Title

Methodology development in medical and genomics data

Author

Vernon, Zoe

Publication Date

2021

Peer reviewed|Thesis/dissertation

Methodology development in medical and genomics data

by

Zoe Vernon

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, Co-chair
Professor Peter Bickel, Co-chair
Professor Sandrine Dudoit

Fall 2021

Methodology development in medical and genomics data

Abstract

Methodology development in medical and genomics data

by

Zoe Vernon

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Haiyan Huang, Co-chair

Professor Peter Bickel, Co-chair

Data that quantify various aspects of medicine and biology are constantly improving and changing. As new techniques become available, opportunities arise to improve our understanding of diseases and other biological properties. With these ever changing data modalities new statistical techniques are required to do proper analysis. In this thesis we analyze, and develop new methods when necessary, for three types of biological data, bulk and single cell RNA-sequencing as well as magnetic resonance imaging (MRI).

First, we develop a statistic for analyzing high-throughput RNA-sequencing data in the context of drug discovery. Changes in gene expression between disease and normal tissues can be used to understand the genomic signature of a disease. When those diseased cells are exposed to a large number of drugs and other perturbations, we can systematically search for the perturbations that reverse the expression of the genes which are altered in the disease. In Chapter 2, we present a new method for quantifying this relationship between disease and drug that outperforms existing methods in simulation, decreases computation time, and is comparable in real data. Additionally, we show an improved ability to quantify the involvement of individual genes in effective drugs. An accompanying software package makes this method easily applicable to new data.

In the second study we extend an existing semi-supervised learning method called GeneFishing for application to single cell RNA-sequencing data (scRNA-seq). Single cell data has unique properties that make its analysis different from the bulk data used in Chapter 2. While direct application of GeneFishing was not always possible in scRNA-seq, due to dissimilar sources of variation and a marked increase in technical noise, the provided modifications allowed for the analysis to be possible. In addition to using GeneFishing as an effective gene prioritization method in single cell data, we show that it can be used as a way to understand how to measure gene-gene co-expression and as a context specific feature selection

method for downstream analyses. Again, we present the accompanying software package, scGeneFishing, for performing GeneFishing in a variety of datasets, including scRNA-seq.

Finally in Chapter 4 of this dissertation we analyze a different type of medical data, brain images. We use MRI scans of patients with primary progressive multiple sclerosis to study the association of regional brain volume at baseline with disease progression. We find that statistics summarizing volume in pre-defined regions of interest (ROIs) are not more predictive of progression than using traditional clinical and MRI variables. However, by deriving data-driven ROIs through voxel-level clustering we are able to achieve better predictive performance.

To my family and friends

Contents

Contents **ii**

1 Introduction **1**

2 Local rank-pattern based reverse correlation: a new statistic for leveraging genomic data in drug discovery **4**
2.1 Introduction 4
2.2 Data and problem formulation 6
2.3 **Lo**cal rank-pattern based reverse **cor**relation: LoCor 10
2.4 Simulations 19
2.5 Real data results 24
2.6 Run time 30
2.7 Discussion 32

3 GeneFishing in scRNA-sequencing data and its applications **34**
3.1 Introduction 34
3.2 Methods 36
3.3 Results 40
3.4 Discussion 59

4 Using regional volumetric data to predict clinical progression in multiple sclerosis **62**
4.1 Introduction 62
4.2 Data 63
4.3 Methods and results 65
4.4 Discussion and Future work 73

Bibliography **75**

A Supplemental material for "Local rank-pattern based reverse correlation" **84**
A.1 Gene-wise statistics theory 84
A.2 Top compounds 91
A.3 Top genes 95

B Supplemental material for "GeneFishing in scRNA-sequencing data and its applications" 98
B.1 Enrichment analysis results 98

C Supplemental material for "Using regional volumetric data to predict clinical progression in multiple sclerosis" 106

Acknowledgments

Thank you so much to my advisors Haiyan Huang and Peter Bickel. Your advice and support are greatly appreciated. I truly could not imagine a better pair of mentors. I am eternally grateful for all the time you spent with me, teaching, encouraging, and supporting me through this experience.

Thank you to my parents, Jane and David, and my siblings, Molly and Nate. I love you all. Thanks for believing in me and for all of the support. I am lucky to have such an awesome family.

Thank you to Sandrine Dudoit and Will Fithian for being on my qualifying exam committee and to Sandrine for reading my dissertation. I would also like to thank Sandrine, Koen Van den Berge, and Elizabeth Purdom for the advice on single data analysis during our many meetings. The ideas that arose during those discussions were critical to the work in Chapter 3 of this thesis.

A special thank you to Yuting Ye who was essential to the work in the Chapter 2.

I would also like to thank Zhuang Song, Thomas Bengtsson, Rick Carano, and Anithapriya Krishnan at Genentech for their support during my internship and subsequent collaboration that is included in Chapter 4.

I have made many great friends while at Berkeley. Thanks to Dan Soriano, Ella Hiesmayr and Tiffany Tang for the friendship and countless trips to get food and Salt and Straw. And to Olivia Anguili for being a great problem set partner and better friend. The first few years at Berkeley would not have been possible without our friendship.

To all the friends I made playing basketball, thank you. From pickup at the RSF, intramurals, Pick Her Up and coaching, I am grateful for escape those spaces provided.

My experience at Berkeley was also enhanced by the wonderful staff in the department, La Shana Porlaris, Mary Melinn, and Laura Slakey, that made navigating graduate school easier.

Chapter 1

Introduction

There is a long-standing symbiotic relationship between biological data and statistics. The statistical techniques that are created for the analysis of biological data allow researchers to solidify and expand our knowledge of living organisms. Additionally, many ideas and technique designed originally for biological data are now used in analysis in other fields. Until recently most data describing biology were relatively small, however with modern technologies this is no longer true. Methods for collecting medical and genomic data are rapidly changing and improving. As new data sources become available and increase in size the requisite for improved methodology to do sound statistical analysis in biology is expanding as well. There is often a need for modifications to existing analysis techniques or the development of new methods altogether, to analyze the increasingly high-dimensional data.

One class of data that has been revolutionary to the field of computational biology, both for methodology development and biological knowledge, is next generation sequencing (NGS). With next generation sequencing it is now possible, and cost effective, to analyze molecular features such as the transcriptome and proteome. There are a wide variety of sequencing platforms and applications, including data types such as chromatin immunoprecipitation sequencing (ChIP-seq), which quantifies DNA binding and ATAC-seq that measures regions of open chromatin along DNA. These data, and data from other NGS technologies, make it possible develop models of regulatory mechanisms with neural networks [115] or determine the epigenetic marks associated with tumor reoccurence in prostate cancer [86]. More details, data types, and applications are included in these reviews [90, 98, 80].

Although, not as recently developed as high throughput sequencing technologies, magnetic resonance imaging (MRI) is another data modality that has important applications in biology and medicine, primarily in disease diagnosis and staging. With MRI scans it is possible to view internal organs, such as the brain, without invasive procedures. Similar to sequencing technology, the applications of MRIs continue to expand with new capabilities such as function MRIs, which detect brain activity through blood flow.

It is in this context that this thesis arises. The constantly changing data modalities that characterize the inner workings of cells and organs necessitate careful and computationally

efficient analyses. These methods must account for the structure and unique aspects of the data that is being analyzed. Methods that are good for one type of data may provide misleading or undesirable results on others. With that in mind we present three studies that use data from either next generation sequencing or MR imaging data to learn novel biology.

The remainder of the introduction contains additional context and motivation for the individual chapters in this thesis.

Systematic drug discovery with bulk RNA-sequencing data

One application of the aforementioned next generation sequencing is bulk RNA-sequencing (RNA-seq), which measures the average expression of genes in a sample of cells. Bulk RNA-seq has a broad set of applications, including, characterizing diseases by comparing expression between diseased and healthy tissues. Differential expression analyses that search for genes with significantly higher, or lower, expression in the disease allow researchers to understand what genomic changes are driving disease.

In this chapter we use bulk RNA-seq data to systematically sort through thousands of potential therapeutics for a disease based on the genomic signature of the disease and the drug. We compare the disease differential expression to the differential expression induced by the drug, when comparing samples exposed to and not exposed to the drug, and look for the therapies that alter the expression of disease associated genes. Specifically, the drugs with potential to be viable treatments reverse the expression changes that are seen when comparing disease and healthy tissue.

This framework has successfully been used for drug discovery previously using methods such as the Connectivity Map [52] and sRGES [23]. However, both these techniques are computationally expensive and have theoretical inconsistencies that we aim to address. In Chapter 2 we present a new statistic for computing the described negative dependency that improves computation time while maintaining power in both simulation and real data analysis. We also present an accompanying R package, `LoCor`, that allows researchers to perform the analysis with their data.

Understanding relationships among genes in scRNA-sequencing data

The bulk RNA-sequencing data, like what is used in Chapter 2, has led to many important discoveries, but because expression is averaged over a sample it cannot be used to study cellular heterogeneity. In the last decade new technology that sequences RNA, and other molecular features, at the single cell level has exploded in popularity [103, 44]. This data measures the quantity of molecules in individual cells and thus allows researchers to understand dynamics that were otherwise impossible when looking at average expression. For example, this data are used to investigate cellular differentiation as well as intratumor heterogeneity in cancer [14].

In this chapter, we focus on single cell RNA-sequencing (scRNA-seq) data, which quantifies the expression of genes at the cellular level. As is the case with most single cell data, scRNA-seq has unique properties that make analysis difficult. The most challenging aspect of this data are the high levels of noise and sparsity. Additionally, the sources of variation within a sample of cells differs from that in bulk data. Due to the intricacies of single cell data there has been a wave of computational tools specially intended for analysis of scRNA-seq.

Many of these methods are for cell level analyses, such as clustering and labeling cell types and pseudo-time ordering. However, there is comparatively less work in developing gene networks from single cell data. In Chapter 3 we aim to further develop our understanding of measuring gene-gene similarity in scRNA-seq while modifying a bulk gene network technique, called GeneFishing, to work well in this data. The extensions to GeneFishing allow us to rank similarity metrics performance in a scRNA-seq dataset. Finally, we show how GeneFishing can be used as a context specific feature selection method for improved downstream analyses. In addition, to these interesting discoveries in scRNA-seq we provide an R package, `scGeneFishing`, that will allow users to perform GeneFishing on a variety of data. The original GeneFishing software is contained in an R script [41], as well as a python package [54]. `scGeneFishing` increases the usability of the existing GeneFishing software and adds additional functionality for use in scRNA-seq.

Using MRI scans to predict progression in multiple sclerosis

In the final chapter of this dissertation we transition from sequencing data to analyze data derived from MR imaging. Imaging data is high dimensional, and similar to single cell data, often requires well thought out pre-processing steps to be used in downstream analyses. One such processing technique is called deformation based morphometry (DBM) [10]. Using DBM, MRI scans are mapped to a common reference image and the amount that each voxel must be shrunk or expanded to match the reference is recorded.

This technique produces data on a common scale, allowing the comparison of regional volume across different time points and individuals. This regional volumetric data it opens the possibility to understand how the spatial distribution of atrophy is associated with progression of neurodegenerative diseases.

In the final chapter of this thesis, Chapter 4, we analyze DBM data from a mulitple sclerosis (MS) clinical trial to predict disease progression in individuals with relapsing remitting MS. RRMS is a disease course where patients experience onsets of new neurological symptoms followed by periods of relief.

Over many years the disability accumulates, however the volatility of symptoms and long term nature of progression makes it difficult to prognosticate advancement of the disease. In this chapter, we cluster the voxels of the DBM images to create data-driven regions of interest (ROIs). We find that these regions were anatomically meaningful and more predictive of progression than using predefined atlas based ROIs.

Chapter 2

Local rank-pattern based reverse correlation: a new statistic for leveraging genomic data in drug discovery

Co-authored with Yuting Ye

2.1 Introduction

Traditional drug discovery strategies rely on leveraging biological knowledge about a known target or system to design small molecules for treatments. However, these development processes are costly and on average only 9.6% of drugs that make it to Phase I clinical trials are approved [67]. Given the high cost and low success rate of these target and systems-based approaches, using computational methods to search through existing drugs and known molecules as potential therapies is becoming popular [22]. Researchers can utilize high-throughput omics data to rank thousands of candidates in a cost effective manner without *a priori* knowledge of a target.

This is made possible by modern sequencing technology which has enabled diseases to be characterized in terms of molecular features, such as the transcriptome and proteome [40, 65, 35, 117]. In addition to these ever expanding disease databases, quantification of molecular activity in the presence of drugs and other perturbations are being collected at a rapid pace [53, 108, 1, 93]. This presents researchers with the opportunity to leverage these two types of data to connect individual diseases with drugs based on their molecular signatures.

The hypothesis behind this omics-based method of drug discovery is that compounds which favorably alter the molecular signature of the disease are more likely to be therapeutic. In particular, most studies have relied on transcriptomic data, which is also the focus of this paper. In that case the goal is to find drugs which tamp down the expression of genes that

are over-expressed in disease as compared to normal tissue, and ramp up the expression of those that are under-expressed in disease as compared to normal tissue. This is done by computing dependency between the disease differential expression (DE), which looks at changes in gene expression between healthy and diseased tissue, and the drug DE, which compares expression between untreated samples and samples treated with a drug or other type of perturbagen.

Studies based on the aforementioned hypothesis, have been successfully applied in a number of ways, including, discovering new potential therapeutics, finding avenues for repurposing existing drugs, and/or uncovering mechanisms of action that can be used for employing more traditional drug discovery strategies [76, 68]. Namely, novel drug candidates have been found in lung cancer [79] and cancer mestatasis [102], among other diseases. There have also been success stories utilizing this approach to find new applications of existing drugs for diseases like muscle atorophy [51], inflammatory bowel disease [30] and lung cancer [88, 111]. Additionally, biologists have used these computational approaches to generate hypotheses about new mechanisms of action in various cancers, including prostate[37] and ovarian [82] cancers. It has also been seen that this computational approach can find treatments that reverse physiological markers of dyslipidemia in mouse models [101] and recover a limited number of known drug indications [70, 26, 101].

Given, that we want to find drugs which reverse the changes found in disease, the statistical goal is to quantify the relationship between disease and drug by measuring negative dependency between the two vectors of differential expression. Notice that this dependency is not global, as we primarily care about reversing expression in the small set of genes that are at the extremes of the disease DE (e.g. genes that are significantly over- or under-expressed in the disease). On the other hand, reversal of genes that show little difference in expression when comparing diseased to healthy tissues is unimportant. Standard methods of measuring dependency, such as or Pearson's or Spearman's rank correlation, are designed to compute dependency over entire lists and thus are not applicable for detecting the described structure. The most common method for computing this type of local dependency in the context of drug discovery is the Connectivity Map (CMap) [52] and various modifications [118, 33, 23].

CMap based methods determine whether the up- and down-regulated disease genes are over-represented at the top or bottom of the ordered drug DE. Where up- and down-regulated genes are the genes which are over- and under-expressed, respectively, above some threshold on the log fold change of the disease DE. Despite attempts to systematically determine the appropriate cutoff values [107], the choice is relatively arbitrary, and can lead to potentially useful information being discarded about the genes which are not classified as up- or down-regulated. Additionally, this thresholding results in statistics that require permutation tests for assessing significance, which is computationally expensive.

We provide a rank based count statistic, called Local rank-pattern based reverse Correlation (LoCor), for quantifying the relationship between drug and disease that does not rely on employing a cutoff to the disease differential expression. We use LoCor to measure gene importance in drugs that reverse expression to provide potential biomarkers for future drug discovery efforts. We evaluate the performance of LoCor in simulation and in real data

and find that LoCor improves computation time while still enabling detection of the desired local structure in real data and simulation. LoCor also improves identification of the genes involved in the reversal for effective drugs over existing methods. Given the vast amount of data that is increasingly available this boost in computational efficiency is important and the accompanying R package allows for easy implementation for researchers in this field. As drug and disease differential expression data become widely available for additional molecular features this method could be readily extended to be used in that context and, with slight modifications, in combination with the gene expression data discussed in this paper.

2.2 Data and problem formulation

As mentioned in Section 2.1, the relationship between disease and drug is determined by utilizing differential expression between disease and control samples as well as between treated and untreated samples. This is depicted in Figure 2.1. The two data sources used in this paper are The Cancer Genome Atlas (TCGA) [106], to quantify the molecular signature of the disease, and The Library of Integrative Network Based Cell Signatures LINCS-L1000 database [53, 93] to quantify the changes in gene expression in the presence of perturbagens for 978 landmark genes. The perturbagens, which we will interchangeably refer to as "drugs", in this database are primarily small molecules, but they include additional categories such as ligands and CRISPR knockdown experiments.

In order to understand the relationship between gene and disease we utilize the differential expression profiles from the sRGES paper [23] for breast invasive carcinoma (BRCA), colon adenocarcinoma (COAD) and liver heptocellular carcinoma (COAD). The differential expression analysis was done using DESeq V1 [6] with data from TCGA. These three cancers were chosen because they have adequate representation in both TCGA and the other data used in our study. The resulting disease differential expression profiles indicate changes in gene expression between disease and normal tissue for over 20,000 genes. In particular, genes that tend to be significantly up- or down-regulated in the disease as compared to the normal tissue are likely to be important to the disease.

To understand the relationship between drug and disease, through gene expression, we utilize data from the LINCS-L1000 database. LINCS is a project which collects gene expression data for 978 genes with and without exposure to perturbagens in a number of cell lines. The perturbagens can be drugs, small molecular compounds, or various genetic perturbations and the 978 genes were selected due to their connection with disease. The resulting drug differential expression profiles measure the changes in the gene expression between cells which have had exposure of a certain dose and duration to one of these perturbagens and unexposed cells. In this report we use the same set of 66,612 samples from the sRGES paper. The samples are made up of over 12,000 unique perturbagens in 71 cell lines, including multiple cell lines in both the breast, colon, and liver, at various doses and duration of exposure.

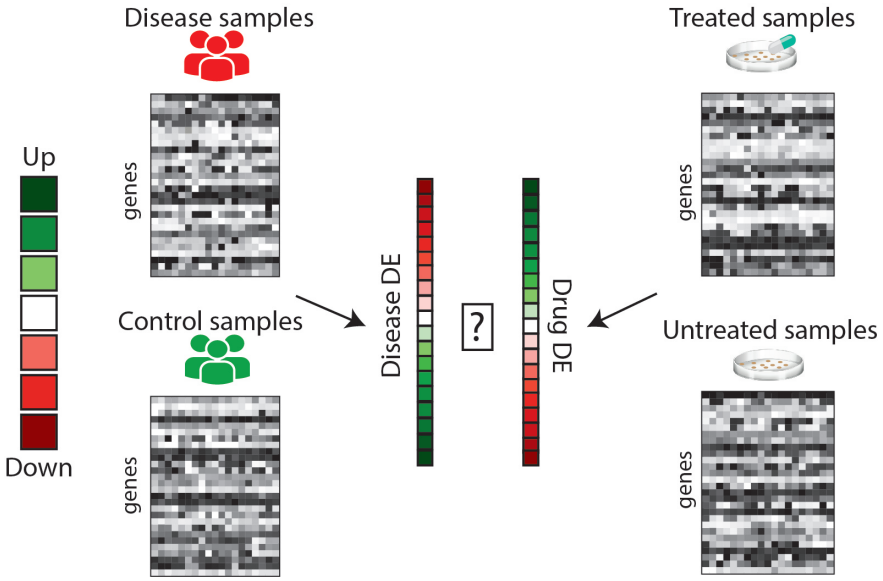


Figure 2.1: Compute disease differential expression and drug differential expression lists by comparing disease samples to control samples and treated samples to untreated samples respectively. The colors in the DE list represent the degree to which the gene is up-regulated (green) in disease or under treatment or down-regulated (red) in disease or under treatment. The relationship between disease and drug DE lists illustrated here is the ideal scenario as all genes are reversed by the drug in question.

Related work

In the context of drug discovery previous methods to compute negative dependency between these two DE vectors, such as CMap, are largely based on the concept of gene set enrichment analysis (GSEA) [94], where GSEA is used to determine whether the up- and down-regulated disease gene sets are randomly distributed in the ordered drug DE list or preferentially fall towards the top and/or bottom. In both CMap and summarized reverse genes expression (sRGES) [23], an extension of CMap which will use as an additional benchmark, the up- and down-regulated genes are determined by applying a threshold on the log fold changes in the disease differential expression profile. Then an enrichment score is computed for the up- and down-regulated disease genes, separately, and combined to get a single score per sample. In this section we discuss in detail how GSEA is used and what issues arise from the resulting statistics.

Assume the drug DE has been measured at n genes for a given sample and assume of those n genes there are a subset of size m_{up} (resp. m_{down}) up-regulated (resp. down-regulated) disease genes. Construct a vector V_{up} (resp. V_{down}) made up of the positions of the m_up (resp. m_down) disease genes in the drug DE list after sorting the list in ascending order. For example, consider $m_{up} = 2$ where the genes with the 8th and 10th largest DE values are

up regulated. We would then have $V_{up} = (8, 10)$. Then compute

$$a_{up} = \max_{1 \le i \le m_{up}} \left\{ \frac{i}{m_{up}} - \frac{V_{up}(i)}{n} \right\} \quad \text{and} \quad b_{up} = \max_{1 \le i \le m_{up}} \left\{ \frac{V_{up}(i)}{n} - \frac{i-1}{m_{up}} \right\}$$

and

$$a_{down} = \max_{1 \le i \le m_{down}} \left\{ \frac{i}{m_{down}} - \frac{V_{down}(i)}{n} \right\} \quad \text{and} \quad b_{down} = \max_{1 \le i \le m_{down}} \left\{ \frac{V_{down}(i)}{n} - \frac{i-1}{m_{down}} \right\}$$

where $V_{up}(i)$ (resp. $V_{up}(i)$) is the value at position i in the vector. The enrichment scores are given by

$$es_{up} = a_{up} - b_{up}$$

and

$$es_{down} = a_{down} - b_{down}$$

The enrichment scores es_{up} (resp. es_{down} represent the absolute enrichment of an up (resp. down) gene list in a given drug DE profile. A large enrichment score (close to 1) is given when the set of up- or down-regulated genes tend to have high ranks (i.e. have large positive DE expression) in the drug DE list while a small score (close to -1) is given when the set of up- or down-regulated genes tend to have low ranks. Then for each sample we obtain a measure of the negative dependency between drug DE and disease DE

$$S^{CMap} = \begin{cases} es_{up} - es_{down} & \text{if sign}\,(es_{up}) = \text{sign}\,(es_{down}) \\ 0 & \text{otherwise} \end{cases}$$

or

$$S^{RGES} = es_{up} - es_{down}$$

It should be noted that in practice and simulation we rarely see $\text{sign}\,(es_{up}) \neq \text{sign}\,(es_{down})$ and thus there is no significant difference between these methods for individual samples. The main difference between these CMap and sRGES arises in the way in which individual scores are combined across samples for a single perturbagen. For most perturbagens, they have been measured in different cell lines at different doses and durations, however in the end the goal is to measure the dependency for a drug or compound, not just a sample. We use the sRGES method to summarize the LoCor sample scores in this paper as it produces optimal results in real data (see Figure 2.12).

Both RGES and CMap weight dependency at the extremes of the disease and drug DE lists more heavily than the middle in two ways (1) by only considering genes that are significantly DE expressed in the disease and (2) in the computation of the enrichment scores. We can see that es_{up} is close to -1 when up-regulated genes are ranked towards the bottom of drug DE list and es_{down} is close to 1 when down-regulated genes are ranked highly and thus the most extreme negative score will be given when up-regulated genes have low ranks and down-regulated genes have high ranks. However, there are some issues with the enrichment

score computation, which we illustrate below in words and examples, that lead to scores that are not consistent with the desired properties.

