begin{aligned} (\#) &= E\left[\frac{(\alpha_2 + Z + \epsilon_2)(\alpha_1 + Z + \epsilon_1)}{R_1 R_2}\right] \\ &\quad - E\left[\frac{\alpha_2 + Z + \epsilon_2}{R_2}\right] E\left[\frac{\alpha_1 + Z + \epsilon_1}{R_1}\right] \end{aligned}$$

where

$$\begin{aligned} E\left[\frac{(\alpha_2 + Z + \epsilon_2)(\alpha_1 + Z + \epsilon_1)}{R_1 R_2}\right] &= E\left[\frac{\alpha_1 \alpha_2 + Z^2 + (\alpha_1 + \alpha_2)Z + (\epsilon_1 + \epsilon_2)Z + \epsilon_1 \epsilon_2 + \alpha_1 \epsilon_2 + \alpha_2 \epsilon_1}{R_1 R_2}\right] \\ &\quad Var(R_1 R_2) \\ &\approx \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{\mu^2 + 2\rho\sigma_R^2} + \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{[\mu^2 + 2\rho\sigma_R^2]^3} [2\mu^2 \sigma_R^2 (1 - \rho) + 4\rho\sigma_R^2 (2\mu^2 - \rho\sigma_R^2)] \end{aligned}$$

and

$$E\left[\frac{\alpha_2 + Z + \epsilon_2}{R_2}\right] \approx \frac{\alpha_2}{\mu} + \frac{1}{\mu^2} [\sigma_R^2 \frac{\alpha_2}{\mu}] = \frac{\alpha_2}{\mu} \left[1 + \left(\frac{\sigma_R}{\mu}\right)^2\right]$$

so

$$\begin{aligned} (\#) &= \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{\mu^2 + 2\rho\sigma_R^2} \left[1 + \frac{2\mu^2 \sigma_R^2 (1 - \rho) + 4\rho\sigma_R^2 (2\mu^2 - \rho\sigma_R^2)}{(\mu^2 + 2\rho\sigma_R^2)^2}\right] \\ &\quad - \frac{\alpha_1 \alpha_2}{\mu^2} \left[1 + \left(\frac{\sigma_R}{\mu}\right)^2\right]^2 \end{aligned}$$



Thus

$$\begin{aligned}
 Var(\Delta SUVr) &= \sigma_{\Delta SUVr}^2 = (*) + (**) - 2 \times (\#) \\
 &\approx \left( \frac{1}{\mu_R^2 + \sigma_R^2} + \frac{2\sigma_R^2(\sigma_R^2 + 2\mu_R^2)}{(\mu_R^2 + \sigma_R^2)^3} \right) (\sigma_Z^2 + \sigma_\epsilon^2 + \alpha_1^2) - \frac{\alpha_1^2}{\mu_R^2} \left[ 1 + \left( \frac{\sigma_R}{\mu_R} \right)^2 \right]^2 + \\
 &\quad \left( \frac{1}{\mu_R^2 + \sigma_R^2} + \frac{2\sigma_R^2(\sigma_R^2 + 2\mu_R^2)}{(\mu_R^2 + \sigma_R^2)^3} \right) (\sigma_Z^2 + \sigma_\epsilon^2 + \alpha_2^2) - \frac{\alpha_2^2}{\mu_R^2} \left[ 1 + \left( \frac{\sigma_R}{\mu_R} \right)^2 \right]^2 - \\
 &\quad 2 \times \left\{ \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{\mu_R^2 + 2\rho\sigma_R^2} \left[ 1 + \frac{2\mu_R^2\sigma_R^2(1-\rho) + 4\rho\sigma_R^2(2\mu_R^2 - \rho\sigma_R^2)}{(\mu_R^2 + 2\rho\sigma_R^2)^2} \right] - \frac{\alpha_1 \alpha_2}{\mu_R^2} \left[ 1 + \left( \frac{\sigma_R}{\mu_R} \right)^2 \right]^2 \right\} \\
 &= \left( \frac{1}{\mu_R^2 + \sigma_R^2} + \frac{2\sigma_R^2(\sigma_R^2 + 2\mu_R^2)}{(\mu_R^2 + \sigma_R^2)^3} \right) (2\sigma_Z^2 + 2\sigma_\epsilon^2 + \alpha_1^2 + \alpha_2^2) - \\
 &\quad \frac{\alpha_1^2}{\mu_R^2} \left[ 1 + \left( \frac{\sigma_R}{\mu_R} \right)^2 \right]^2 - \frac{\alpha_2^2}{\mu_R^2} \left[ 1 + \left( \frac{\sigma_R}{\mu_R} \right)^2 \right]^2 - \\
 &\quad 2 \times \left\{ \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{\mu_R^2 + 2\rho\sigma_R^2} \left[ 1 + \frac{2\mu_R^2\sigma_R^2(1-\rho) + 4\rho\sigma_R^2(2\mu_R^2 - \rho\sigma_R^2)}{(\mu_R^2 + 2\rho\sigma_R^2)^2} \right] - \frac{\alpha_1 \alpha_2}{\mu_R^2} \left[ 1 + \left( \frac{\sigma_R}{\mu_R} \right)^2 \right]^2 \right\} \\
 &= \left( \frac{1}{\mu_R^2(1 + CV_R^2)} + \frac{2CV_R^2(2 + CV_R^2)}{\mu_R^2(1 + CV_R^2)^3} \right) (2\sigma_Z^2 + 2\sigma_\epsilon^2 + \alpha_1^2 + \alpha_2^2) - \\
 &\quad \frac{(\alpha_1 + \alpha_2)^2}{\mu_R^2} \left[ 1 + CV_R^2 \right]^2 - \\
 &\quad 2 \times \frac{1 + 2CV_R^2 + 10\rho CV_R^2}{(1 + 2\rho CV_R^2)^3} \frac{\alpha_1 \alpha_2 + \sigma_Z^2}{\mu_R^2}, \\
 &\quad \text{where } CV_R^2 \equiv \frac{\sigma_R^2}{\mu_R^2}
 \end{aligned}$$