In these methods, when we threshold the disease DE profile to obtain the up- and down-regulated disease genes we remove all information about the magnitude and significance of their fold changes. The thresholds on disease DE profiles need to be stringent enough that genes that have no true association with disease are excluded, but lenient enough that there is not a loss of important information from genes which are related to disease. This difficulty in choosing a threshold, and the impact on the power of CMap for detecting local dependency, is explored in more detail in simulations in Section 2.4. We find that the statistics are sensitive to the amount of genes which are classified as up- or down-regulated across many scenarios. LoCor does not rely on such a threshold, instead there is a robust parameter which governs a more gradual increase in weighting at the extremes of the DE profiles.

Consider the following toy example, with $m_{up} = m_{down} = 1$. Let the up-regulated gene have the largest negative drug DE and the down-regulated genes have the largest positive drug DE. That is if the drug DE has $n = 1000$ genes we would have $V_{up} = 1000$ and $V_{down} = 1$. This results in $S_{CMap} = S_{RGES} = -1.99$ (a very significant score) as the range of scores is $(-2, 2)$. However, because we do not consider the dependency over the remaining 998 genes it could be that all those genes show a positive dependency. In fact, if we increase the cutoff on p-value such that $m_{up} = m_{down} = 2$ and it happens that now the second most positive DE drug gene is up-regulated and the second most negative DE drug gene is down-regulated we would have $V_{up} = (2, 1000)$ and $V_{down} = (1, 999)$, giving $S_{CMap} = S_{RGES} = -0.001$, which indicates almost no dependency. Of course, this is a somewhat contrived example, but the point remains that the choice of threshold on p-value for the log fold changes can effect the scores dramatically. We will see this in simulation and in the real data later in the paper.

Secondly, there is a lack of comparability between scores from different diseases or molecular features when using GSEA methods. This inability to directly compare scores is because the maximum range of enrichment scores vary depending on the number of genes in the disease signature. In fact, counter to intuition, disease signatures with a small number of genes, all of which are truly reverse, takes on more extreme values than disease signatures with a larger number of genes, again which all are truly reversed.

For example, consider the following two disease signatures both with $m_{up} = m_{down} = 100$. In (1) we have $V_{up} = (501, 502, \ldots, 600)$ and $V_{down} = (401, 402, \ldots, 500)$. There is some negative dependency in this scenario as all up-regulated genes have DE below the median and all down-regulated genes have drug DE above the median, however reversal is happening all in the genes that show little DE after exposure to the drug. In (2) let $V_{up} = (501, 502, \ldots, 545, 946, 947, \ldots, 1000)$ and $V_{down} = (1, 2, \ldots, 55, 456, 458, \ldots, 500)$. Clearly, (2) exhibits stronger reversal that (1), but they have the exact same reversal score (i.e. $S_{CMap}(1) = S_{CMap}(2)$). This is a consequence of the fact that outside of those up- and down-regulated sets the only information we consider from other genes is how they contribute to the position of those sets in the overall list.

An additional way in which comparison of scores from these methods is not possible as the most extreme values the statistics can take depend on the number of genes in the

up- and down-regulated sets as well as the the number of genes in the reference (drug) signature. For example, if $n = 1000$ and we have $V_{up} = 1000$ and $V_{down} = 1$ we saw that $S_{CMap} = S_{RGES} = -1.99$. However, for $V_{up} = (1000, 999, \ldots, 991)$ and $V_{down} = (1, 2, \ldots, 10)$ we get $S_{CMap} = S_{RGES} = -1.981$, which is less extreme than the case with only one up- and down-regulated genes.

A final important difficulty arises in the p-value computation. In both methods p-values are computed based on permutation tests, which is computationally expensive. To get an accurate p-value, for each sample, the disease signature is randomly sampled thousands of times and the CMap or RGES score is recomputed for each of those samples. This reference distribution is then compared to the score from the observed data. In practice, this means that when a user queries a score for a large number of samples they have to wait for a while for these computations to be done in parallel over the internet.

To summarize, both RGES and CMap weight are able detect local dependence at the extremes of the DE lists, however, there are a number of theoretical and computational disadvantages. These difficulties motivated us to develop a more robust statistic with stronger theoretical and computational properties. LoCor, which is described in detail in the next section, does not require such a permutation test, while still maintaining adequate power for detecting the desired local dependency. It allows users to do the computations through an R package on local computers and utilizes information across all genes by naturally weighting negative dependency at the extremes more heavily than dependency in the middle of the drug and disease DE lists.

2.3 Local rank-pattern based reverse correlation: LoCor

We propose a new measure to compute reversal at the extremes called **Lo**cal rank-pattern based reverse **Cor**relation, LoCor. We say LoCor is a rank-pattern based measure of dependency because the core idea is to count decreasing pairs in the ranked drug DE vector after applying the permutation that sorts the disease DE in descending order. The statistic is motivated by the successful rank-pattern based dependency measures from Wang et al. [104, 105].

As is illustrated in the schematic in Figure 2.2, we are counting these decreasing pairs that fall within a (circular) window length of each other. By circular window, we mean that we allow windows to cross from one edge of permuted drug DE vector to the other. As we slide the circular windows along the permuted drug DE vector it allows us to detect local structure that may otherwise be lost with a more global measure of association.

We formally define the statistic below. Assume there are n genes with DE data in both drug and disease. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$ be the disease and drug differential expression profiles respectively. Also, let $l < \frac{n}{2}$ be the number of indices (i.e. the window length) which we will count decreasing pairs within. Additionally, let $\boldsymbol{z} = \sigma(\boldsymbol{x})$

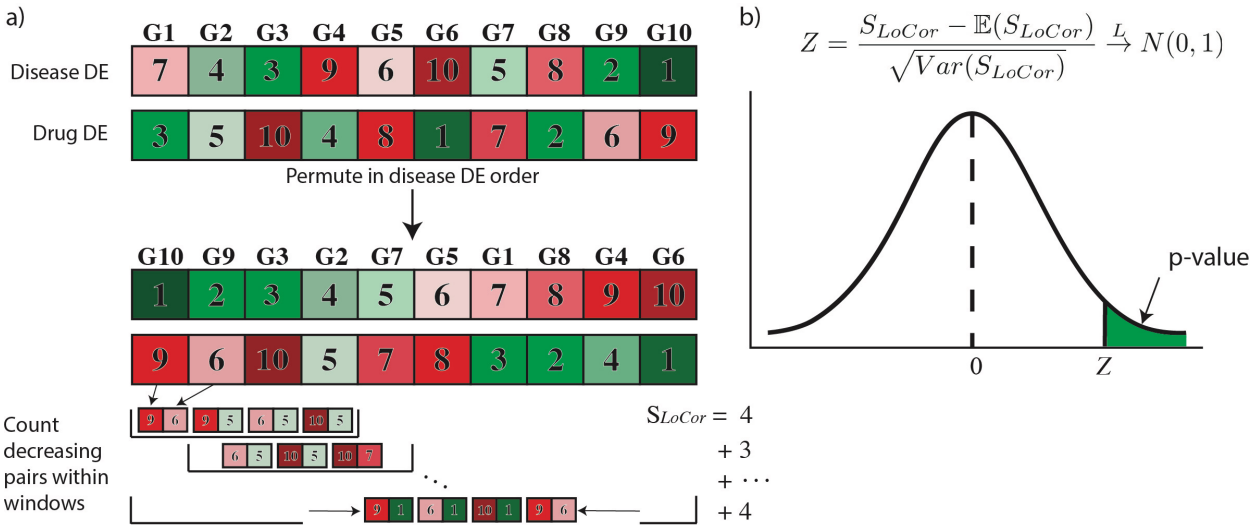


Figure 2.2: a) Compute LoCor by comparing the disease and drug DE lists. The numbers inside the box represent the rank of the differential expression within east list respectively. First, we permute the disease DE list in decreasing order and apply the same permutation to the drug DE list. Next, we count the number of decreasing pairs within a window length $l = 4$. In the first window, including genes G10, G9, G3, and G2, there are four decreasing pairs (genes G10 and G9, genes G10 and G2, genes G9 and G2, and genes G3 and G2). Note, genes G9 and G3 do not form a decreasing pair because gene G9 is less down-regulated than gene G3. The second window starts with gene G9 and ends with gene G7 and has three decreasing pairs. We continue sliding the window until the tenth one, which starts at gene G6 and and includes genes G10, G9 and G3. In that window there are 4 decreasing pairs (genes G6 and G10, genes G6 and G9, genes G6 and G3, and genes G10 and G3). Then to get $S(l)$ we the sum of the counts across all windows. b) After subtracting the mean and dividing by the standard deviation of the null distribution, we have a Z-score which can be used to compute p-values and compare significance across samples.

be the vector that applies the permutation, σ, which sorts the elements of the disease DE vector $\boldsymbol{y}$ in decreasing order to the drug DE vector $\boldsymbol{x}$. That is,

$$\boldsymbol{z} = (z_1, \ldots, z_n) = \left(x_{\sigma^{-1}(1)}, \ldots, x_{\sigma^{-1}(n)}\right).$$

Then the measure of per-sample negative dependency, $S(l)$, is defined as

$$S(l) = \sum_{i=1}^{n} \sum_{j_1 < j_2 : j_1, j_2 \in \mathcal{I}_i^+(l)} \mathbb{I}\{z_{j_1} > z_{j_2}\} \tag{2.1}$$

where $\mathcal{I}_i^+(l)$ are the indices of window starting at position i shifting right by adding 1 modulo n until there are l genes in the window. Let $i' = i - 1$. This can be explicitly written as,

$$\mathcal{I}_i^+(l) = \{i' + 1, (i' + 1) \bmod n + 1, (i' + 2) \bmod n + 1, \ldots, (i' + l - 1) \bmod n + 1\}$$

By summing over all $j_1 < j_2 : j_1, j_2 \in \mathcal{I}_i^+(l)$ we are only counting decreasing subsequences inside the windows of length l defined by the index sets $\mathcal{I}_i^+(l)$. Notice that these windows are circular in nature, in that the index sets $\mathcal{I}_i^+(l)$ with $i > n - l$ include entries from both the top and the bottom of $\boldsymbol{z}$.

For example, if we have window length $l = 10$ and $n = 100$ genes, then the sliding window starting at position $i = 98$ is defined by $\mathcal{I}_{98}^+(10) = \{98, 99, 100, 1, 2, \ldots, 7\}$. Then, because we only sum over $j_1 < j_2$ for $j_1, j_2 \in \mathcal{I}_{98}^+(10)$, we look for all decreasing pairs in the following subset of $\boldsymbol{z}$, $(z_1, z_2, \ldots, z_7, z_{98}, z_{99}, z_{100})$.

The circular nature is a major aspect that helps detect the dependencies that are only local the extremes of the list. If the up- and down-regulated disease genes are truly reversed when we consider these circular windows the entries in $\boldsymbol{z}$ at small indices will be large relative to the entries in $\boldsymbol{z}$ at the large indices leading to a large number of decreasing subsequences and a significant overall score. This allows us to detect the relationship even in presence of large amounts of noise, as is typically the case with genetic data.

From Theorem 1 we know that normalizing by the appropriate mean and variance $S(l)$ is asymptotically standard Gaussian. We show in Section 2.4 that for $n = 1000$, which is the number of genes available in the LINCS data, it is reasonable to use the asymptotic distribution to assess significance of $S(l)$, allowing us to get p-values without using a permutation test.

Theorem 1 *Assume that $l = o(n^{\frac{1}{3}})$, $k = 2$, and*

1. *$\boldsymbol{x}$ and $\boldsymbol{y}$ are independent and have no ties within themselves.*
2. *at least one of $\boldsymbol{x}$ and $\boldsymbol{y}$ has an exchangeable distribution*

For $n \to \infty$

$$\frac{S(l) - \mu}{\sigma} \xrightarrow{d} N(0,1)$$

where $\mu = n\frac{\binom{l}{k}}{k!}$ *and* $\sigma^2 = Var(S(l))$

Proof Define

$$\mathbb{I}_{j_1,\dots,j_k}(\boldsymbol{z}) = \mathbb{I}\left\{z_{j_1} > \cdots > z_{j_k}\right\}$$

and

$$S_i(l) = \sum_{\substack{j_1<\dots<j_k \\ j_1,\dots,j_k \in \mathcal{I}_i^+(l)}} \mathbb{I}_{j_1,\dots,j_k}(\boldsymbol{z}).$$

This allows us to write

$$S(l) = \sum_{i=1}^{n} S_i(l)$$

When no confusion arises, we will respectively use $\mathbb{I}_{j_1,\dots,j_k}$, S and S_i instead of $\mathbb{I}_{j_1,\dots,j_k}(\boldsymbol{z})$, $S(l)$ and $S_i(l)$ in order to simplify notation. Assume $\boldsymbol{x}$ has an exchangeable distribution and that $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent with no ties. This implies $\boldsymbol{z} = \sigma(\boldsymbol{x})$ also has an exchangeable distribution and the ranks can be treated as a random permutation of $\{1, ..., n\}$.

To see the asymptotic distribution of S, we leverage the Stein's method [83] for normal approximation with dependency neighborhoods. Formally,

$$d_W\left(\frac{S-\mu}{\sigma}, Z\right) \leq \frac{D^2}{\sigma^3}\sum_{i=1}^{n}\mathbb{E}\left|S_i - \mathbb{E}S_i\right|^3 + \frac{\sqrt{28}D^{3/2}}{\sqrt{\pi}\sigma^2}\sqrt{\sum_{i=1}^{n}\mathbb{E}\left[S_i - \mathbb{E}S_i\right]^4},$$

where $\mu = \mathbb{E}[S]$, $\sigma^2 = var(S)$, $D := \max_{i\in\{1,\dots,n\}}|N_i| = 2l$ (N_i is the dependency neighborhood for S_i).

Next, we will calculate the desired quanties in the above formula.

We have

$$\mathbb{E}S = \mathbb{E}\left[\sum_{i=0}^{n-1} S_i\right] = n\mathbb{E}[S_i] = n\frac{\binom{l}{k}}{k!}.$$

The variance can be written as

$$var(S) = \sum_i \sum_{i'} cov(S_i, S_{i'}).$$

Now we only investigate $k = 2$. Note that

$$cov(S_i, S_{i'}) = cov(\sum_{\substack{a_1<a_2 \\ a_1,a_2\in\mathcal{I}_i^+(l)}} \mathbb{I}_{a_1,a_2}, \sum_{\substack{b_1<b_2 \\ b_1,b_2\in\mathcal{I}_{i'}^+(l)}} \mathbb{I}_{b_1,b_2}) = \sum_{\substack{a_1<a_2 \\ a_1,a_2\in\mathcal{I}_i^+(l)}} \sum_{\substack{b_1<b_2 \\ b_1,b_2\in\mathcal{I}_{i'}^+(l)}} [\mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{I}_{b_1,b_2} - \mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{E}\mathbb{I}_{b_1,b_2}].$$

Suppose $i < i'$ and $i' - i = A$. When $A \geq l$, $cov(S_i, S_{i'}) = 0$. So we only need to consider $A < l$. Then $\mathcal{I}_i^+(l)$ and $\mathcal{I}_{i'}^+(l)$ overlap with the shared length $B = l - A$. To get a non-trivial covariance between $\mathbb{I}_{a_1,a_2}$ and $\mathbb{I}_{b_1,b_2}$, it must hold that $\{a_1, a_2\}$ and $\{b_1, b_2\}$ share at least one index. We discuss the below scenarios.

- $a_1 = b_1$ but $a_2 \neq b_2$. We have $\mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{I}_{b_1,b_2} - \mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{E}\mathbb{I}_{b_1,b_2} = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$. Now we compute the number of index pairs that satisfy $a_1 = b_1$ but $a_2 \neq b_2$:

$$N_1 = (B-1)\cdot(A+B-2) + (B-2)\cdot(A+B-3) + \cdots + 1\cdot A = \frac{B(B-1)}{6}(3A+2B-4).$$

- $a_2 = b_2$ but $a_1 \neq b_1$. Similarly as above, we have $\mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{I}_{b_1,b_2} - \mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{E}\mathbb{I}_{b_1,b_2} = \frac{1}{12}$ and $N_2 = N_1$.

- $a_1 < a_2 = b_1 < b_2$. We have $\mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{I}_{b_1,b_2} - \mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{E}\mathbb{I}_{b_1,b_2} = \frac{1}{6} - \frac{1}{4} = -\frac{1}{12}$. The number of index pairs that satisfy $a_1 < a_2 = b_1 < b_2$ is

$$\begin{aligned} N_3 &= A\cdot(B-1+A) + (A+1)(B-2+A) + \cdots + (A+B-1)\cdot A \\ &= \sum_{k=0}^{B-1}(A+k)(B-1+A-k) \\ &= \frac{B}{6}[6A^2 + (B-1)(B+6A-2)]. \end{aligned}$$

- $b_1 < b_2 = a_1 < a_2$. We have $\mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{I}_{b_1,b_2} - \mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{E}\mathbb{I}_{b_1,b_2} = \frac{1}{6} - \frac{1}{4} = -\frac{1}{12}$. The number of index pairs that satisfy $b_1 < b_2 = a_1 < a_2$ is

$$\begin{aligned} N_4 &= 1\cdot(B-2) + 2\cdot(B-3) + \cdots + (B-2)\cdot 1 \\ &= \sum_{k=1}^{B-2} k\cdot(B-1-k) \\ &= \frac{B(B-1)(B-2)}{6} \end{aligned}$$

- $a_1 = b_1 < a_2 = b_2$. It is easy to get that $\mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{I}_{b_1,b_2} - \mathbb{E}\mathbb{I}_{a_1,a_2}\mathbb{E}\mathbb{I}_{b_1,b_2} = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$. The number of index pairs that satisfy $a_1 = b_1 < a_2 = b_2$ is $N_5 = \binom{B}{2}$.

Putting the above results together, we can get that

$$\begin{aligned} cov(S_i, S_{i'}) &= \frac{1}{12}\cdot N_1 + \frac{1}{12}\cdot N_2 - \frac{1}{12}\cdot N_3 - \frac{1}{12}\cdot N_4 + \frac{1}{4}\cdot N_5 \\ &= \frac{B[(B-1)(2B+5) - 6A^2]}{72} \\ &\overset{A=l-B}{=} \frac{B(12Bl - 6l^2 - 4B^2 + 3B - 5)}{72}. \end{aligned} \tag{2.2}$$

Particularly, when $i = i'$ or $B = l$, we have

$$var(S_i) = \frac{l(l-1)(2l+5)}{72},$$

which can also be verified by the result of [74] when $k = 2$. Hence,

$$\begin{aligned}
var(S) &= \sum_i \sum_{i'} cov(S_i, S_{i'}) \\
&= \sum_i \sum_{j=-(l-1)}^{l-1} cov(S_i, S_{i+j}) \\
&\overset{\text{by (2.2)}}{=} \sum_i [2 \sum_{j=0}^{l-1} \frac{(l-j)[12(l-j)l - 6l^2 - 4(l-j)^2 + 3(l-j) - 5]}{72} - var(S_i)] \\
&= \sum_i [2 \sum_{j=1}^{l} \frac{j(12jl - 6l^2 - 4j^2 + 3j - 5)}{72} - var(S_i)] \\
&= n \cdot (\frac{l(l+1)(l-1)}{18} - \frac{l(l-1)(2l+5)}{72}) \\
&= n \cdot \frac{l(l-1)(2l-1)}{72}.
\end{aligned}$$

We emphasize that this result is not exactly correct because when $i < l$ or $i > n - l + 1$, $Cov(S_i, S_{i'})$ is much more involved that above due to the selected circular pattern. In this case $\sum_{j=0}^{l-1} cov(S_i, S_{i+j}) = \mathcal{O}(l^4)$ but not $\mathcal{O}(l^3)$. However, it does not affect the order of $var(S)$ if $n \gg l$. So we keep the above form.

It is easy to see that

$$\mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^4 = \sum_{\substack{a_1<a_2 \\ a_1,a_2 \in \mathcal{I}_i^+(l)}} \sum_{\substack{b_1<b_2 \\ b_1,b_2 \in \mathcal{I}_i^+(l)}} \sum_{\substack{c_1<c_2 \\ c_1,c_2 \in \mathcal{I}_i^+(l)}} \sum_{\substack{d_1<d_2 \\ d_1,d_2 \in \mathcal{I}_i^+(l)}} \mathbb{E}(\mathbb{I}_{a_1,a_2} - \frac{1}{2})(\mathbb{I}_{b_1,b_2} - \frac{1}{2})(\mathbb{I}_{c_1,c_2} - \frac{1}{2})(\mathbb{I}_{d_1,d_2} - \frac{1}{2}).$$

To ensure

$$\mathbb{E}(\mathbb{I}_{a_1,a_2} - \frac{1}{2})(\mathbb{I}_{b_1,b_2} - \frac{1}{2})(\mathbb{I}_{c_1,c_2} - \frac{1}{2})(\mathbb{I}_{d_1,d_2} - \frac{1}{2})$$

to be non-zero, it requires $\{a_1, a_2\}, \{b_1, b_2\}, \{c_1, c_2\}, \{d_1, d_2\}$ to satisfy either of the below condidtions:

- Each index pair intersects with at least one of the other index pair, so all the four index pairs are "connected".
- The four index pairs can be divided to two groups. In each group, there are two intersected index pairs. The two groups are disjoint.

The first condition corresponds to $\mathcal{O}(\binom{l}{5}) = \mathcal{O}(l^5)$ possible cases, while the second corresponds to $\mathcal{O}(\binom{l}{3}) \cdot \mathcal{O}(\binom{l}{3}) = \mathcal{O}(l^6)$ cases. So in total $\mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^4 \leq \mathcal{O}(l^6)$. Note that

$$\mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^4 \geq \left\{\mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^2\right\}^2 = \mathcal{O}(l^6).$$

So we have $\mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^4 = \mathcal{O}(l^6)$.

Next, by Cauchy-Schwarz inequality, we have

$$\mathbb{E}\left|S_i - \mathbb{E}S_i(l)\right|^3 \leq \sqrt{\mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^2 \cdot \mathbb{E}\left[S_i - \mathbb{E}S_i(l)\right]^4} = \mathcal{O}(l^{\frac{9}{2}}).$$

Since $\mathbb{E}\left|S_i - \mathbb{E}S_i(l)\right|^3$ must scale at an integer power of l, it follows that $\mathbb{E}\left|S_i - \mathbb{E}S_i(l)\right|^3 = \mathcal{O}(l^4)$. With all the ingredients above, it follows that

$$d_W\left(\frac{S-\mu}{\sigma}, Z\right) \leq C_1 \frac{l^2 \cdot n \cdot l^4}{n^{3/2} l^{\frac{9}{2}}} + C_2 \frac{l^{3/2} \cdot \sqrt{n} l^3}{n l^3} \asymp \frac{l^{\frac{3}{2}}}{\sqrt{n}},$$

where C_1 and C_2 are two positive constants. It indicates $d_W(\frac{S-\mu}{\sigma}) = O(\frac{l^{\frac{3}{2}}}{\sqrt{n}}) \to 0$, when $l = o(n^{\frac{1}{3}})$. ■

Gene-wise statistics

In addition to computing the overall dependency we can determine how much the gene contributes to $S(l)$. This allows us to assess the importance of individual genes to the overall reversal and thus uncover potential biomarkers for drug efficacy in a disease. Below in equation (2) we rewrite $S(l)$ such that $G_i(l)$ is the number of decreasing pairs that include gene at index i, weighted by the number of times that pair falls in the same window.

$$S(l) = \frac{1}{2} \sum_{i=1}^{n} G_i(l) \tag{2.3}$$

where

$$G_i(l) = \sum_{j>i : j \in \mathcal{I}_i^+(l) \cup \mathcal{I}_i^-(l)} w_{i,j}(l) \mathbb{I}\left\{z_i > z_j\right\} + \sum_{j<i : j \in \mathcal{I}_i^+(l) \cup \mathcal{I}_i^-(l)} w_{i,j}(l) \mathbb{I}\left\{z_i < z_j\right\}$$

and the weights are given by

$$w_{i,j}(l) = \begin{cases} l - |i-j| & |i-j| \leq l-1 \\ l - (n - |i-j|) & |i-j| \geq n-l+1 \end{cases}$$

Recall that $\mathcal{I}_i^+(l)$ are the indices of the circular window starting at position i, and we define $\mathcal{I}_i^-(l)$ to be the indices of the window ending at position i and moving backwards modulo n,

$$\mathcal{I}_i^-(l) = \{i'+1, (i'-1) \bmod n+1, (i'-2) \bmod n+1, \ldots, (i'-l+1) \bmod n+1\}$$

This weighting scheme is another reason for the good performance of the statistic. By applying larger weights to the pairs that are close together it allows us to put more emphasis on the subsequences that are close together in a window, but far apart in the $\boldsymbol{z}$ vector (e.g. z_1 and z_n) as opposed to pairs such as z_1 and z_{n-100} which are less likely to be decreasing if the dependency is only at the extremes. As we move towards the middle of the list, where the circular indexing no longer applies, this weighting scheme does not provide any added benefit, but it also does not detract from the power of the statistic as the middle is assumed to be independent.

Interestingly, the asymptotic distribution of the gene-wise statistics, which is described in Theorem 2, varies depending on the position of the gene in the permuted list. Genes at the extremes of the list are asymptotically uniform, while genes in the middle are asymptotically closer to Gaussian, although with heavier tails (See Figures 2.3a and 2.3b).

In Theorem 2, we use define the shift s to be $s = \max\{0, l-1-i, l-n+i\}$. The shift value is the number of indices in $\mathcal{I}_i^+(l) \cap \mathcal{I}_i^+(l)$ which are circular in nature. When s is large, i is close to the edge (e.g. $i \approx 0$ or $i \approx n$) and the gene at index i is included in many windows with circular indexing. When $s = 0$, then i is in the middle of z and the gene at position i is never included in any windows that are cross from one edge of vector to the other.

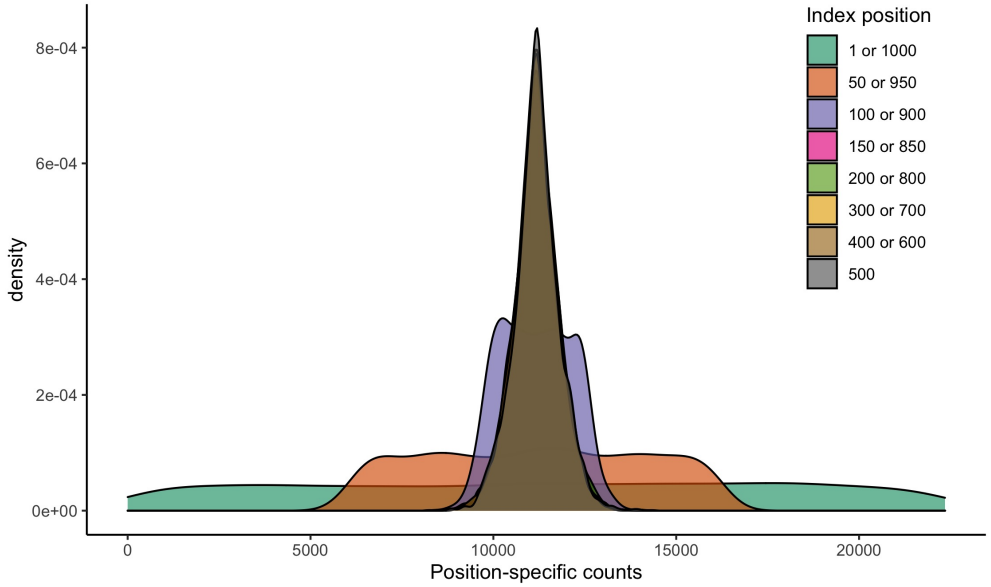
Theorem 2 *Assume that $l < n/2$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent and have no ties within themselves, and at least one of $\boldsymbol{x}$ and $\boldsymbol{y}$ has an exchangeable distribution. Then*

1. *When $s = o(l^{\frac{3}{4}})$, $G_i(l)$ is asymptotically a Gaussian-like random variable.*
2. *When $s = O(l^\alpha)$ with $\alpha > 3/4$, $G_i(l)$ is asymptotically a uniform distribution random variable.*
3. *When $s = O(l^{\frac{3}{4}})$, $G_i(l)$ is asymptotically a random variable from a mixture of a Gaussian-like distribution and a uniform distribution.*

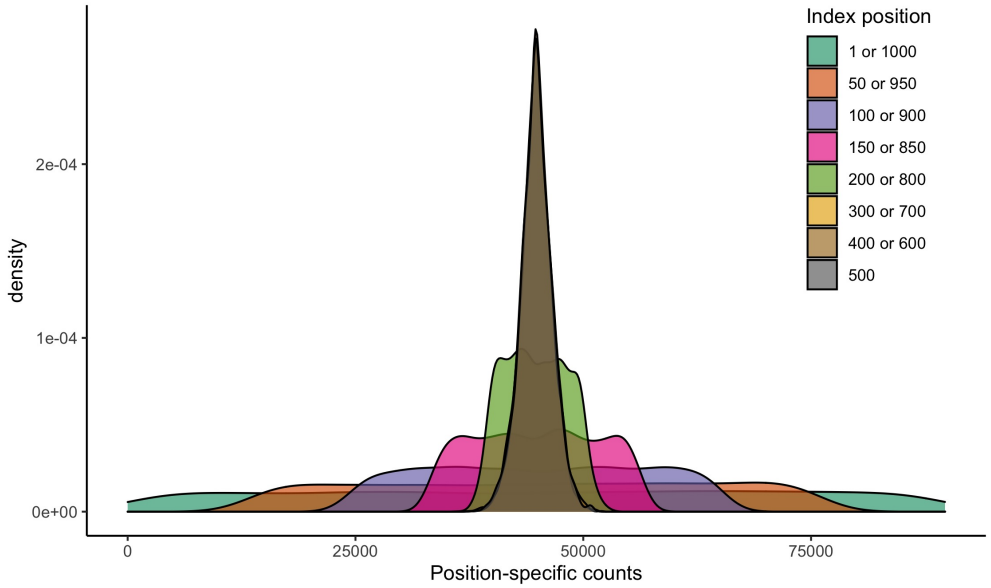
For the finite-sample property, define $\tilde{\sigma}^2 = \frac{l(l-1)(2l-1)}{24}$. we have

1. *If $s = 0$,*

$$\mathbb{P}\left(G_i(l) - \frac{(l-1)^2 + l}{2} \geq g\right) \leq \exp\left(-\frac{g^2}{\tilde{\sigma}^2}\right) \cdot \mathbb{I}(g > 0) + \mathbb{I}(g \leq 0) \tag{2.4}$$



(a) $l = 150$



(b) $l = 300$

Figure 2.3: Density of gene-specific counts $Y_{i,l,s}^{(w)}$ when generating two independent vectors with window length (a) $l = 150$ and (b) $l = 300$. In both we have $n = 1000$. The distribution of the counts depends on the position of the gene i. For example, when $l = 150$, then when $150 \leq i \leq 850$ the distribution is close to Gaussian and for $i = 1$ or $i = 1000$ the distribution is approximately uniform.

2. If $s \neq 0$,

$$\mathbb{P}\left(G_i(l) - \frac{l(l-1)}{2} \geq g | z_i = z\right) \leq \exp\left\{\frac{-(g-\eta(s,z))^2}{\tilde{\sigma}^2}\right\} \mathbb{I}(g \geq \eta(s,z)) + \mathbb{I}(v < \eta(s,z)), \quad (2.5)$$

where $\eta(s,z) = \frac{1}{2}(2z-1)(s^2+|s|)$

These theoretical properties allow us to compute conservative p-values for the each gene in each disease/drug sample pair which serve as normalized gene-wise statistics. We use these gene-wise statistics to assess the importance of each gene by looking for genes which are reversed in potentially therapeutic drugs but not reversed in drugs which are not deemed to have therapeutic potential. See Supplemental Information for details and proofs.

2.4 Simulations

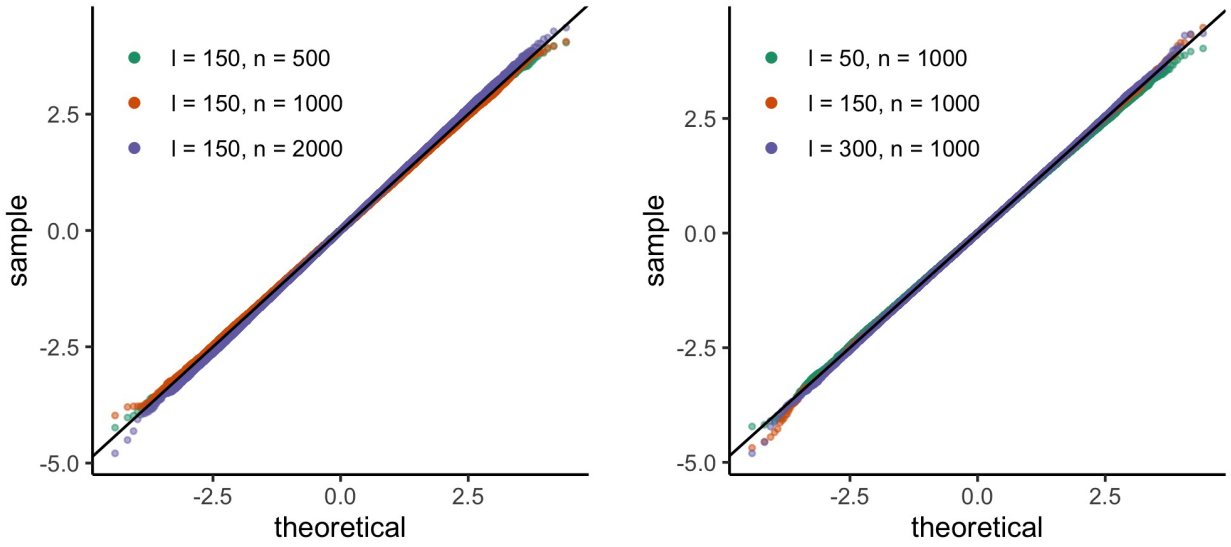
Asymptotic normality of S

To determine the viability of using the asymptotic p-values for computing significance of $S(l)$ we looked at the normality of the normalized $S(l)$ for different numbers of genes n and window lengths l. We simulated 10^5 independent standard normal vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. In Figure 2.4 (a) we increase n, holding $l = 150$ and in Figure 2.4 (b) we hold n constant at 1000 and vary l.

We can see that the normality assumption is reasonable for all depicted combinations of n and l. Note that in Theorem 1 we can only prove asymptotic normality for $l = o\left(n^{1/3}\right)$, however the normality is holding in practice when $l = 300$ and $n = 1000$. This likely indicates a tighter bound in the proof is possible or the constant describing the relationship between l and n is large enough such that in practice we can have $l > n^{1/3}$. Given that these simulations indicate convergence to the asymptotic distribution with $n = 1000$ (the size of the data in this project) for various values of l, we feel confident that the normal distribution results in accurate p-values for assessing statistical significance.

Set up

Given that small number of verified drugs for a given disease benchmarking methods in this field is difficult [45, 25]. For this reason we utilize simulations which are designed to mimic the local structure we hope to detect in order to compare LoCor's performance with CMap and sRGES. To do this we consider simulations of three bivariate relationships. The first such relationship is (a) linear. This is necessary to include in our simulations as most correlation methods are designed to detect such a relationship and thus we need to compare the performance of our statistic in such a scenario. We also investigate performance in



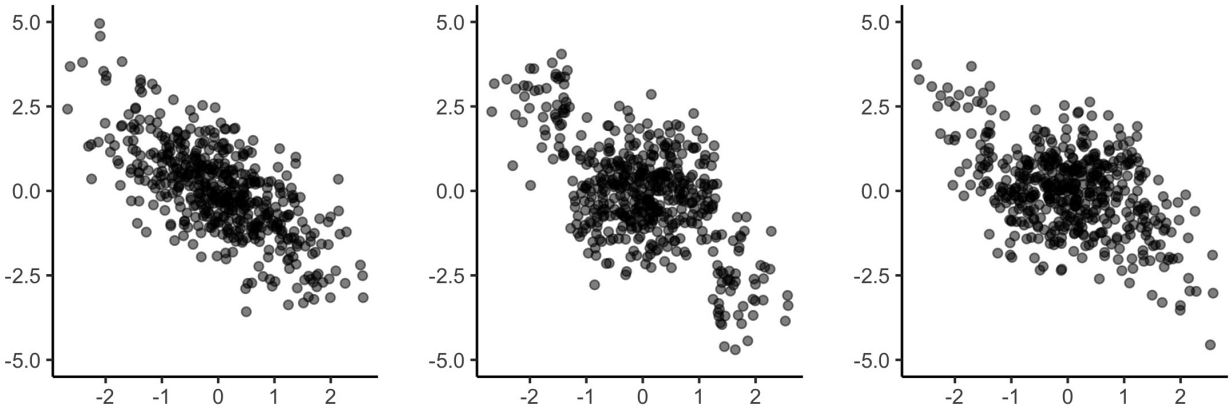
(a) Varying n

(b) Varying l

Figure 2.4: QQ-plots comparing the empirical null distribution of S_{LoCor} to a standard normal distribution. (a) We hold window length constant at $l = 150$ and vary the number of genes n (b) We hold $n = 1000$ constant and vary window length l.

comparison to other methods in relationships that are thought to be more realistic in the context of drug discovery, namely (b) that the reversal occurs at the top and bottom involves genes at the extreme of $\boldsymbol{x}$ appearing at the opposing extreme of $\boldsymbol{y}$ with high probability, and (c) that the linear reversal only occurs at the top and bottom of the gene expression vectors. In both (b) and (c) we varying the amount of reversed genes based on the parameter p, which is the proportion of truly reversed genes. The relationships are summarized in Figure 2.5 and Table 2.1 where $x_i \overset{iid}{\sim} N(0,1)$, $y_i = f(x_i) + \epsilon_i$ and $\epsilon_i \overset{iid}{\sim} N(0,3)$. We compute the power for increasing fractions of replacing ϵ_i with η_i for $\eta_i \overset{iid}{\sim} N(0,8)$.

We used the scenarios in Figure 2.5 to guide our choice of the two tuning parameters, window length l and decreasing subsequence length k. Note that in the previous analyses in Section 2.3 we only present the statistics where we count decreasing pairs ($k = 2$). This is because we found in the simulations that are in the Figure 2.6 that had the highest power for detecting the desired relationships. We also found that choosing $l = 150$ provided balance for detecting local relationships as amounts of dependency at the extremes. Choosing $k = 2$ has the additional advantages of increased computation speed, simplicity, and allows us to defined gene level statistics $S_i(l)$.



(a) Linear (b) Extremes swapped (c) Extremes linear

Figure 2.5: Simulation scenarios we consider to compare the power between methods. (a) is a negative linear relationship with $y_i = x_i + \epsilon_i$ for all $i = 1, \ldots, 1000$. (b) the top and bottom $p = 10\%$ are swapped such that there is an overall negative relationship, but within the extremes the data are independent. (c) negative linear relationship in the top and bottom $p = 10\%$ of the data and independent in the middle. For the purposes of visualization the points generated in these figures have less noise than in our simulation.

Comparison to other methods

We compare the power of our method to Pearson and Spearman's correlation as well as the GSEA based methods CMap and sRGES. Recall that the primary difference between CMap and sRGES is in the method to summarize scores across samples of the same drug. Given that our simulation is focused on the assessing the power to detect the desired patterns of reversal in an individual sample we found that the two methods were equivalent, and thus in all figures we show a single value that corresponds to both methods.

In order to assess the performance of LoCor to previous methods we compared the methods in two ways. First, in Figure 2.7 we assess the sensitivity of the methods hyperparameters by varying the proportions of truly reversed genes p, for scenario (b), as well as the number of up- and down-regulated genes for CMap / RGES and the window length l for LoCor. Second, Figure 2.8 for a single hyperparameter from each model we compare across all three scenarios in Figure 2.5.

We investigate the claim that LoCor is less sensitive to window length than CMap and sRGES are to threshold on the disease signature by considering varying percentages of reversal at the extremes and compare to RGES (same for different window lengths and numbers of up and down regulated genes, which correspond to varying thresholds on the p-values of the log fold changes. Note, that when the reversal occurs in the top $100 \times p\%$ and bottom $100 \times p\%$ of the genes the ideal choice for $m_{up} = m_{down} = np$, while the idea choice of window

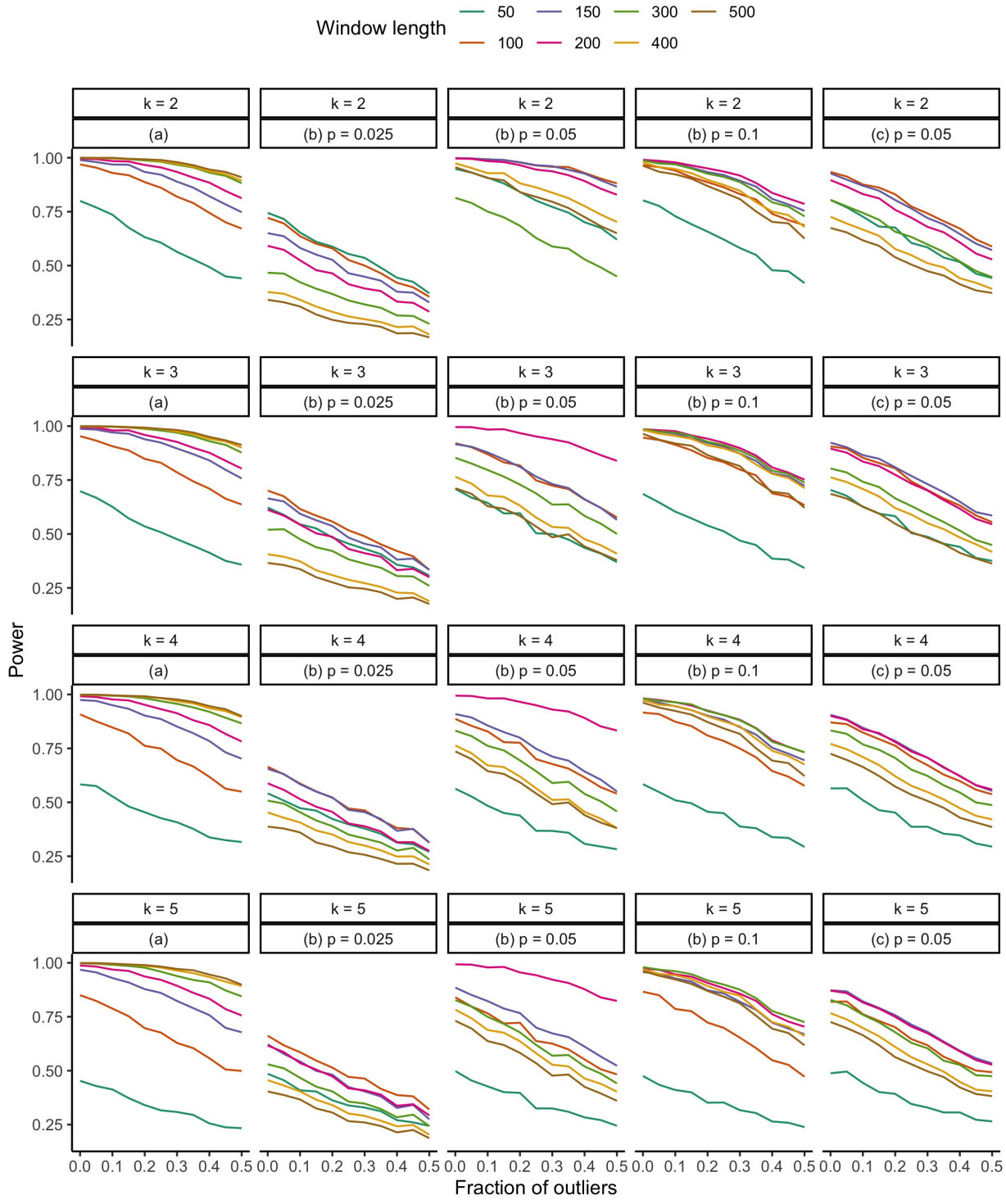


Figure 2.6: Power for the simulation scenarios, subsequence lengths k, and window lengths l.

length $l = 2np$. The results are shown in Figure 2.7.

We can see that at the lowest percentage if we pick exactly the correct number of up and down regulated genes RGES method out performs ours for any choice of l, however at the remaining percentages the optimal choice of l has comparable power to the optimal choice of up/down regulated genes. Another important aspect to note from this plot is that our method is less sensitive to choosing a larger or smaller window size than RGES is to having the correct number of up and down regulated genes. For example in Figure 2.7(b) RGES with 150 and 150 up- and down-regulated genes performs significantly worse that our method with $l = 300$, which is the corresponding window length, even though both are ill-suited as the ideal number of up- and down-regulated genes is 50 and the ideal window size is 100. Similar patterns exist across all three scenarios.

In addition to considering the effect of hyperparameter selection we compared LoCor to other methods and standard correlation methods across simulation scenarios (a)-(c) using $S_{LoCor}(150)$ and the corresponding CMap/RGES with 75 up-regulated and 75 down-regulated genes, as $75 + 75 = 150$. In Figure 2.8 we can see that across these scenarios $S_{LoCor}(150)$ has relatively high power. In scenario (a) as expected Spearman's and Pearson's correlations are able to perform well because the negative dependency is linear and exists across the entirety of the list. However in scenarios (b) and (c), which are the dependencies structures that are of interest to the data discussed in this paper, LoCor outperforms these standard measures of correlation. When comparing to CMap/RGES we are able to maintain power with a single choice of l while the power for different numbers of up- and down-regulated genes falters when those choices are incorrect.

Relationship type	Label	Functional form
Linear	(a)	$y_i = -x_i + 2\epsilon_i$
Top/bottom switched	(b)	$y_i = \begin{cases} \max(x) + 2\epsilon_i & x_i < x_{\lfloor p \times n \rfloor} \\ \min(s) + 2\epsilon_i & x_i > x_{\lfloor (1-p) \times n \rfloor} \\ 2\epsilon_i & \text{otherwise} \end{cases}$
Top/bottom linear	(c)	$y_i = \begin{cases} -x_i + 2\epsilon_i & x_i < x_{\lfloor p \times n \rfloor} \text{ or } x_i > x_{\lfloor (1-p) \times n \rfloor} \\ 2\epsilon_i & \text{otherwise} \end{cases}$

Table 2.1: Simulation scenarios. Here p is the percentage of the simulated genes which exhibit the reversal relationship at each extreme. For example, if $p = 0.05$ and $n = 100$ the top 5 genes and bottom 5 genes are reversed while the remaining 90% of genes are simulated independently.