So if we have patients  $1, \dots, n$ , then we will have  $\Delta SUVr^1, \dots, \Delta SUVr^n$  i.i.d, and the statistic of interest will be  $\Delta \bar{SUVr} = \frac{1}{n} \sum \Delta SUVr^i$ , and the p-value behaves like

$$\frac{E[\Delta SUVr]}{\sqrt{\frac{Var(\Delta SUVr)}{n}}}$$

### B.4.2 $\Delta$ -Model

For the  $\Delta$ -model, we have

$$\Delta T^i = (\alpha_2 - \alpha_1) + \beta \Delta R^i + \epsilon^i$$

For notation simplicity, let us denote the above formula as

$$Y = \Delta \alpha + \beta X + \epsilon$$

And from our assumptions we have  $X \sim N(0, 2(1 - \rho)\sigma_R^2)$  and  $\epsilon \sim N(0, 2\sigma_\epsilon^2)$ .

Since least squared estimate is unbiased, we have

$$E[\hat{\Delta}\alpha] = \Delta\alpha$$

And

$$\begin{aligned} \text{Var}(\hat{\alpha}) &= \text{Var}\left(\underbrace{E(\hat{\alpha}|X)}_{=\alpha, \text{ since unbiased}}\right) + E(\text{Var}(\hat{\alpha}|X)) \\ &= E(\text{Var}(\hat{\alpha}|X)) \\ &= E\left[\frac{2\sigma_\epsilon^2}{n} \frac{\sum X_i^2}{\sum (X_i - \bar{X})^2}\right] \\ &= \frac{2\sigma_\epsilon^2}{n} E\left[\frac{\sum X_i^2}{\sum (X_i - \bar{X})^2}\right] \end{aligned}$$

Now

$$\begin{aligned} \sum \left(\frac{X_i}{\sqrt{2(1-\rho)\sigma_R^2}}\right)^2 &\sim \chi_n^2 \\ E\left[\sum X_i^2\right] &= 2(1-\rho)\sigma_R^2 E\left[\sum \left(\frac{X_i}{\sqrt{2(1-\rho)\sigma_R^2}}\right)^2\right] \\ &= 2n(1-\rho)\sigma_R^2 \end{aligned}$$

And it is well known that

$$E\left[\sum (X_i - \bar{X})^2\right] = 2(1-\rho)(n-1)\sigma_R^2$$

Also

$$\begin{aligned} \text{Cov}\left(\sum X_i^2, \sum (X_i - \bar{X})^2\right) &= \text{Cov}\left(\sum (X_i - \bar{X})^2 + n\bar{X}^2, \sum (X_i - \bar{X})^2\right) \\ &= \text{Var}\left(\sum (X_i - \bar{X})^2\right) \\ &\quad \text{since sample mean and sample variance are uncorrelated} \\ &= 4(1-\rho)^2\sigma_R^4(n-1) \end{aligned}$$

So

$$\begin{aligned} E\left[\frac{\sum X_i^2}{\sum (X_i - \bar{X})^2}\right] &\approx \frac{2(1-\rho)n\sigma_R^2}{2(1-\rho)(n-1)\sigma_R^2} + \\ &\quad \frac{1}{4(1-\rho)^2\sigma_R^4(n-1)^2} [4(1-\rho)^2\sigma_R^4(n-1) \left[\frac{2(1-\rho)n\sigma_R^2}{2(1-\rho)(n-1)\sigma_R^2}\right] - \\ &\quad 8(1-\rho)^2\sigma_R^4(n-1)] \\ &= \frac{n}{n-1} + \frac{n}{(n-1)^2} - \frac{2}{(n-1)} \\ &= \frac{n-2}{n-1} + \frac{n}{(n-1)^2} \end{aligned}$$

Thus

$$Var(\hat{\Delta\alpha}) \approx \frac{2\sigma_\epsilon^2}{n} \left[ \frac{n-2}{n-1} + \frac{n}{(n-1)^2} \right]$$

As one can see

- First of all,  $E[\hat{\Delta\alpha}] = \Delta\alpha$ , and does not depend on any parameters whereas  $E(\Delta SUVr) = (\alpha_2 - \alpha_1)(\frac{1}{\mu} - \frac{\sigma_R^2}{\mu^3})$  also depends on the mean and variances of the reference SUV. If there's too much variation in reference, the expected value might change signs.
- $Var(\hat{\alpha})$  depends only on  $\sigma_\epsilon$  and  $n$ . Whereas  $Var(\Delta SUVr)$  is a complicated function involving  $\sigma_\epsilon, \sigma_R, \sigma_Z$ , and  $\mu$ .