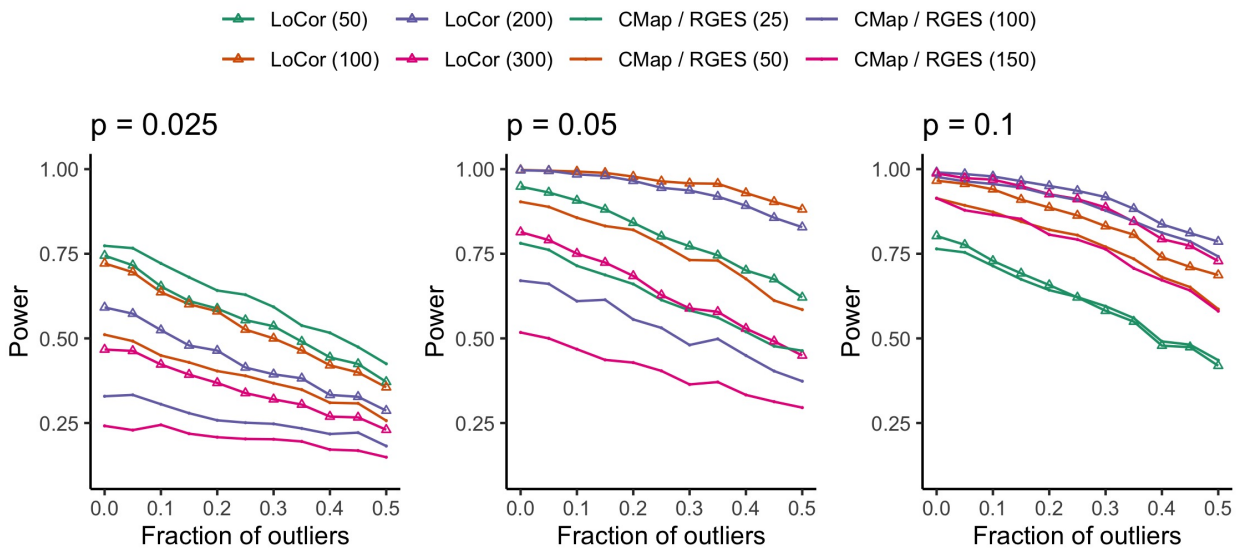


Figure 2.7: Comparison of power for our method and RGES for different fractions of negative dependency at the extremes of the list exhibiting the reversal relationship when genes randomly change sign with high probability. We have (a) top and bottom 2.5%, (b) 5%, and (c) 10%. Lines with triangles are LoCor and the line with a triangle of the same color is the CMap/sRGES statistic with the number of genes included in the computation corresponding to the window size of LoCor.

2.5 Real data results

One practical note is that we had to choose a window length to use for the analysis. We choose $l = 150$ as in simulation it appeared to strike a good balance between reversal across most of the list and reversal that only occurs at small portions of the extremes. By tuning the window length for each disease we can achieve increased performance in validation. However, a priori it is unclear how to choose the window length, so we decided to use a universal window length suggested by our simulations.

Finding compounds

We computed our results using the entirety of the LINCS database to get a score for each of the 66,612 samples and summarized applying the method described in Chen et al. We compare the effectiveness of our method with that of CMap and sRGES in breast, liver and colon cancers. To validate the results we compare the LoCor statistics to median IC_{50} values for each disease from the ChEMBL dataset [15]. IC_{50} measures the amount of a drug that needed to kill 50% of the cells in a sample, thus we will have an idea that LoCor is performing well if we find that drugs with large negative LoCor scores have low IC_{50} values.

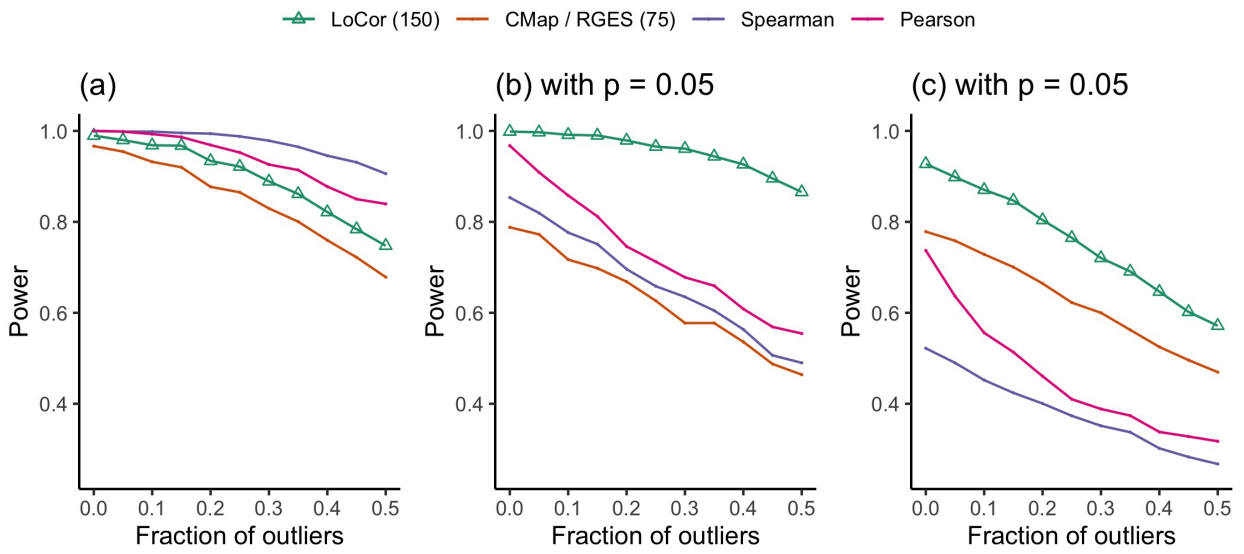


Figure 2.8: Comparison of power for different methods where (a) is the linear dependency across the whole list,(d) has the top and bottom 5% randomly switched and (b) has the top and bottom 5% linearly dependent. In LoCor we choose a window length of $l = 150$ and for CMap / RGES we compute the power with 75 up-regulated and 75 down-regulated genes.

This validation technique limits us to considering drugs that are present in both datasets, of which there are 115 for BRCA, 93 for COAD, and 53 for LIHC.

In previous papers the correlation of drug efficacy statistics with median IC_{50} values have been used as the primary way to compare methods. We computed Pearson and Spearman's correlation between each method and the IC_{50} data. The results are summarized in Figure 2.9. We can see that in all three diseases the correlation there is a positive correlation, while the bars representing the 95% confidence intervals of the correlation show that the only disease with correlations of both types that are significantly different than zero is breast invasive carcinoma. Additionally, notice that although sRGES tends to have the strongest correlation, followed by LoCor, and finally CMap, there is large overlap in the confidence intervals across all the diseases.

In addition, to looking at the correlation with IC_{50} we wanted to try to tease out the differences between the methods in a more nuanced manner. The most important thing is that highly ranked drugs are effective, as those are the drugs that biologists will investigate further. To this end we plotted the percentile rank of each compound with both IC_{50} and LINCS L1000 date for LoCor (x-axis) versus the other methods (y-axis) in Figure 2.10. Each point in this scatter plot is a compound and we use $\log_{10}(IC_{50}) > 4$ as a cutoff to indicate whether the compounds effective or not treating a given disease. This cutoff was chosen as it is the same one used in the sRGES paper. In the figure the different colors represent

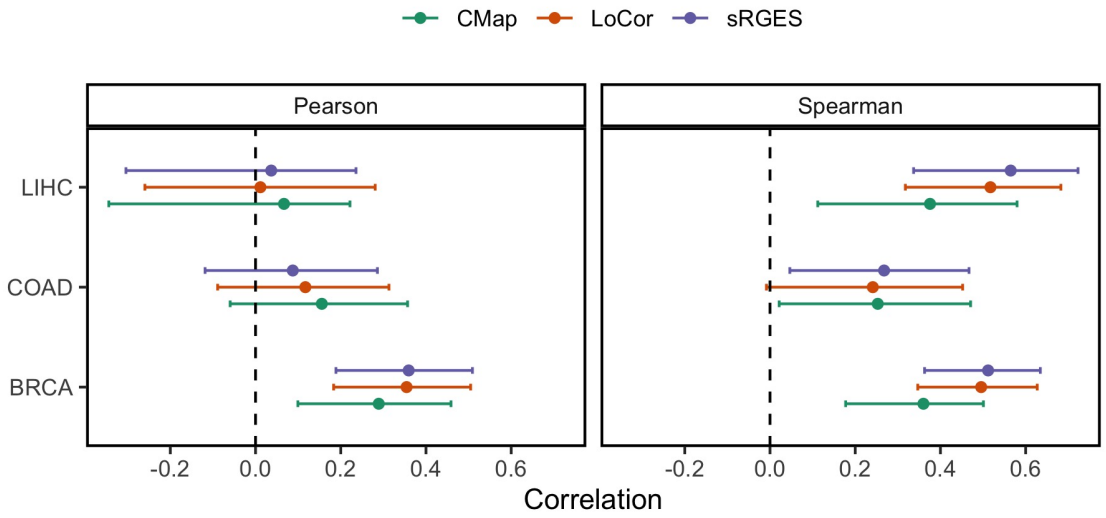


Figure 2.9: Correlation of reversal scores with IC_{50} for drugs with overlapping data.

ineffective and effective label based on IC_{50} and the black line is at $y = x$. Any point below the line indicates a lower rank (e.g. less effective) when comparing sRGES (left) or CMap (right) to LoCor while any point below the line indicates that either sRGES or CMap ranks the compound as more effective than our method.

We can see for the top ranked compounds, particularly in BRCA and COAD, our method tends to ranks effective compounds higher than sRGES (right side) because the compounds are primarily above the $y = x$ line. After the compounds ranked in the top 25% we start to see a mix of ineffective and effective drugs, however as we stated previously, the most important thing is that a method is able to rank effective compounds highly while not ranking ineffective compounds highly. When we look at the left-hand side of the plot to compare LoCor with CMap we see that while there are some highly ranked effective drugs in CMap that LoCor does not rank highly and vice versa. However, we also see a number of ineffective compounds (green points) that fall below the $y = x$ line indicating that CMap is giving high ranking to those compounds while LoCor does not. In summary, Figure 2.10 shows that our method has higher specificity for sorting effective from ineffective compounds than LoCor and while sRGES has similar specificity LoCor is able to consistently rank effective compounds higher than sRGES.

Finding genes

Now that we have discussed how to use our method for determining which compounds have the potential to treat a given disease we investigate using the gene-wise number of decreasing subsequences S_i to determine which genes are exhibiting more reversal in effective

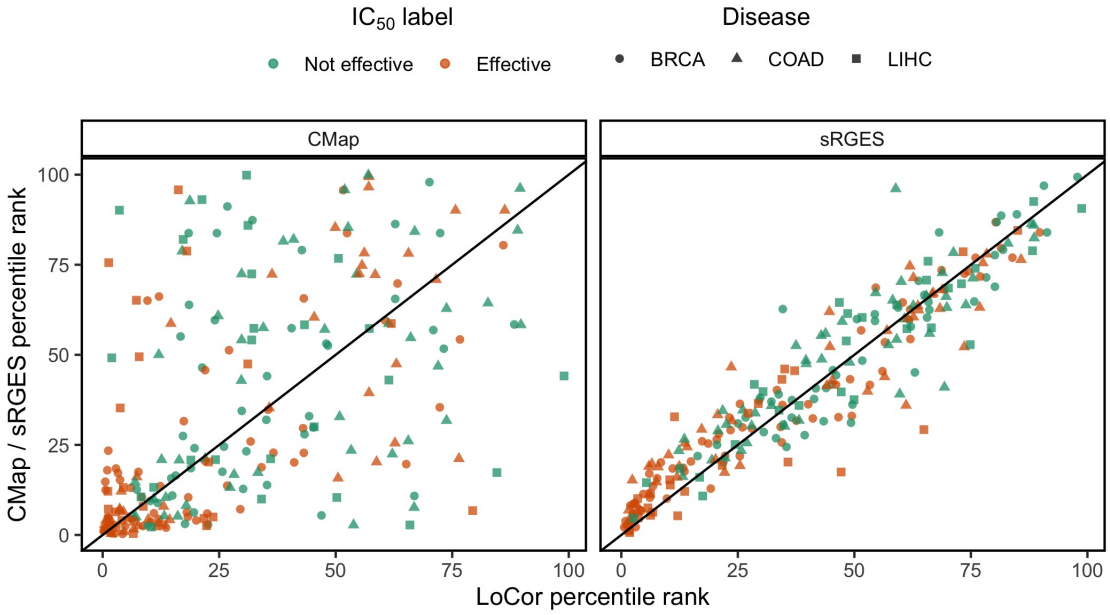


Figure 2.10: Percentile rank of the scores from sRGES (left) and CMap (right) versus LoCor where the color of points indicate where a compound is effective based on the corresponding median that have IC_{50} value. The three different cancers are indicated by the shapes. The diagonal black line is at $y = x$. Any point above the line implies that the compounds is ranked higher, i.e. more effective, by LoCor than one of the other two methods.

compounds, defined as a compound with $\log_{10}(IC_{50}) < 4$ and $S^{LoCor} > 2$, than ineffective compounds, $\log_{10}(IC_{50}) > 4$ and $S^{LoCor} < 1$. We use the resulting conservative p-values from each of these statistics to find pairs of genes and drugs with stronger reversal in the effective group than the ineffective group. Finally, we take the weighted average over the effective drugs, weighted by the S^{LoCor} for each drug, to get a single score for each gene. The details for this computations are included below:

1. Subset compounds into an effective and ineffective groups based on $\log_{10}(IC_{50})$ and LoCor. The effective group has $\log_{10}(IC_{50}) < 4$ and LoCor > 2. The ineffective group has $\log_{10}(IC_{50}) > 4$ and LoCor < 1.

2. Compute conservative p-values based on the gene contribution to the overall LoCor scores for each gene in each sample. Take the $-\log$ of the conservative p-values.

3. For each drug in the effective group, compute mean difference between effective samples and ineffective samples for each gene and divide by the pooled standard deviation of all samples of the effective drug and all ineffective samples.

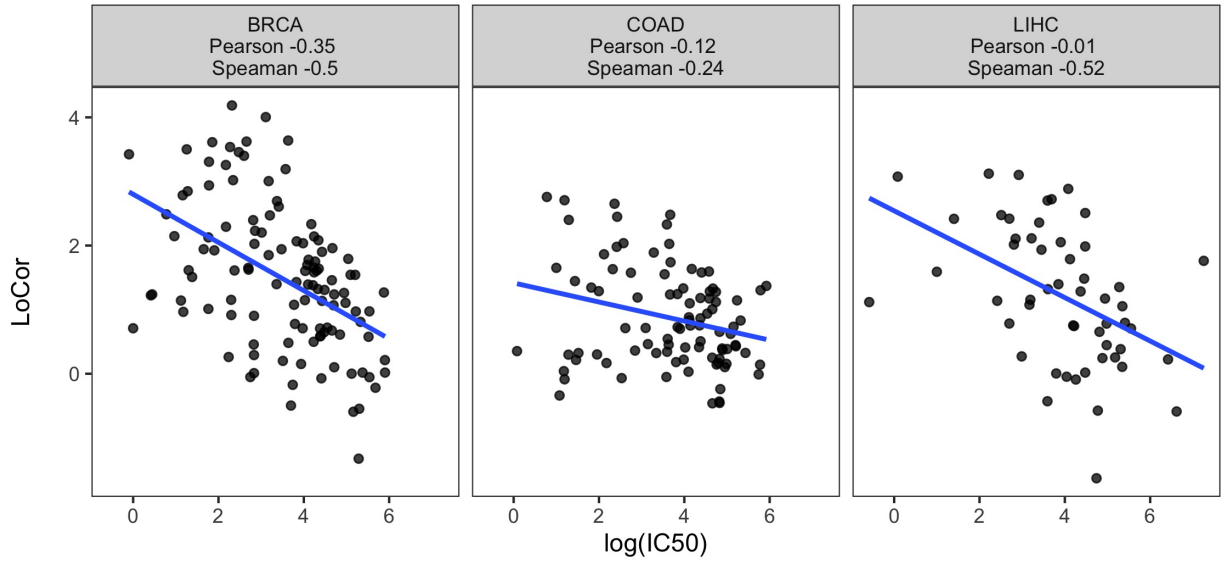


Figure 2.11: Comparison of sLoCor (y-axis) to IC_{50} (x-axis). Notice, the relatively small number of compounds for computing correlation with IC_{50}. This points to the need for continued validation of these methods as new data become available.

4. For each of the effective drugs we now have a score per gene. Take the average, weighted by the LoCor score for the drugs, to get a single score for each gene in a given disease.

To validate these results we looked at the Project Achilles database from the Cancer Dependency Map[96]. Project Achilles quantifies gene essentiaility for cancers by performing genome-scale RNAi and CRISPR-Cas9 genetic perturbations to cancer cell lines to determine which genes are most essential to cancer cell survival. We compared the LoCor and sRGES gene scores to the Achilles effect scores to determine if LoCor is able to effectively discover genes that are essential to survival of the disease in question. A large Achilles score indicates that a gene is important for the disease.

In order to compare the gene scores we do a t-test for difference in means of Achilles effect scores after grouping the LoCor and sRGES gene scores into low and high groups. We choose to do a t-test, instead of correlation, for this comparison because the distribution of the sRGES scores is almost discrete, with almost all gene statistics either close to 0 (essential gene) or close to 1 (non-essential gene). Additionally, the sRGES genes scores are only available for a subset of the genes, 83 in BRCA, 65 in COAD, and 73 in LIHC, thus although LoCor produces a gene statistic for all genes we will only consider the genes with sRGES scores.

The results are in Figure 2.13 and the scatter plot comparing the LoCor gene scores to the Achilles effect score is in Figure 2.14. In Figure 2.13 we can see that for all diseases the genes we find to be important have a significantly higher mean Achilles effect than genes

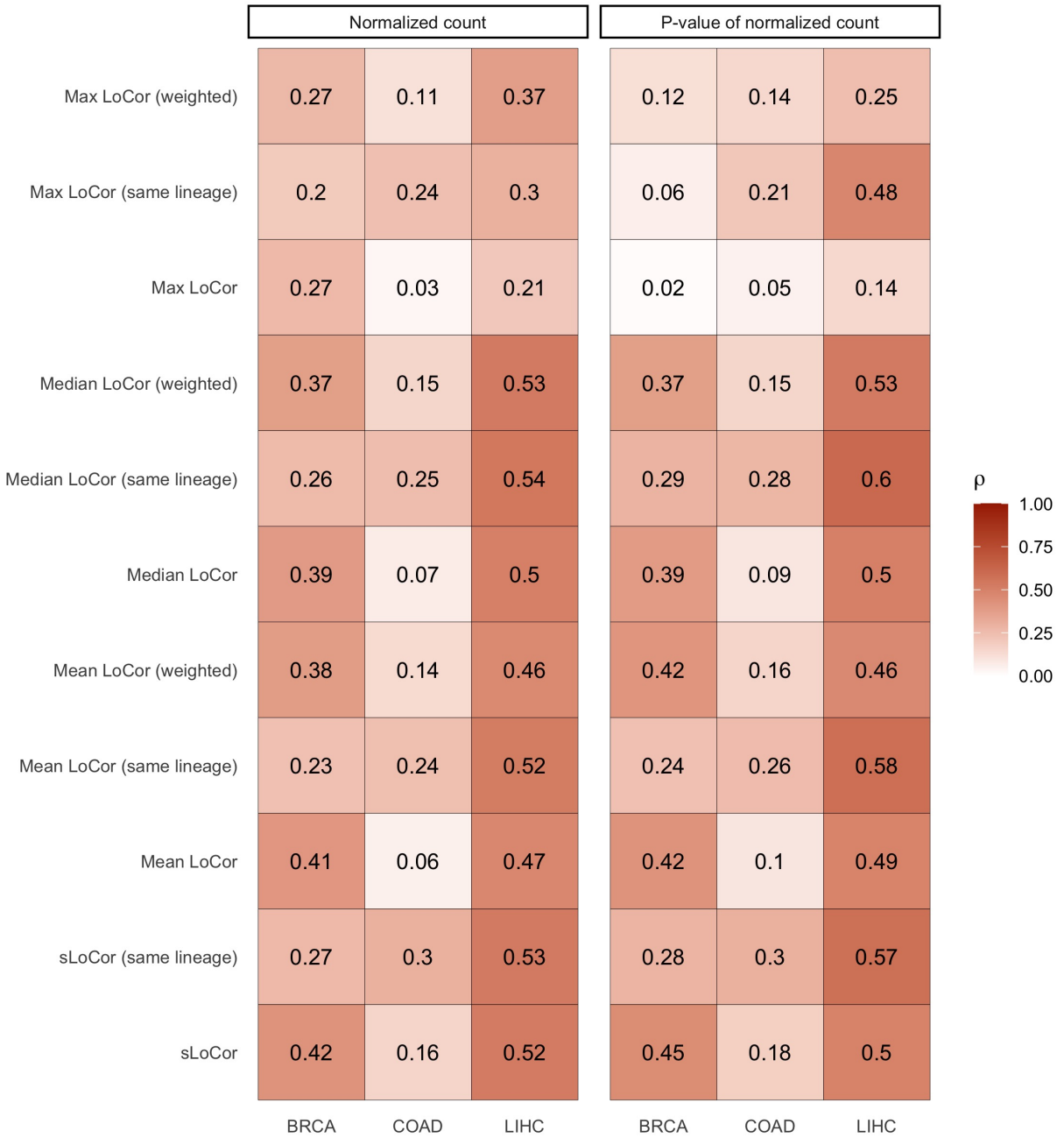


Figure 2.12: Spearman's rank correlation with IC_{50} summarized LoCor scores with different summarization methods. The left panel shows the summarization on the normalized LoCor values, while the right panel is the summarization over the p-value of those normalized statistics.

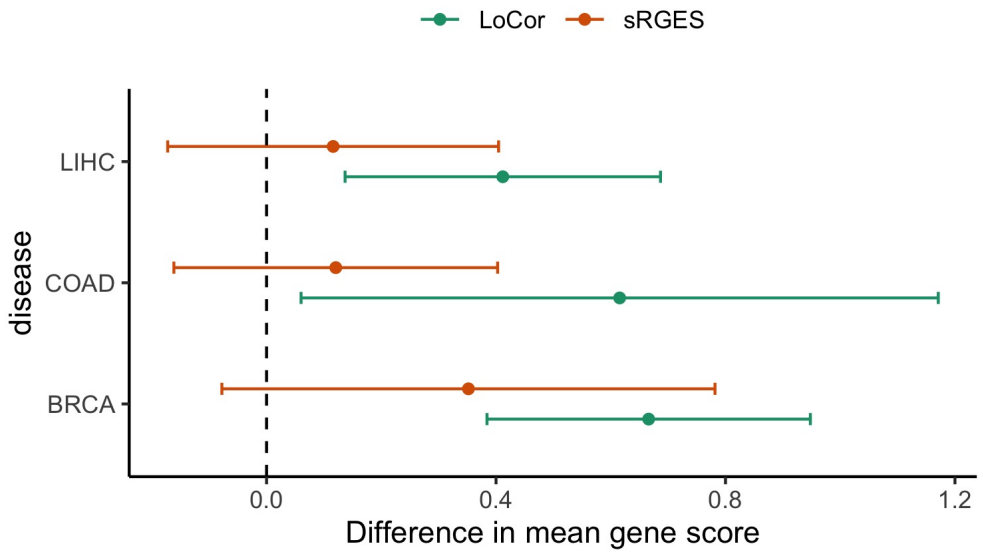


Figure 2.13: Results from a t-test for difference in means of Achilles gene scores when binning the LoCor and sRGES scores.

which we find to be be non-essential. On the other hand, while the sRGES genes score have somewhat higher means, they are not significantly different than zero. Again, there is a strong overlap in confidence intervals indicating that more data is needed to determine whether this difference is real.

Additionally, the scatter plots in Figure 2.14 show while there is some relationship with large Achilles scores corresponding to genes deemed important through the LoCor analyses there are a number of genes with high Achilles importance that are not picked up by LoCor and vice versa. This provides further evidence that additional datasets or methods are warranted to differentiate and validate the gene scores from both LoCor and sRGES.

2.6 Run time

We compared the run time for computing RGES with p-values and LoCor for subsequences of length $k = 2$ and a window size of $l = 150$ with p-values using the package `LoCor` available on GitHub. Note the code for RGES was taken from the corresponding GitHub. CMap code is only available in an online tool where it is possible to control the number of samples, so we were not able to directly compare timing with that method. However, given that CMap and sRGES have similar computational strategies it is likely the computational time between those methods is comparable. Recall, to compute p-value for sRGES it requires a permutation test, the sRGES code defaults to using 10,000 permutations, which is what

we use for this comparison. The results using the `microbenchmark` package in R in the table below. The times were compared using the LIHC data to create a disease signature and subsets of varying size of the LINCS data to see how run time changes as sample size increases. The LoCor algorithm can be simplified when the user does not ask to save gene specific counts, so we show the computational time for LoCor when the user wants to understand the gene contributions and one where the user does not need that information.

	100	1,000	10,000	66,510
sRGES	28.95	31.21	65.23	333.60
LorCor (with gene counts)	6.33	9.03	36.53	247.34
LorCor (without gene counts)	6.20	7.22	16.52	76.98

Table 2.2: Comparision of run for $k = 2$ and $l = 150$ on the LIHC data.

Of note in Table 2.2 we notice that there is significant decrease in run time for LoCor when compared to RGES. This is primarily due to the fact that in RGES (and CMap) p-values for individual samples are computed using a permutation test, however we know the asymptotic distribution of our statistic under the null and have seen in simulation that for n as small as 500 that we approach the asymptotic distribution. Thus, our method has almost automatic computation of p-values that does not rely on a permutation test. Both CMap and sRGES are cloud based computation tools due to issues with run time, but a distinct advantage of LoCor is the ability to compute statistics and associated p-values on a personal computer in the matter of mintues, even for a large number of samples.

We also compute run times for $k = 2$ and various window lengths, which is included in

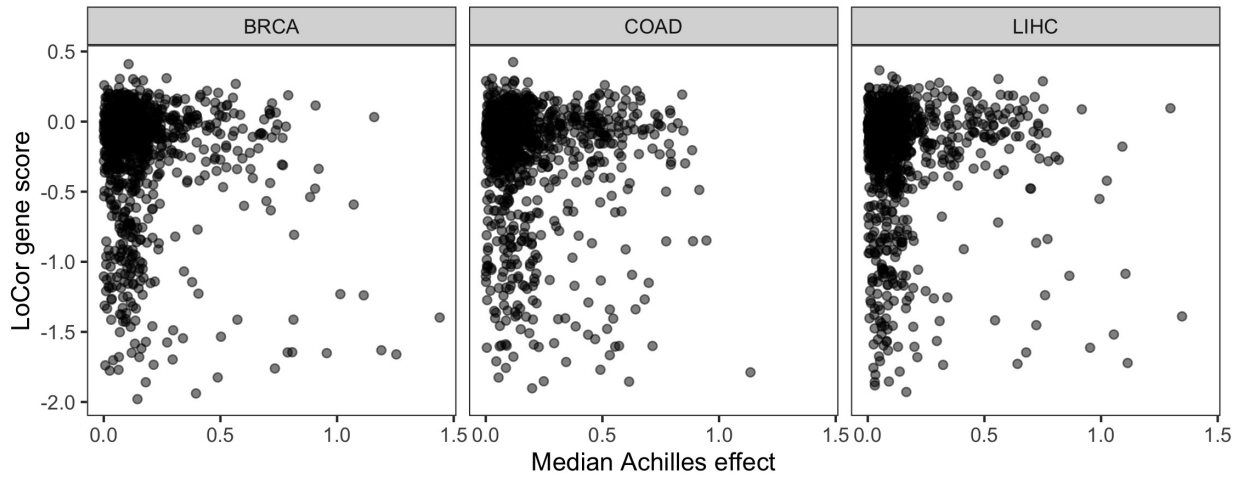


Figure 2.14: Comparison of LoCor gene score (x-axis) to median Achilles effect score (y-axis).

Tables 2.3 and 2.4. It should be noted that for $k > 2$ we see a significant increase in run times, because the algorithm required to compute decreasing counts for subsequences that are larger than 2 must be implemented as a dynamic program which is much more intensive the relatively simple computations when we only care about finding pairs of decreasing subsequences.

Window length l	100	1,000	10,000	100,000
100	4.50	6.42	12.85	86.60
150	6.27	7.62	19.52	128.40
200	8.76	9.54	23.91	170.77

Table 2.3: Comparison of run time in simulated data for varying window lengths l, when we do not need to compute gene-wise statistics

Window length l	100	1,000	10,000	100,000
100	4.65	6.85	29.35	242.32
150	6.58	9.05	37.37	318.10
200	8.71	12.42	50.10	412.19

Table 2.4: Comparison of run time in simulated data for varying window lengths l, when we do need to compute gene-wise statistics

2.7 Discussion

In this paper we introduced a new statistic, called LoCor, to assess the relationship between diseases and drugs based on changes in gene expression. LoCor is a computationally efficient method that overcomes the statistical disadvantages of current methods by removing the need to apply a cutoff to the disease differential expression profile. The statistic is still able to detect the local pattern of reversal in the disease genes by naturally applying more weight to reversal at these genes through the counting decreasing pair in circular windows on the permuted drug DE profile. From our simulation we found that LoCor outperformed previous methods in a number of scenarios, while also being less sensitive to tuning parameters. Using IC_{50} as the pseudo-ground truth to evaluate performance LoCor performs comparably to both CMap and sRGES in real data. Admittedly, IC_{50} is not a perfect evaluation framework given that it is not a definitive indicator of true drug efficacy. Even if we ignore this concern about IC_{50} the limited overlap of drugs with IC_{50} and LINCS L1000 data further limits this validation strategy. We are only able to compare methods on a subset of less than 1% of compounds, and those tend to be the well studied compounds. Ideally, in the future we

would be able to effectively validate a wider range of perturbagens and test discovered drug indications *in vitro*.

Another avenue for future work is to improve summarization across samples of the same compound, particularly as better validation strategies become available. Finally, and potentially most importantly, as data measuring other differential changes of molecular features in diseases and with exposure to drugs, increases in the coming years methods can be applied to these new data modalities will be crucial. The drug discovery statistics need to be flexible enough to incorporate new data types and combine information across these data types. That is one advantage that LoCor has over previous methods, because sRGES and CMap rely on picking a informed cutoff to determine disease related features that will require researchers to appropriately choose such a cutoff in other data types. LoCor, on the other hand, has window length as a tuning parameters, which is less sensitive, and thus more likely to be directly used on these new molecular features. Finally, given the asymptotic distribution of LoCor is independent of this choice of window length, it may be possible to combine information across data modalities by simply taking the mean or median of the individual modality scores.

Chapter 3

GeneFishing in scRNA-sequencing data and its applications

3.1 Introduction

Data which profile the transcriptome at cell-level resolution have been a revolutionary addition to the field of genomics. The so-called single cell RNA-sequencing (scRNA-seq) data have allowed researchers to uncover dynamics that were otherwise impossible to see in bulk transcriptomic data. It is now feasible to create atlases of cells and their sub-types, such as the Human Cell Atlas [78], and to investigate cellular differentiation, among many other applications. Accompanying the opportunities that arise with scRNA-seq data there are unique difficulties, such as high sparsity due to dropout [48] and the need to account for cell-to-cell heterogeneity [100]. The particular characters of this data often require modifying and/or creating new computational tools from those used previously in bulk RNA-seq [114].

While much effort in the past decade has been made to develop statistical methods for tasks in single cell data, such as differential expression analysis, imputation, and cell type clustering, less is understood about quantifying relationships between genes in scRNA-seq. In bulk data using gene-gene co-expression and other modeling techniques to infer gene regulatory networks (GRNs) have been successful [92, 116]. Some of these methods originally developed for assessing gene similarity in bulk data have been applied to single cell data [50, 43].

Additionally, there are recent attempts to develop models and co-expression measures designed specifically for single cell data [91, 20, 66, 97]. Many of these methods require pseudo-time estimates to infer the networks, and thus are only applicable to single cell datasets where such an ordering is possible. Moreover, efforts to benchmark similarity measures and the inferred networks in single cell data have generated conflicting results, with no method consistently producing better inference or outperforming methods originally designed for bulk data [24, 89, 75].

In this paper we extend the semi-supervised learning method called GeneFishing [55],

to assess similarity among genes in single cell data. GeneFishing is a gene prioritization method, meaning that it searches for genes likely to be involved in a biological process based on a small set of genes that are already known to play a role process in question, which we call "bait" genes. The output of GeneFishing is a list of genes that tightly co-express with the genes which are known to be involved in the biological process, thus providing biologists with a manageable group of genes to further study.

To the best of our knowledge this is the first gene prioritzation approach that is applicable to single cell data. Common techniques such as ENDEAVOUR [95] and GIANT [109] are run through online databases which currently have no single cell data for analysis.

While originally designed for use in bulk RNA-sequence data, GeneFishing leverages the statistical techniques of subampling, dimension reduction, clustering, and aggregation to tease out signal in large genomic datasets. In each step of the GeneFishing method small groups of non-bait genes are randomly sampled and subsequently gene-gene similarity is computed between those random sets of genes and the bait. This dramatically reduces the size of the co-expression matrix, and when combined with the other features mentioned above it allows the true biological signal to stand out. Without this subsampling, even in bulk data where gene-gene co-expression is comparatively easier to measure, the large number of gene pairs can lead to high false positive rates. This problem is exacerbated in single cell data due to the higher levels of noise, and has been shown to worsen with the use of imputation [7].

Despite the advantageous properties of GeneFishing, we found two main difficulties with applying GeneFishing directly to single cell data due to the higher noise levels. Namely, it was not trivial to define the biologically relevant bait genes and the application of traditional spectral clustering was often overwhelmed by outliers. In order for GeneFishing to function correctly, the method requires a group of genes with similar expression that are dissimilar to most other genes in the data. The original GeneFishing process used spectral clustering [69] to determine gene groupings. However, we found that in single cell data spectral clustering was not sufficient. We use uniform manifold projection (UMAP) [63] as a second dimension reduction step after computing the spectral decomposition of the gene-gene similarity matrix. The UMAP coordinates result in the desired clustering pattern when the spectral coordinates alone did not.

Additionally, we provide a bait gene selection algorithm that overcomes the difficulties with finding the appropriate group of tightly co-expressing genes. The algorithm takes a larger set of potential bait, usually a group of genes annotated to a pathway or gene ontology (GO) term, and finds subsets that have high co-expression. This allows biologists to begin with a loose hypothesis about related genes that will are expected to be expressed in a sample and still fish out biologically relevant genes.

Initially, the goal of this project was apply GeneFishing to single cell data, through the extensions described above. However, we found that we could also use the methodology as a means to better understand how to to measure of gene-gene similarity in a specific scRNA-seq dataset and as a context-specific feature selection technique for downstream analyses such as clustering cells.

The central part of the automatic bait selection algorithm is a statistic quantifying the tightness of bait set. We found that this can be used to check which similarity measure is most applicable to a given dataset by seeing if genes, which we believe should be co-expressed in a sample are exhibiting that co-expression in the data. Additionally, with the same tightness metric we noticed that cellular heterogeneity is important for detecting co-expression among related genes. We find that when measuring similarity among genes using a homogeneous sample of cells, say all cells with the same cell-type label, the relationships between related genes is not distinguishable, despite being so when additional cell types are included in the analysis. These two observations provide additional insights to the growing literature on GRNs in single cell data.

Finally, we can use the list of biologically relevant genes that are outputted from GeneFishing as a context-specific feature selection method. Many downstream analysis of scRNA-seq require an initial feature selection step, selecting a smaller set of genes, to overcome the curse of dimensionality. There are a number of methods for performing this feature selection, such as using highly variable genes [18] or removing genes with higher than expected dropout-rates [8]. However, these methods choose genes based on expression in all cells in the sample and do not leverage any prior information about genes we know that should be expressed in a particular cell type. We use the output from GeneFishing to select a reasonable size subset of genes which are highly expressed in the targeted cell type. Then, we show that these genes allow us isolate these individual cell types and uncover additional heterogeneity that is challenging to see otherwise.

GeneFishing has been shown to work well in bulk RNA-sequencing data, but our goal in this paper is to improve the method for single cell RNA-sequencing (scRNA-seq) data and present an accompanying R package called `scGeneFishing`. We provide two modifications to GeneFishing that improve performance in single cell data and demonstrate the unique ways that the method can be used to understand the intricacies of single cell data, from cell type heterogeneity to cell type clustering.

3.2 Methods

GeneFishing framework

As mentioned in the introduction, GeneFishing is a semi-supervised technique that relies on clustering and random sampling of genes. The input to the algorithm is a gene expression matrix and a tightly co-expressed and biologically relevant group of genes, which we call "bait". The bait is then used to "fish out" genes that are likely to be related to the original biological process. The authors found that the method worked well in bulk RNA-sequencing data. Namely, using the liver data from GTEx [57] they were able to discover novel cholesterol metabolism genes, which were experimentally validated.

An overview of the framework is included in right panel of Figure 3.1. Given a tightly co-expressed set of bait genes the method works by randomly partitioning all other genes

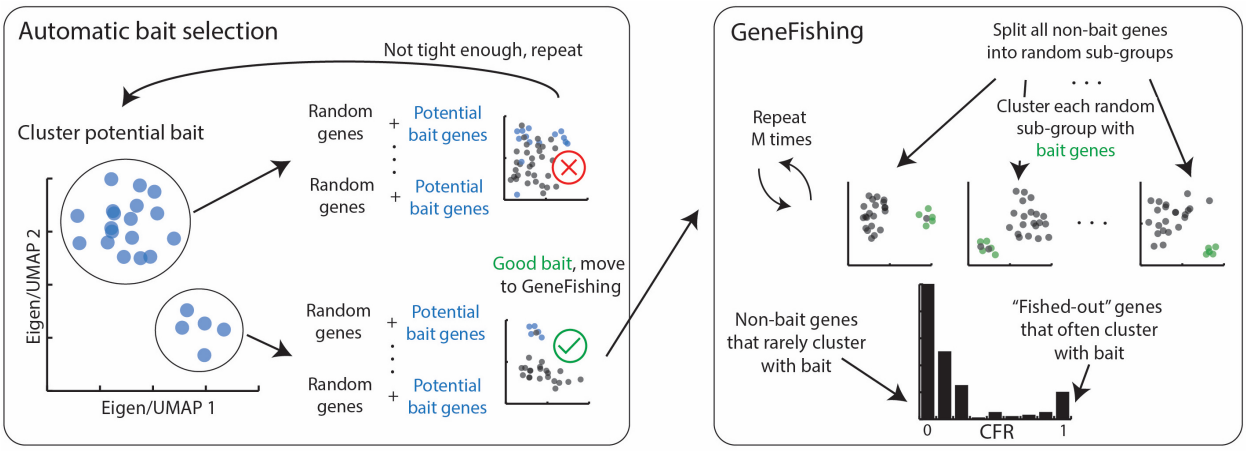


Figure 3.1: Schematic of automatic bait selection and GeneFishing procedure. Starting with a larger set of potential bait genes (e.g. genes from a GO term) we search for subsets of those potential bait that are able to be used as bait. If there is a subset with tight co-expression we call that bait and do GeneFishing. In GeneFishing, all non-bait genes, are split into random sub-groups that are clustered with the bait genes. This is repeated n times and the proportion of times that a non-bait gene clusters with the majority of the bait over the n samples is called the capture frequency rate (CFR). High CFR indicates the non-bait gene is highly co-expressed with the bait genes.

into sub-search-spaces, which contain α times the number of bait genes. Typically, we set $\alpha = 5$, but this is a parameter of the method that can be changed. These sub-search-spaces are the key of GeneFishing as they increase the signal-to-noise ratio and allow the biological structure in the data to be detected. For each sub-search-space we apply spectral clustering [69] with $k = 2$ clusters to the non-bait and bait genes. The non-bait genes which cluster with the majority of bait genes have similar expression patterns in that sample, while the non-bait genes in the other cluster have less similar expression. For this to work the bait genes need to form a tight cluster in most sub-search-spaces such that only a small number of bait are in the bait gene cluster. This desired clustering structure is depicted by the green bait genes in Figure 3.1.

This random partitioning is repeated N times and we count the proportion of times that each non-bait gene clusters with the bait genes for each partition, which we call the capture frequency rate (CFR). Typically, the CFR distribution looks similar to that in Figure 3.1, where the majority of genes have CFR ≈ 0 indicating they rarely cluster with the bait, and a smaller set of genes have CFR ≈ 1 indicating they almost always cluster with the bait. Those genes with CFR ≈ 1 are what we say are fished out. The steps for this algorithm as summarized below:

Step 1: Randomly partition the data into sub-groups of size m.

Step 2: For each sub-group compute co-expression matrix of m sub-group genes and the

bait genes.

Step 3: For each sub-group co-expression matrix do spectral clustering and determine which, if any, sub-group genes cluster with the majority of the bait genes.

Step 4: Repeat steps 1-3 N times and compute capture frequency rate (CFR), which is the proportion of times that each non-bait genes clusters with the majority of the bait genes. The CFR is used to rank relevance of genes.

Step 5: Determine cutoff on CFR for fished genes using method from [112]

For scRNA-sequencing data we optionally modify **Step 3** by using UMAP on the original spectral coordinates before applying the k-means algorithm to perform clustering. This additional dimension reduction step improved performance by increasing signal in the presence of noisy genes.

Automatic bait selection algorithm

In the original GeneFishing method the user needed to *a priori* select a tightly co-expressing set of bait genes. However, finding such a set is not trivial in bulk data and due to the sparsity of single cell data very challenging in single cell data. We alleviate this problem by adding an automatic bait selection algorithm that will take a larger list of biologically related genes, which we will call potential bait W and search for a set of subsets $\mathcal{B} = \{B_1 \subseteq W, \ldots, B_m \subseteq W : B_i \cap B_j = \emptyset,\ \forall i \neq j\}$ that cluster tightly in the presence of randomly sampled genes. Each of the subsets $B_k = (b_1, \ldots, b_{m_k})$ can then be used as bait. Typically, W will be all genes annotated in a GO term or pathway that is thought to be expressed in the sample of cells present in the data.

If there are no subsets of W that can be used as bait that is likely either due to the potential bait being unrelated to the sample of cells in the data or the need to use a different measure of similarity. In this sense the automatic bait selection algorithm is a good way to check the appropriate measure of similarity in a given dataset. We illustrate this idea in Section 3.2 below. For this paper we find that using Spearman's rank correlation is sufficient, but the R package allows for the use of different similarity measures such as Pearson's correlation and cosine similarity.

The tightness of the set of bait B_k in the presence of a set of the $m_k \times \alpha$ randomly sampled gene set $R_l = (r_1, \ldots, r_{m_k \times \alpha})$ is determined by the following metric:

$$t_l(B_k) = \begin{cases} \frac{\text{median}(||b_i - b_j||)}{||m_B - m_{R_l}||} & \text{spectral coordinates} \\ \frac{\text{median}(||b_i - b_j||)}{\sqrt{||m_B - m_{R_l}||}} & \text{UMAP coordinates} \end{cases}$$

for all $i \neq j : b_i, b_j \in B_k$. We do this for $l = 1, \ldots . M$ samples to get the average tightness

$$t(B_k) = \frac{1}{M} \sum_{l=1}^{M} t_l(B_k)$$

With $\alpha = 5$, we are sampling five times the number of genes in B_k. Intuitively, this metric compares the average distance between all bait genes to the distance between most bait and most random genes. We use medians and medoids to avoid strong influence from outliers. The final step in the algorithm is to ensure that for each discovered bait set B_k for $k = 1, \ldots, m$ does not cluster with too many random genes for most samples. This is necessary to avoid large influence from lowly expressed genes, which often form a small cluster of their own.

If we are using UMAP, instead of spectral coordinates, to probe for bait we will slightly modify the tightness metric to mitigate the fact that separation between clusters is exaggerated in the UMAP space. We do so by taking the square root of the distance between m_B and m_{R_l}.

The algorithm for selecting bait, which is summarized in Algorithm 1 works by first computing the tightness of the potential bait set W, $t(W)$. If $t(W)$ is below the tightness threshold it will be used as bait, otherwise we split W into two groups. Splitting the genes is done with by applying k-means with two clusters, either on the spectral coordinates of the gene-gene correlation matrix or the UMAP coordinates of those spectral coordinates. We then check the tightness of the two clusters, and if it is below the threshold we say those are a set of bait genes. We typically use 0.5 as a threshold on the tightness of a bait set. This roughly corresponds to the distance between the bait and random genes being twice as big as the distance between bait genes. If the set of genes are not tight enough to classify as bait, we try splitting the cluster again and recomputing tightness. We stop when there are no sets with less than a minimum number of genes, usually 10.

Algorithm 1 Automatic bait selection

$\mathcal{P} \leftarrow \{W\}$
while $|\mathcal{P}| > 0$ **do**
  $\mathcal{P}_{tmp} \leftarrow$ []
  for P in $\mathcal{P}$ **do**
    Cluster $P = (p_1, \ldots, p_m)$ into two clusters
    for $k \leftarrow 1$ to 2 **do**
      if $t(\{p_i : i \in k\}) \leq$ `cutoff` **then** ▷ check if set if tight enough
        $cnt1 \leftarrow cnt1 + 1$; $B_{cnt1} \leftarrow \{p_i : i \in k\}$ ▷ these genes are a bait set
      else if $|\{p_i : i \in k\}| >$ `min genes` **then**
        $cnt2 \leftarrow cnt2 + 1$; $\mathcal{P}_{tmp}[[cnt2]] \leftarrow \{p_i : i \in k\}$ ▷ these genes will be split and tested again
      end if
    end for
  end for
end while

3.3 Results

Data Processing

For all real data analyses we used data from the Tabula Muris Consortium [28]. We followed the pre-processing steps outline in Chapter 6 of "Orchestrating Single Cell Analysis" [5]. We used `quickPerCellQC` from the `scater` R package [61] to remove any cells with a high proportion of spike-in reads.

After removing low quality cells we followed Chapter 7.5.1 of "Orchestrating Single Cell Analysis" to normalize the expression matrix. We again used the `scater` package to normalize the counts using the default `quickCluster` and `computeSumFactors` to normalize by applying the deconvolution strategy [59]. Finally, we log-transformed the data. It is important for single cell data to use the normalized and log-transformed counts as input to the automatic bait selection and GeneFishing algorithms in order to appropriately assess similarity of genes.

In addition to filtering out low quality cells we removed genes with low expression. In order to reduce the search space for GeneFishing we removed 25% of genes with the lowest variance across the remaining cells. If any genes remained that were not expressed in at least 5 cells those were removed as well.

Automatic bait selection in liver data

Bulk RNA-seq GTEx data

As an initial test of viability of the described automatic clustering algorithm we looked at the bulk RNA-seq liver data from GTEx [57] used the original GeneFishing paper. In the original paper the authors used with 21 genes from the GO term "GO:0008203 cholesterol metabolic process" from Ensembl BioMart, which were hand selected from all genes annotated in the GO term. Since the publishing of the paper additional genes have been added to the GO term. Using the automatic bait selection algorithm we recovered a bait set with 29 genes, including all 21 of the original bait set genes, 20 of which are in the cholesterol biosynthesis pathway.

In Figure 3.2 (a) we can see the spectral coordinates for all GO term genes. The newly discovered bait set contains 8 genes that were not in the previous bait set (the green circles) while the blue points represent annotated GO term genes that will not be used as bait. Recall, that the bait needs to cluster tightly in the presence of non-bait genes and the selected bait satisfies that constraint while the unselected GO term genes do not as is shown in Figure 3.3. Overall, this algorithm produces a bait set that is very similar to the bait set manually selected in the original paper as Algorithm 1 is appropriately subsetting GO term genes based on their co-expression.

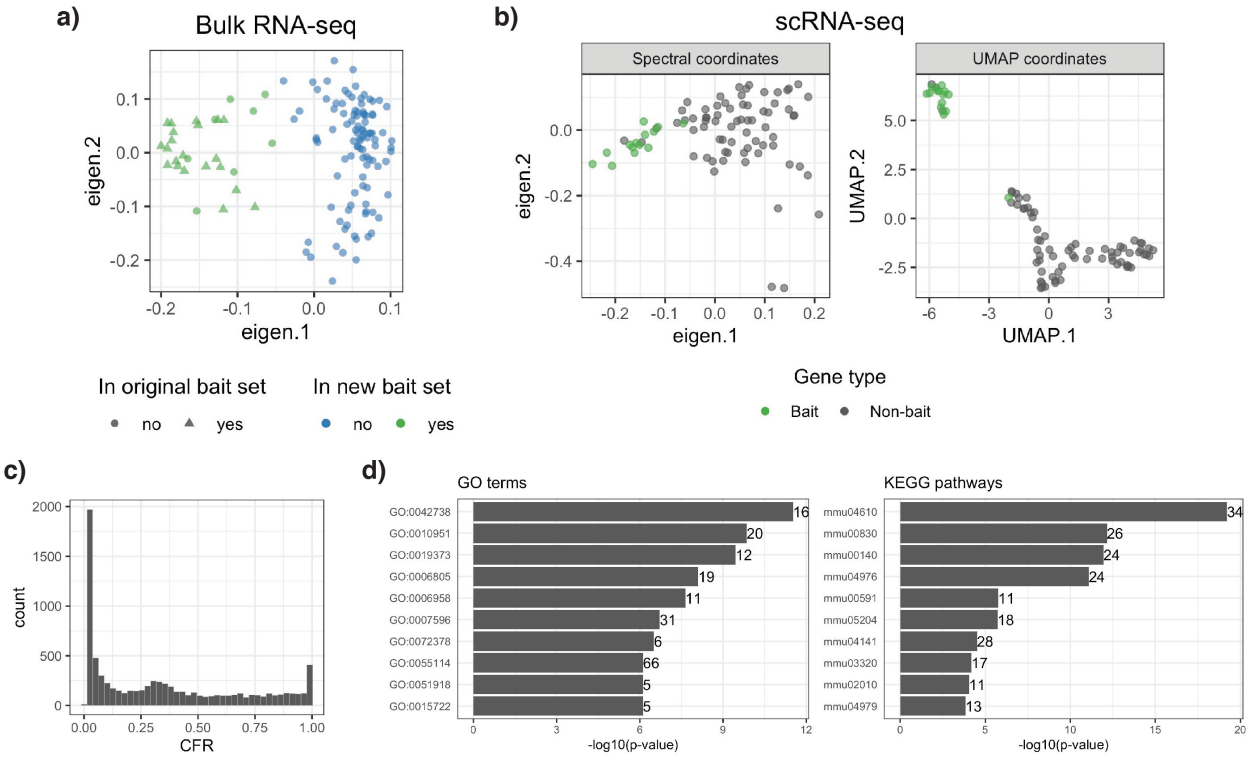


Figure 3.2: Results from automatic bait selection and GeneFishing in liver data. **a**) Comparing selected bait GTEx bulk RNA-sequence data to the bait used in the original GeneFishing paper. All points on this plot are from the cholesterol metabolic process GO. We discover a set of 29 genes (green points) to be used as bait and recover all original bait (triangles). **b**) Comparing spectral and UMAP coordinates for bait selection in mouse scRNA-sequence data. Green points are the 14 genes from cholesterol metabolic process GO term that were discovered by the algorithm and red points 70 genes randomly sampled from all other genes in the data. UMAP coordinates clearly cluster bait genes which is not the case with spectral coordinates. **c**) Capture Frequency Rate (CFR) from GeneFishing with UMAP bait from **b**). **d**) GO term (left) and KEGG pathway (right) enrichment results for the fished out genes from **c**). To the left of the bar are the number of genes in the fished out gene set (excluding the original bait) that are annotated in the GO term or pathway.

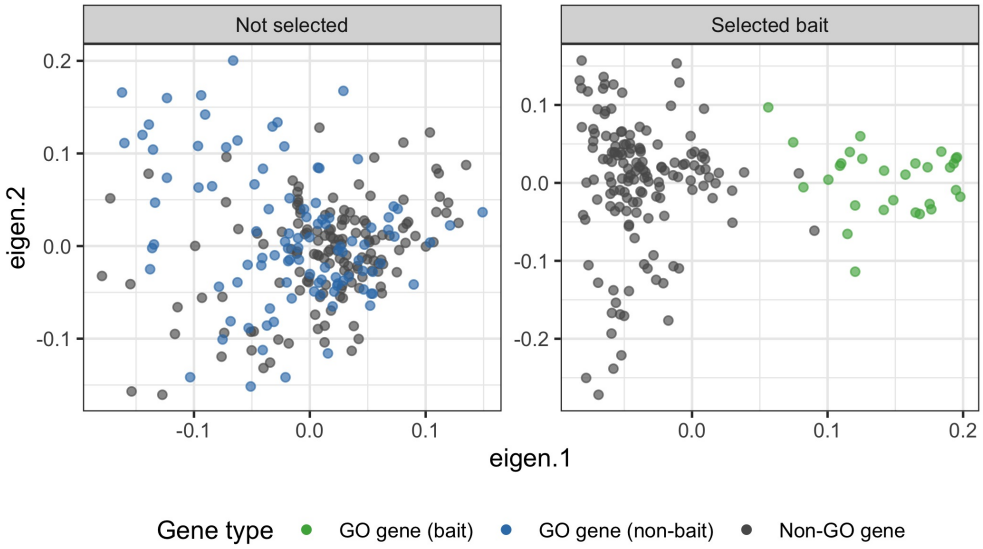


Figure 3.3: Automatically selected bait (right) vs. non-selected cholesterol metabolic process GO term annotated genes in the presence of genes that are not annotated in the cholesterol metabolic process GO-term (grey). We see that the selected bait (green) separate along the first eigen vector from the non-GO term annotated genes, which is not the case for any reasonable sized subset of the group of GO-term annotated genes that were not selected (blue).

Using UMAP to select bait and perform GeneFishing in liver scRNA-seq data

In addition to performing Algorithm 1 on the bulk data we used the same GO term for mice in scRNA-seq liver data from the Tabula Muris consortium [28]. The cholesterol metabolic process GO annotated genes come from http://www.informatics.jax.org/go/term/GO:0008203. We used the normalized log counts from 7,878 genes in 1,924 liver cells. This data was collected using microfluidic droplet-based 3'-end counting technology. Of these cells, 1,764 are hepatocyte cells making this a relatively homogeneous sample. To remove unexpressed genes from the search space we did an initial feature selection step of removing the 25% of the genes with the least variable expression.

After removing genes with low expression we were left with 99 annotated genes, which we used to search for bait sets with Algorithm1. Using spectral coordinates we were unable to recover any subsets of the cholesterol metabolic process genes that would be used as bait. However, by using the UMAP coordinates of those spectral coordinates in **Step 3** we found two sets of 14 and 11 genes, respectively, that are tightly co-expressed. The 14 bait genes selected have three cholesterol metabolic process pathway genes while the seven of the eleven genes for the second bait gene are in the steroid biosynthesis pathway.

The spectral and UMAP coordinates for the 14 gene bait set (green points), which is the

tighter of the two, in the presence of 70 randomly sampled genes (grey points) are shown in Figure 3.2 (b). We can see that the UMAP coordinates produce a bait gene cluster that is clearly separated from most random genes. On the other hand in the left panel of the spectral coordinates do not exhibit such separation. This is mostly due the large influence on the spectral decomposition from the small number of randomly sampled genes that in the bottom right hand of the figure. Although we are only showing one sample of non-bait genes this phenomenon is consistent across the majority of samples we saw for this liver data.

We have found that there are times where both spectral and UMAP coordinates can be used to discover similar, or exactly the same, bait sets. This is particularly true when there is a tightly co-expressing set of bait genes that separates from non-bait genes in the spectral coordinate space. If that is the case using UMAP coordinates provides no additional benefit. The primary benefit for the purposes of bait discovery is in the situation that is depicted in Figure 3.2 (b) where a small number of non-bait genes consistently separate themselves from all other genes which makes the bait gene cluster indistinguishable from the majority of the non-bait genes.

In addition to using UMAP to in the automatic bait selection algorithm, we can also use UMAP throughout the GeneFishing process, as additional layer of dimension reduction prior to performing k-means. The CFR distribution for doing so with this 14 gene bait set is show in Figure 3.2 (c). We can see that this looks very similar to the idea CFR distribution in that we are fishing out a relatively small group of non-bait genes, in this case 453 genes. We do fish out 31 genes that were in the original GO term, despite there being on 14 genes in the bait set.

In order to assess the quality of the fished out genes we check which KEGG pathways and GO terms those genes are enriched for in comparison to the other genes in the data using the R packages `clusterProfiler` [113] and `topGO` [4], respectively. The 10 most enrichment pathways and GO terms are shown in Figure 3.2 (d) where the number at the edge of each bar indicates the number of fished out genes in either the pathway or GO term. This data is also shown in more detail the Supplementary information Tables B.1 and B.2.

We find that fished out genes are enriched for a number of pathways that are related to cholesterol metabolism such as steroid hormone biosynthesis (mmu00140), bile secretion (mmu04976), linoleic acid metabolism (mmu0591), and cholesterol metabolism (mmu04979). Cholesterol is the precursor for many classes of steriod hormones and is relied on in the biosynthesis pathway [85] and bile is a direct byproduct of cholesterol biosynthesis [27]. Additionally, linoleic acid is known to lower blood cholesterol levels [19]. While we fish out 13 additional cholesterol metabolism pathway genes that were not in the original bait set.

However, the two pathways with the strongest enrichment, the complement and coagulation cascades (mmu04610) and retinol metabolism pathways (mmu00830) have a less clear relationship with the cholesterol metabolic process. There is a limited amount of research connecting blood coagulation with cholesterol levels [12, 64] as well as blood retinol and blood cholesterol levels [99], but the connection is less direct than the other enriched pathways mentioned above.

Additionally, cholesterol, drug and xenobiotic metabolism are all subject to feedback from

P450 enzymes and transporters [81], so the enriched GO terms of exogenous drug catabolic process (GO:0042738), negative regulation of epoxygenase P450 pathway (GO:0019373) and xenobiotic metabolic pathway (GO:0006805) all make sense biologically. Again there is not much research connecting cholesterol metabolism to other enriched GO terms like endopeptidase activity (GO:0010951) or blood coagulation (GO:0007596), but overall the fished out genes are enriched for a number of pathways and GO terms with direct implications in cholesterol metabolism. This indicates that we are the genes clustering with the cholesterol metabolism bait have biological importance.

Overall, we were able to use UMAP coordinates to detect viable bait sets in scRNA-seq data when it was not possible with spectral coordinates. Those bait genes were then used to perform GeneFishing, also with UMAP coordinates, and produce a fished out gene set that contains genes which are present in pathways and GO terms relevant to cholesterol biosynthesis.

In general, we have found that using UMAP will always produce results with smaller sets of fished out genes than spectral coordinates on the same bait set. This is important if the biologist is interested in having a small set of hypotheses to test. However, UMAP does increase the computation time considerably, so it is recommended to use spectral coordinates if you are able to discover a tight enough bait set in that coordinate space prior to using the GeneFishing algorithm with UMAP. The timing comparison for spectral and UMAP is included in Tables 3.1 and 3.2.

Coordinate type	Correlation method	Time (s)
Spectral	Pearson	480.4
	Spearman	604.2
UMAP	Pearson	914.4
	Spearman	1075.8

Table 3.1: Time in seconds for using `probeFishability` function to search for bait sets from potential bait on 20 cores. Using the liver Tabula Muris data with 7,878 genes and 1,924 cells. Our potential bait set is the cholesterol metabolic process GO term genes. There are 97 of these genes in the data. Throughout we set $\alpha = 5$, $k = 2$, and $M = 100$

Coordinate type	Correlation method	Time (hh:mm:ss)
Spectral	Pearson	01:05:36
	Spearman	01:48:25
UMAP	Pearson	05:02:32
	Spearman	05:27:01

Table 3.2: Time in seconds for using `scGeneFishing` function to perform GeneFishing using bait selected from the automatic bait selection algorithm on 20 cores. Using the liver Tabula Muris data with 7,878 genes and 1,924 cells. Our bait set is 14 genes chosen from the cholesterol metabolic process GO term genes. Throughout we set $\alpha = 5$, $k = 2$, and $N = 1000$

Importance of cell heterogeneity and metric choice to gene-gene similarity

In addition to using UMAP coordinates to refine the analysis in single cell data, we can use the automatic bait selection algorithm to to assess the importance of cell heterogeneity and similarity metric choice to co-expression analysis. The viability of GeneFishing is directly related to whether there is signal in gene-gene similarity. In this section we use a sample of 1,419 pancreas cells, again from the Tabula Muris consortium, to that how we can use Algorithm 1 to determine a good similarity metric and to illustrate that using a set of cells that is too homogeneous makes it difficult to separate signal from noise.

Impact of similarity metric

We looked at three measures of similarity, Pearson correlation, Spearman's rank correlation, and cosine similarity. Using these measures we found that Pearson and Spearman's rank correlation outperformed cosine similarity in the pancreas data. After data pre-processing the pancreas scRNA-seq data used from Tabula Muris contained measurements from 10,856 genes in 1,419 cells. This data was collected using FACS technology. There are nine cell types that were labeled in the pancreas including 439 beta cells and 386 alpha cells.

Using the aforementioned measures of similarity we found almost exactly the same bait set of 11 genes from the insulin secretion GO term http://www.informatics.jax.org/go/term/GO:0030073 with the exception that there was one extra gene in the Spearman's correlation bait.

Using the 11 genes common to all three bait sets we computed the tightness of the bait $t_l(B)$ over $M = 500$ random samples of non-bait genes. In Figure 3.4 (a) we plot the tightness of each similarity metric using UMAP coordinates across these samples and see that Pearson and Spearman's correlation consistently has the bait set separating from the non-bait genes. Recall, that a tightness of 0.5 implies that the bait genes are twice as close to each other as they are from the non-bait genes in the sample. Cosine similarity still finds this bait to be tight enough with $mean\,(t_l(B)) < 0.5$, however it is on average much more disperse than

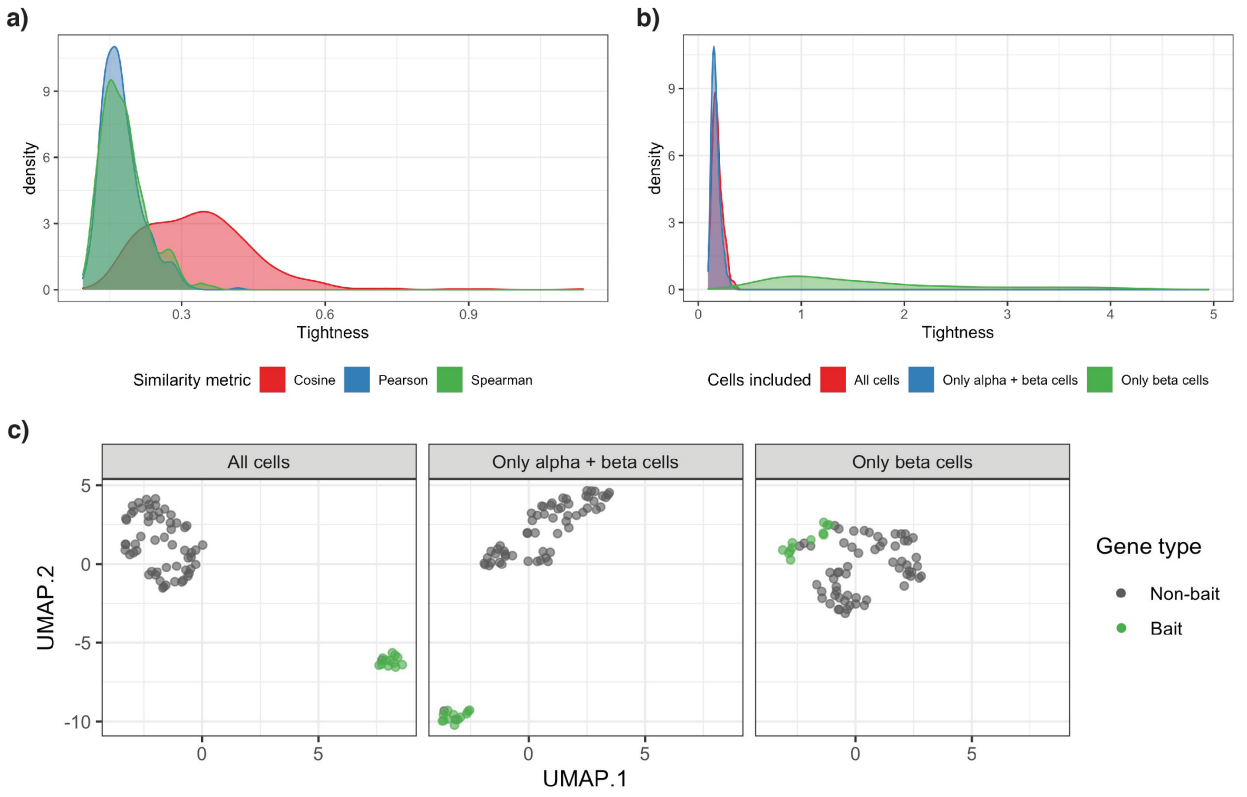


Figure 3.4: Impact of cellular heterogeneity and similarity metric choice of GeneFishing. **a**) Tightness of a set of 11 bait genes (the same set was found by all three similarity metrics) in pancreas cells over 500 random samples of non-bait genes. Pearson and Spearman's correlation on average produce tighter bait sets, which results in more precise GeneFishing results, than cosine similarity in this data. **b**) Tightness of the same 11 bait genes when using different groups of cells to cluster genes. When using all the cells in the data (red) or using only the alpha and beta cells (blue) we are able to produce tight clusters of non-bait. Using only beta cells (green), which are the cells expected to have high expression of insulin secretion genes, we see much less differentiation in co-expression from the non-bait genes. **c**) Example of a sample of non-bait genes in the pancreas using those 11 insulin secretion genes. We can see how tightly the bait cluster away from the non-bait when using all cells or alpha and beta cells. That is not the case when we only consider beta cells.

the the other two metrics. This implies that in this data using one of the two correlation measures is appropriate. In Figure 3.5, we show the UMAP projection of two samples of non-bait genes for each similarity measure. We also a similar pattern using the liver data and bait from Section 3.3 where, again, Pearson and Spearman's correlation work best in Figures 3.6 and 3.7.

In general, the user can use the automatic bait selection algorithm to determine whether there is a similarity metric that allows for a subset of genes in a pathway or GO term to show strong co-expression. If there no similarity metric exhibits co-expression, then it is likely that those genes are not related in the sample of cells.

Impact of cell heterogeneity

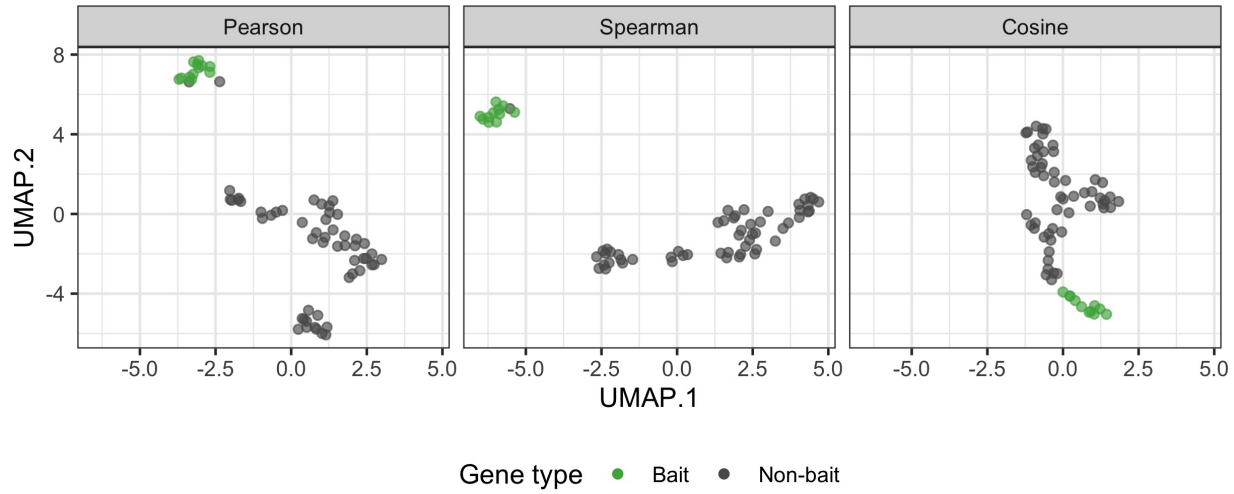
We found that the level of heterogeneity in the cells has an interesting relationship with the ability to perform GeneFishing in this data. When we use cells, all of the same cell type, the gene-expression is too homogeneous to see clear clusters of the bait genes. However, if we include one additional cell type in the analysis, the contrast of expression in those cell types allows for clear patterns to arise. It is often the case that this one additional cell type produces similar results to using all the cells in the data.

We illustrate this point by considering the same pancreas data. Of the 1,419 pancreas cells there are 439 cells that are labeled as pancreatic beta cells and 386 as pancreatic alpha cells, as well as smaller numbers of an additional seven cells types If we consider the insulin secretion GO term we expect the expression of these genes to be strongest in beta cells, because insulin is secreted in response to high blood glucose levels from beta cells in the pancreas.

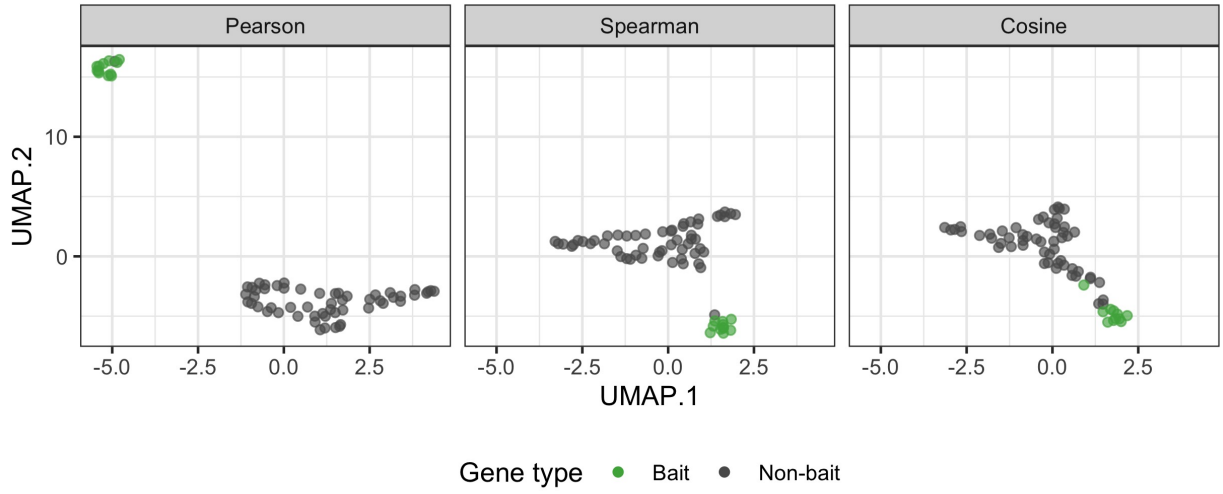
In Figure 3.4 b we see the bait tightness for the 12 bait genes selected from the insulin secretion GO term using Spearman's rank correlation in three different groups of cells. The distribution of tightness values is centered around 0.2 when using the two larger groups of cells. However when we restrict to only use beta cells we see much higher tightness scores, indicating that the bait genes are clustering much closer to the non-bait genes in most samples. In Figure 3.4 c we see an example of this for a single sample of non-bait genes. Clearly, the bait is separating well when using all cells or both alpha and beta. This is somewhat surprising as those insulin secretion genes are more highly expressed in the beta cells.

One possible explanation is that we are showing this for bait selected using all cells, so it may be that we can find viable bait with Algorithm 1 when only using the beta cells. We did this and found a small group of 7 insulin secretion genes, 4 of which were in the original bait set found from all cells. These 7 genes do cluster more strongly in the beta cells only (See Figure 3.8), however the tightness is much lower than when we use all cells and has much higher variability.

Additionally, using these bait to perform GeneFishing we get better results when we use the 12 bait genes in all cells as opposed to the 7 bait genes in beta cells. We are only able to fish out a set of 4 additional genes using only the beta cells, while we get a group of 158



(a) Non-bait sample 1.



(b) Non-bait sample 2.

Figure 3.5: Similarity for three different similarity metrics in two samples of non-bait genes in the pancreas using a set of 11 bait genes selected from the insulin secretion GO-term. In both (a) and (b) we are able to get clear clusters of bait genes using Pearsons and Spearman's rank correlation. Cosine similarity gives good separation in (b) but not (a).

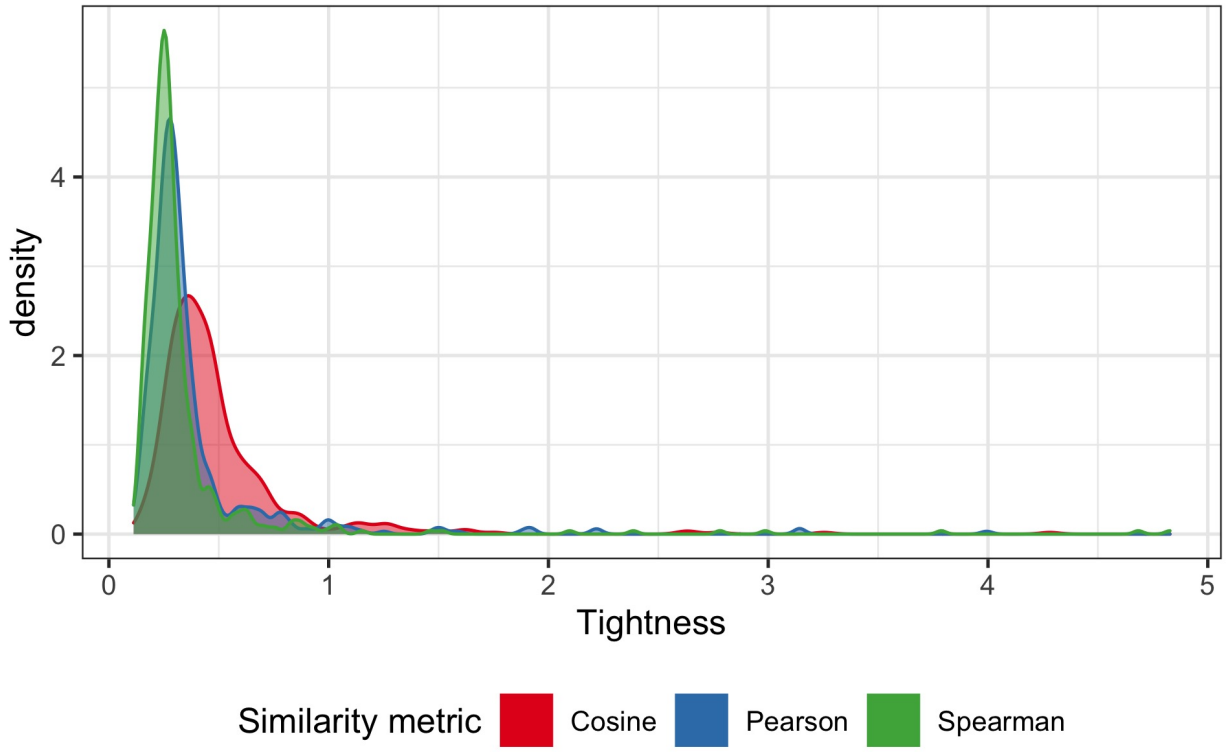
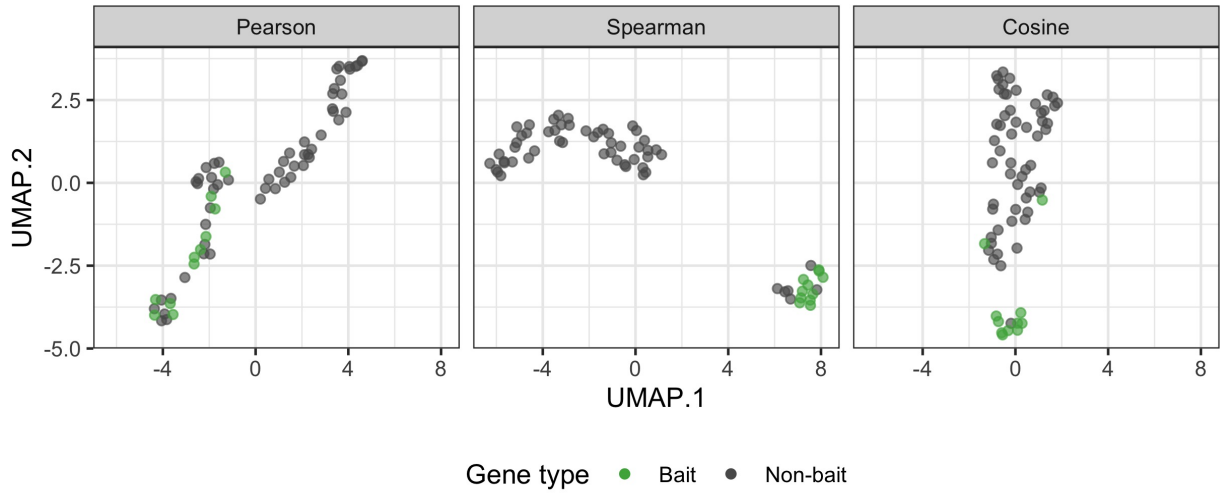


Figure 3.6: Tightness for three different similarity metrics in liver using genes selected from cholesterol metabolic process GO term as bait.

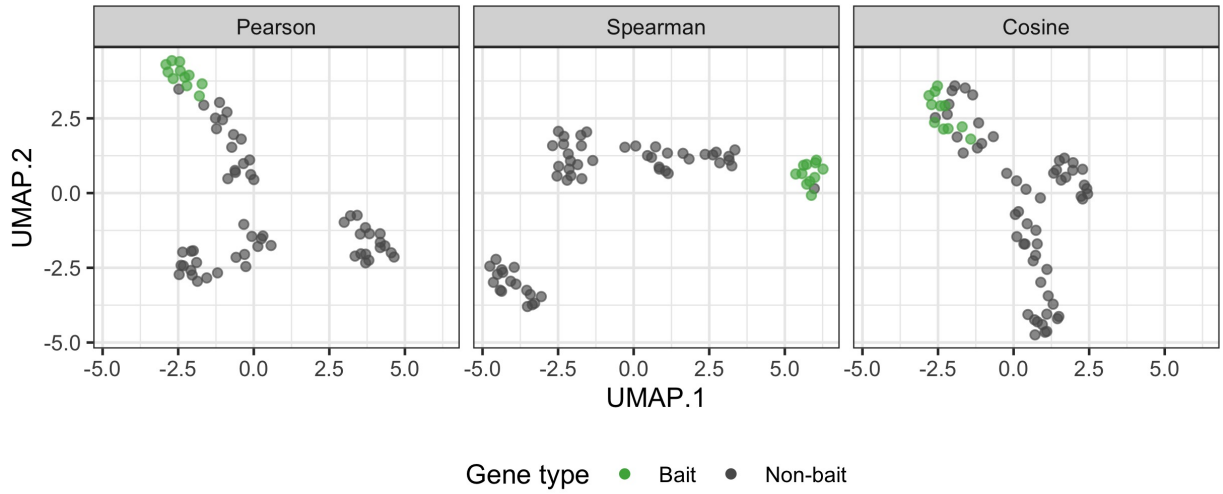
genes when using all cells that are enriched for pathways associated with insulin secretion such as maturity onset diabetes of the young (mmu04950) and prolactin signaling pathway (mmu04917). Diabetes is a disease marked by the inability to properly secrete insulin for controlling blood sugar levels while prolactin increases insulin production [73]. The 158 genes are also enriched for related GO terms such as glucose homeostasis (GO:0042593) and G protein-couple adenosine receptor signaling (GO:0001973), which play a role in regulating insulin prodction [46, 38].

We also checked that the tighter co-expression is not just a result of the number of cells in the analysis by randomly selecting 439 cells from all pancreas cells and re-computing the tightness. We found that we were still able to have very tight clustering of all 11 genes across a number of samples of cells. In Figure 3.9 we show the UMAP projection for three of these random samples. The bait is able to strongly separate from the non-bait genes in all three instances. This implies that the decrease in co-expression when we restrict to only using beta cells is not simply due to there being a low number of cells.

Finally, we see a similar phenomenon using a sample of heart and aorta cells, which is shown in Figure 3.10 where the tightest co-expression arises when we use all cells to cluster

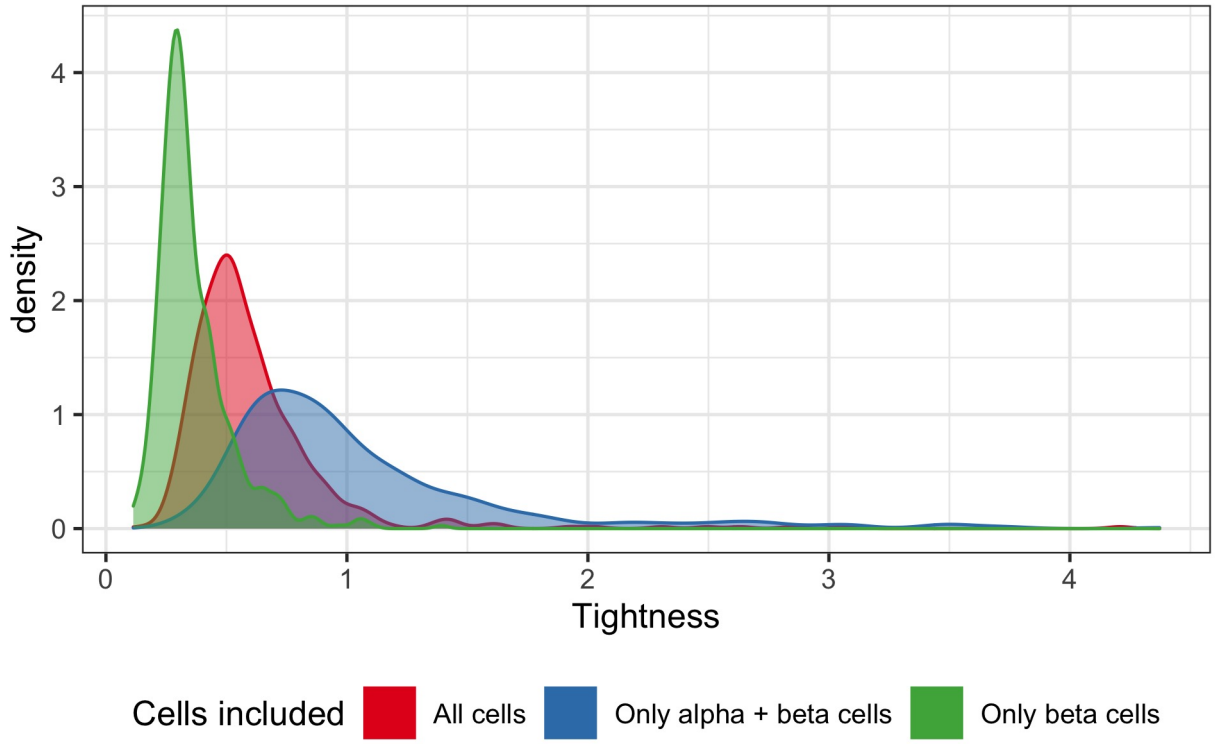


(a) Non-bait sample 1.

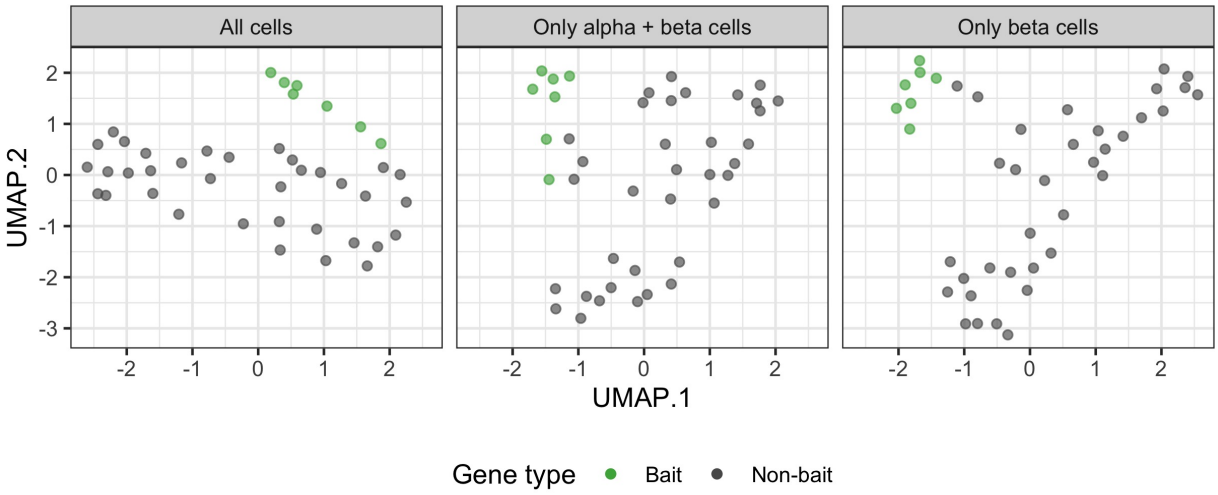


(b) Non-bait sample 2.

Figure 3.7: Similarity for three different similarity metrics in two samples of non-bait genes in the liver. Using genes selected from cholesterol metabolic process GO term as bait. In both (a) and (b) we are able to get clear clusters of bait genes using Spearman's rank correlation. Cosine similarity gives good separation in (a) but not (b), and Pearson's correlation has the worst separation. These are just two samples of non-bait genes, but it is representative as we see in Figure 3 a of the main text.



(a) Non-bait sample 1.



(b) Non-bait sample 2.

Figure 3.8: Tightness of 7 bait genes selected from insulin secretion GO term in only beta cells in pancreas. We can see that in all cell groups, even beta cells, the tightness is relatively high compared to Figure 3 in the main text, but we are able to get tighter clustering when only using beta cells than with all cells or when we add alpha cells.

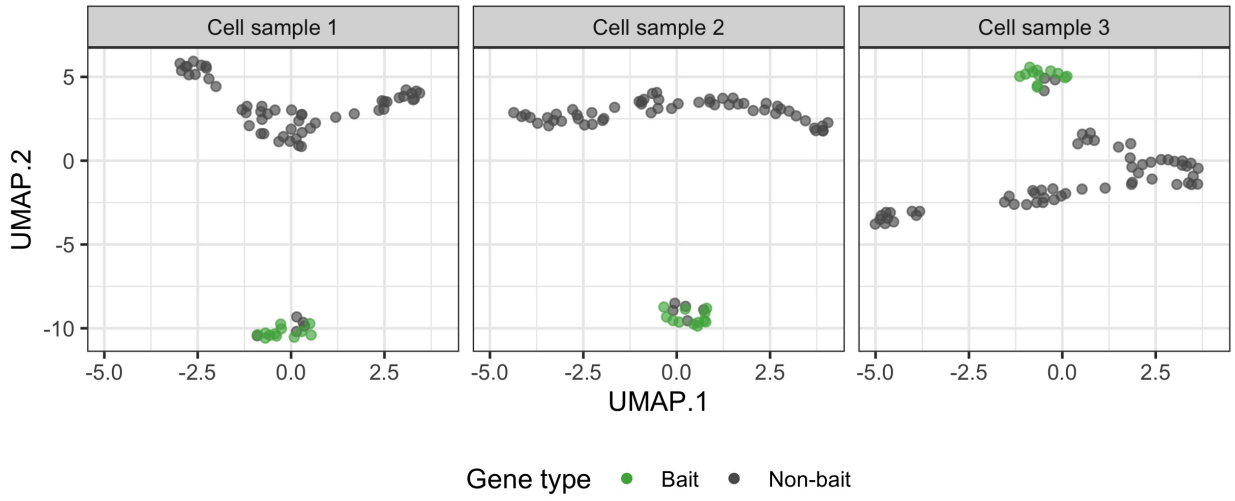


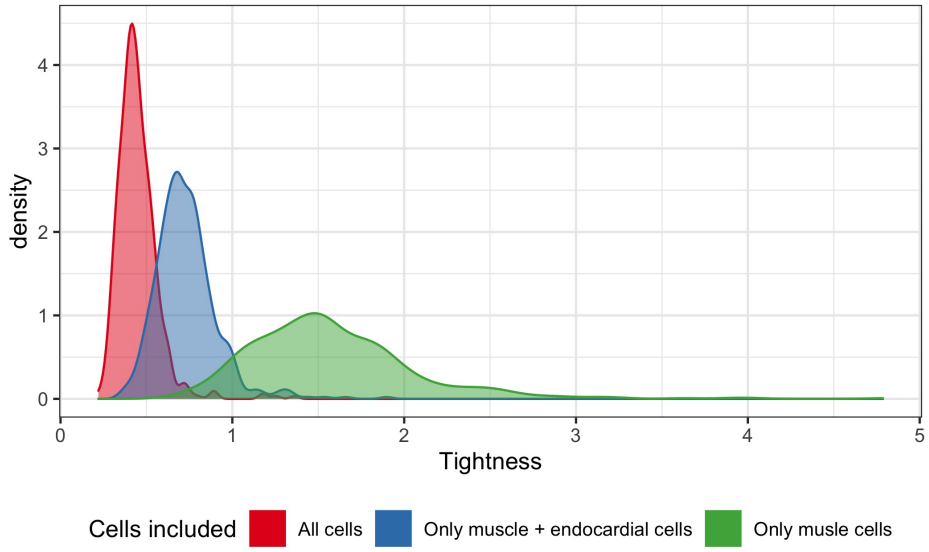
Figure 3.9: Tightness of bait genes selected from insulin secretion GO term when randomly sampling pancreas 439 cells (the number of beta cells in the data). This shows three different samples of cells, and in each sample we still see clear clustering of the bait genes. This illustrates that the increased co-expression is not simply due to a smaller number of cells, but more likely due to the contrast in cells that allows for co-expression to be seen.

the bait genes as opposed to the cell type with the highest expression of the biological process of interest. This indicates that understanding gene-gene co-expression requires contrast in expression levels within the cells being used to compute that correlation exists in multiple datasets.

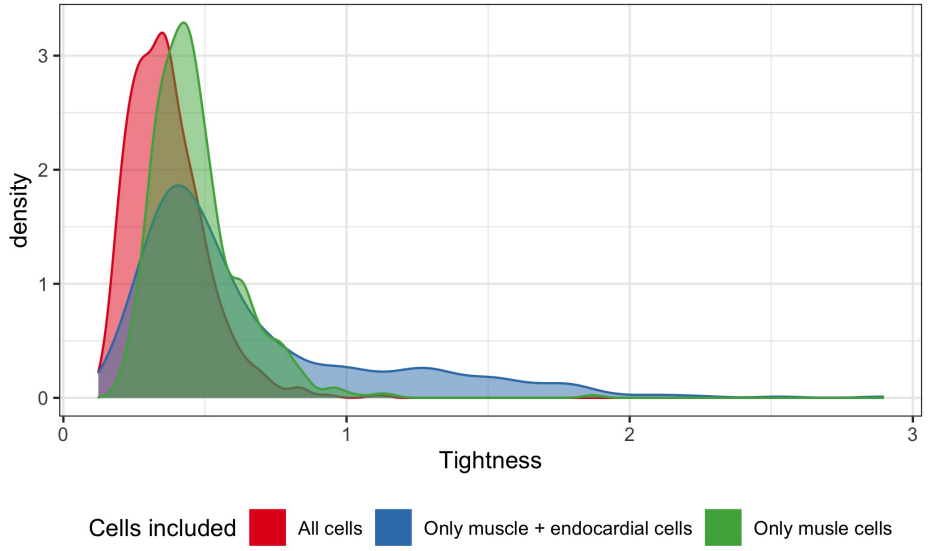
We hypothesize that the inability to detect strong co-expression is due to the homogeneity of gene expression that exists within each cell type. The beta cells are marked by having high expression of these insulin secretion genes, but with the group of cells which cells express which genes is somewhat random, potentially due to dropout. When we include just one additional group of cells, say the alpha cells, we now have samples with contrasting expression of the insulin secretion genes. This difference in expression along the sample is what allows us to see differentiate insulin secretion genes from other genes when using similarity metrics.

Using GeneFishing to refine cell type clustering in heart and aorta

A final interesting application of GeneFishing in scRNA-seq data is to use the fished out gene set as a context specific feature selection method in a sample of cells. As mentioned in the Section 1 selecting genes is an initial step that most algorithms require to prior to performing analysis. These selected features and then used in the downstream analyses, such as dimension reduction and clustering which are commonly used for creating cell atlases. In this section we show how using GeneFishing with a GO associated with a specific cell type



(a) Bait selected from all cells.



(b) Bait selected using only cardiac muscle cells.

Figure 3.10: Importance of cell heterogeneity to co-expression in a sample of Heart and Aorta cells. Here we are using the cardiac muscle contraction GO term as potential bait and find stronger correlation, using Spearman's rank correlation, among the selected bait when we use the bait selected from all cells in the data (a). If we only use bait selected from cardiac muscle cells and endocardial (b) we still have tighter co-expression when we use all cells to cluster the bait (red) as opposed to only the muscle cells (green).

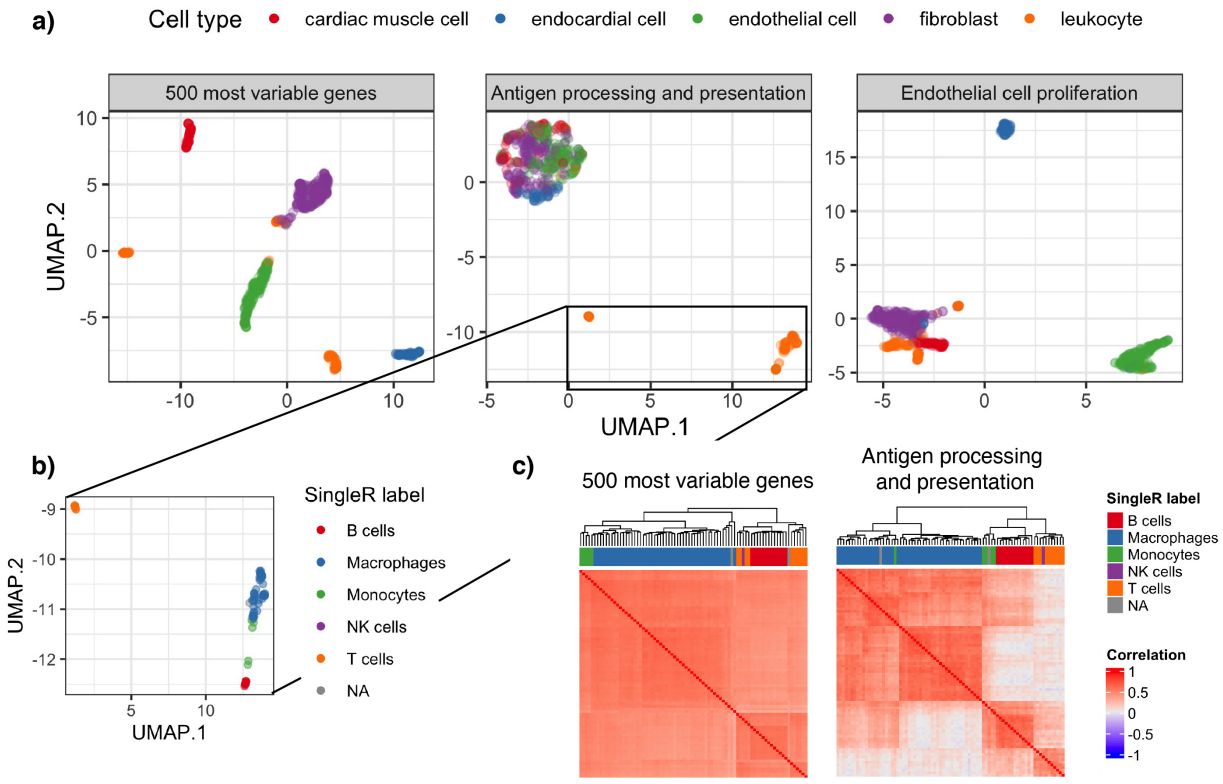


Figure 3.11: Using fished out genes to cluster cells. **a**) UMAP of heart and aorta cells using the 500 most variable genes (left), 354 genes that were fished out using genes from the antigen processing and presentation GO term as bait (middle), and 359 genes that were fished out using genes from the endothelial cell proliferation GO term as bait (right). **b**) Zooming in on middle plot showing the cell type labels from the `SingleR` package. **c**) Hierarchical clustering on the correlation matrix of all cells in **b**) using the 500 most variable genes (left) and 354 genes that were fished out using genes from the antigen processing and presentation GO term as bait (right).

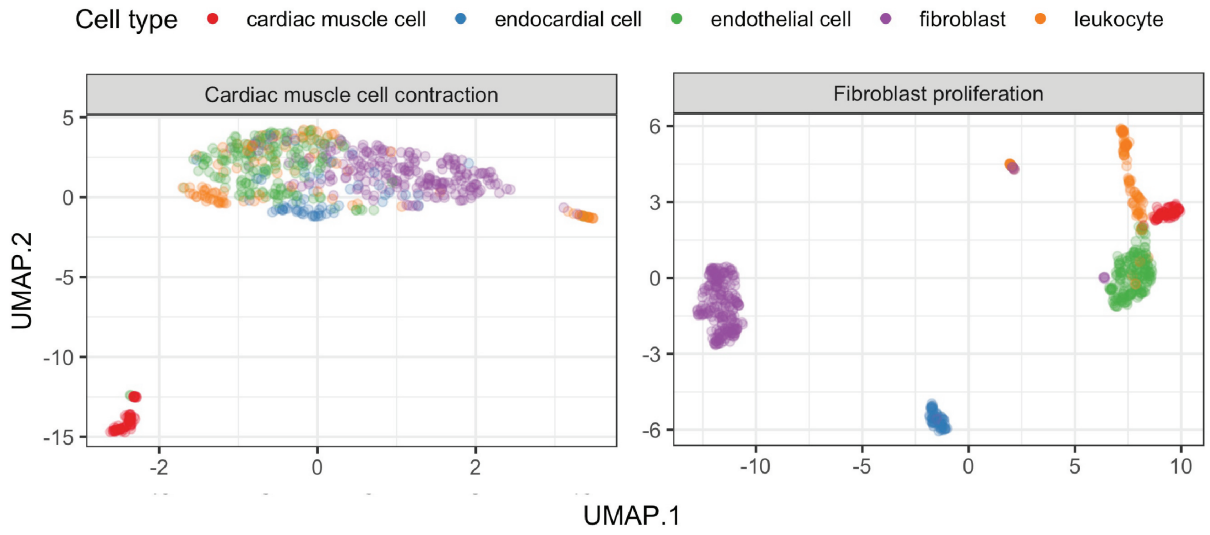


Figure 3.12: Cell type separation from a heart and aorta single cell sample when using various GO terms corresponding to the additional cell types in the data.

can be used as features for isolating that cell type from the remaining cell types. In addition to isolating the cell type we also illustrate how the fished out genes help elucidate cell type heterogeneity within the targeted cell type.

In this analysis we consider a sample of 654 heart and aorta cells which was collected using microfluidic droplet-based 3'-end counting technology. After removing performing quality control on the genes and cells we were left with 9,312 genes measure in 619 cells. There are five cell types in this data (57 cardiac muscle cells, 63 endocardial cells, 176 endothelial cells, 97 leukocytes, and 226 fibroblast cells). For each cell type, with the exception of endocardial cells, which are a subtype of endothelial cells, we found a GO term that describes a function expected to be done by the cell type and using the automatic bait selection and GeneFishing algorithms to find highly related genes. The fishing results, including enrichment analyses are included in the Supplemental Information. We then used those fished out genes, along with the discovered bait set, as features to cluster the cells with the hopes of isolating the targeted cell type from all other cells, but not necessarily separating other cell types.

We show two examples of this in Figure 3.11 and additional results for GO terms related to endothelial cells and fibroblasts are included in the Figures 3.12. To isolate leukocytes (white blood cells) we used the antigen processing and presentation GO term (GO:0019882). White blood cells are part of the immune system and thus rely on antigen presentation to function properly. The antigen processing and presentation GO term has 114 genes, 74 of which had expression high enough to be included in the analysis. Using UMAP coordinates we found a set of 11 bait genes, mostly coming from the major histocompatibility complex II, and were able to fish out 337 genes (excluding the bait set). The 337 genes are enriched for

biologically relevant pathways like osteoclast differentiation and B cell receptor signaling and GO terms such as innate immune response, adaptive immune response and inflammatory response.

We can see in Figure 3.11 a) when we compare the leftmost panel, which shows the UMAP coordinates for the cells based on the 500 most variable genes that there are three groups of leukocytes, one of which is indistinguishable from the fibroblast cluster. When we instead use these 354 genes from GeneFishing we still see three groups of these cells, but the group of labeled leukocyte cells that were clustering with fibroblasts are now in the cloud of cells including all other cell types. Using `SingleR` [9] we annotated the leukocyte cells and found that these outlier cells are more likely to be fibroblast cells than leukocytes. This is supported by the fact that when using primarily immune system genes we are unable to separate that group from the other cell types.

In Figures 3.11 b) we show that the resulting clusters separated by cell type labels from `SingleR` illustrating that these fished out genes are maintaining the ability to see cellular heterogeneity within the leukocyte group, despite not being able to in the large cloud of other cells. In Figure 3.11 c) we compare the hierarchical clustering of the cells in Figure 3.11 b) when using the 500 most variable genes to do the clustering versus the fished out genes from the antigen processing and presentation GO term. Both sets of genes produce a clustering that roughly matches that of the SingleR labels, however the ratio of within group to out of group correlation is much higher with the fished out genes than the variable genes. This signals that we are selecting genes with more ability to distinguish between subtypes of leukocytes than simply using variable genes.

In addition to the antigen processing and presentation GO term we also looked at the endothethial cell proliferation GO term (GO:0001935) which has 153 genes. 108 of which passed the threshold to be included in the analysis. Endothelial cells are cells which line the blood vessel. We found a bait set of 21 genes which are involved in pathways like Rap1 signaling, which is a regulator of endothelial barrier function [72]. We fish out 359 non-bait genes which are enriched for biologically relevant pathways like Rap1 signaling [72] and GO terms such as angiogenesis (the formation of new blood cells) and establishment of endothelial barrier.

We can see that these genes are mostly able to separate endothelial cells as well as endocardial celll, a subtype of endothelial cells that are found lining the way of the heart, from the remainder of the data. There is an exception of 4 cells which are labeled by Tabula Muris as endocardial that are not clustering with the other endocardial cells using endothelial cell proliferation but when we use the most variable genes. We compared those cells to the SingleR reference. There are no endocardial cells in the reference, but those 4 cells had the lower similarity to the endothelial cells than any other cell labeled as endocardial. The correlation of the endocardial cells with the SingleR endothelial label is shown in Figures 3.13.

Additionally, we looked for genes with large differences in expression betwen those 4 outlier endocardial cells the endocardial cluster using the `scran` R package [58]. As is depicted in Figure 3.14, we found many genes with low expression in the outlier cells including a number of genes with documented importance in endothelial and/or endocardial cell devel-

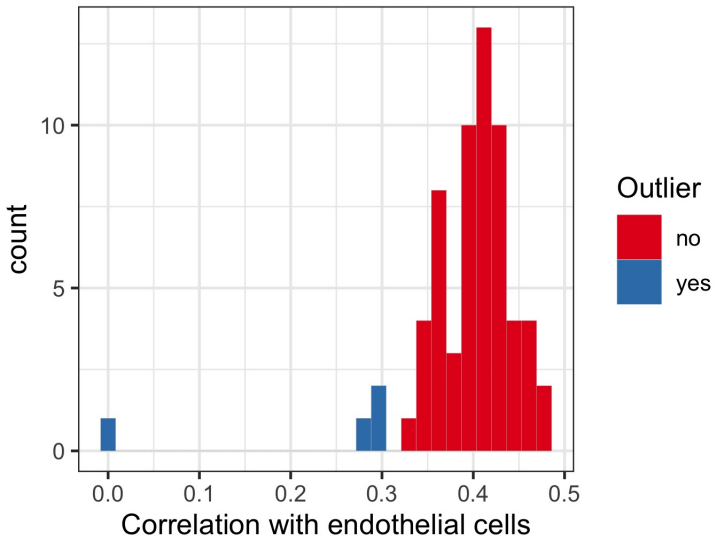


Figure 3.13: Correlation with endothelial reference cells using `SingleR` for all endocardial cells. Note, that there was no endocardial reference and endocardial cells are a subtype of endothelial cells. The cells marked in blue are the 4 endocardial cells from Figure 3 in the main text that did not cluster with the others. `SingleR` labels all these cells as endocardial, but the four outliers do have the least correlation with the reference, and thus are less likely to truly be of that cell type.

opment and function such as *Ephb4* [60], *Arhgap29* [110], *Ablim1* [77] and *Rasip1* [110]. Further investigation would be needed to determine whether these cells are truly mislabeled, but there at least some indication of a difference in those outliers that would not be picked up by simply using the most variable genes.

We performed a similar analysis using the pancreas data from Section 3.3 and similarly found that GeneFishing results using the insulin secretion or glucose homeostasis GO terms isolated pancreatic beta cells. The cell type clustering for the pancreas cells is shown in Figure 3.15, where we show that with bait derived from the insulin secretion and glucose homeostatsis GO terms we can separate pancreatic beta and delta cells from the remaining cells in the data. Additionally, we performed marker gene analysis using only the fished out genes we found that GeneFishing isolated a number of genes that mark the pancreatic beta cells such as insulin encoding genes *Ins1* and *Ins2* as well as the *Iapp* gene which encodes for proteins released by the pancreas in conjunction with insulin [47]. This is shown in Figure 3.16.

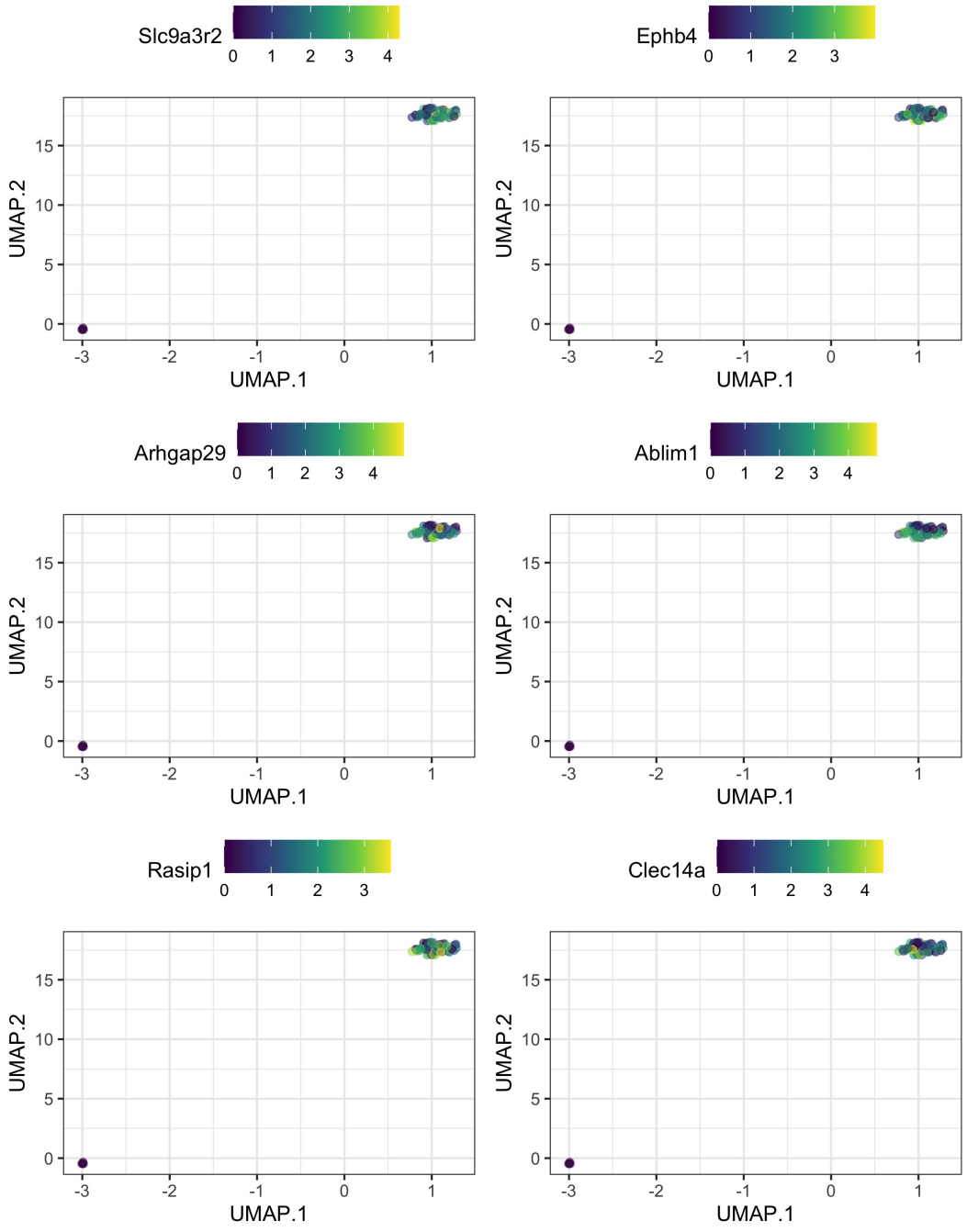


Figure 3.14: Gene markers, selected only from the fished out gene using genes from endothelial cell proliferation GO term as bait, that differentiate the large cluster endocardial cells from the four outlier cells in main text Figure 3

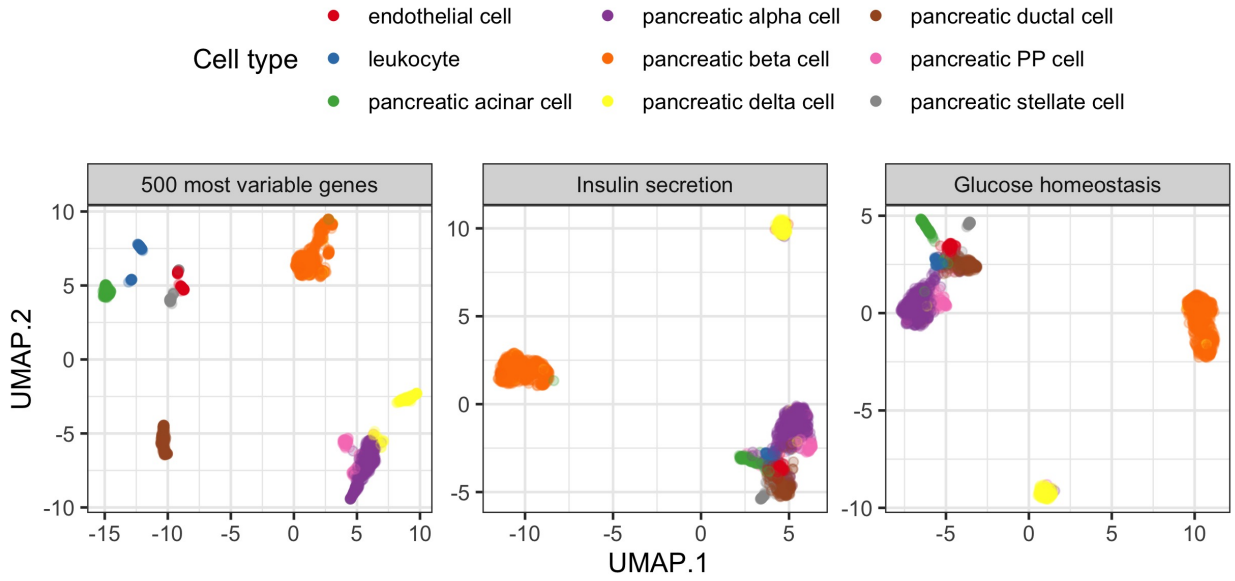


Figure 3.15: Cell type separation from a pancreas single cell sample when using various GO terms corresponding to the individual cell types in the data.

Overall, we are able to use the GeneFishing results as a context specific feature selection method that can help isolate individual cell types by pulling out biologically relevant genes. This is useful in single cell analysis as it may be able to help elucidate cell type clusters that have large overlap when using all cells. We have seen here that is can also maintain subgroupings of larger cell types, in the case of leukocytes, and thus could be used to discover finer structure in the data and potentially assist in relabeling cell types that were previously mislabeled.

3.4 Discussion

GeneFishing is a semi-supervised learning method that utilizes subsampling, dimension reduction, and clustering to uncover signal in large datasets. The method is applicable in a wide range of scenarios where the user *a priori* knows a group of items that are thought to be associated. The automatic bait selection algorithm can be used to determine if the relationship between those items is present in the data. If they are, then GeneFishing can be applied, to search for other observations in the data with similar expression. In this paper we illustrate its application in single cell genomics datasets as tool to discover new genes related to a biological process of interest.

We are able to apply GeneFishing to scRNA-seq by using UMAP prior to clustering. Typically UMAP is used a visualization tool in cell type clustering, but we show its impact

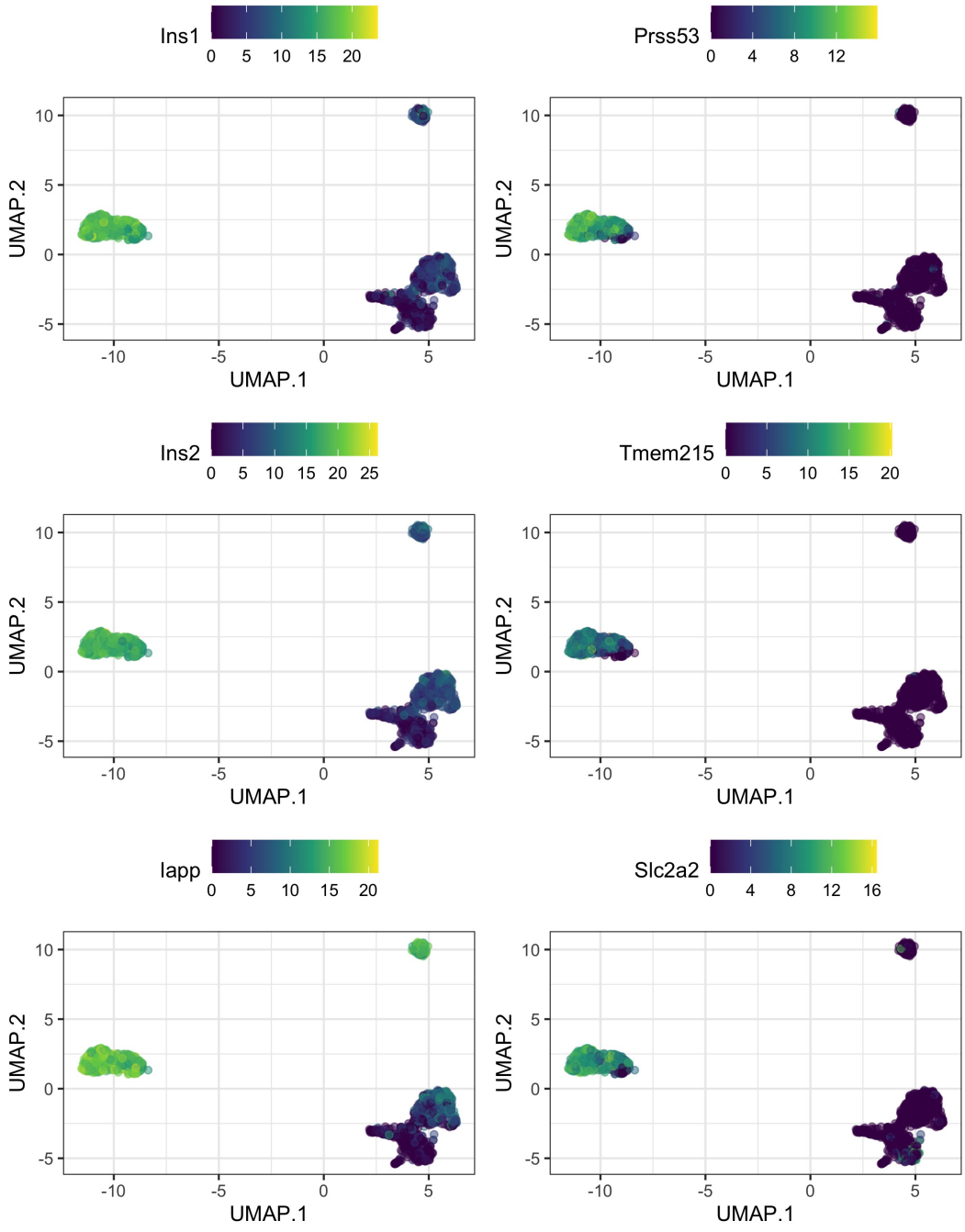


Figure 3.16: Gene markers, selected only from the fished out gene using genes from insulin secretion GO term as bait, that differentiate the pancreatic delta cell and pancreatic beta cell clusters in Figure 3.15

extends into gene clustering. It allows us to see distinct groupings among biologically related bait genes that would otherwise be overwhelmed by noise when we use standard techniques such as spectral clustering.

Additionally, GeneFishing in scRNA-seq data is not only useful as a gene discovery method, but can be used to understand better ways to measure gene-gene similarity in this data. This is important, because choosing the appropriate way to quantify relationships between genes in single cell data is difficult [89] and additional information about how the data and metric of association are related can lead to increased understanding. We show that that cellular heterogeneity is important when assessing similarity among genes as a largely homogeneous sample of cells does not provide the necessary contrast to cluster biologically related genes.

The output of GeneFishing can also assist in downstream analyses, such as cell type clustering, when used as a feature selection method. By starting with a group of genes known to be expressed in a particularly cell type that is thought to be present in the data the fished out genes can be used as features for isolating the cell type of interest, and potentially uncovering heterogeneity among the targeted cell type. This can be particularly useful if applied to datasets with hard to separate clusters of cells.

To summarize, the extensions in this paper allow for GeneFishing to be used in scRNA-sequencing data. The code to perform the automatic bait selection and GeneFishing is available in the `scGeneFishing` R package. In conjunction with its intended use of understanding function of genes the method has additional applications that are unique to single cell data.

Chapter 4

Using regional volumetric data to predict clinical progression in multiple sclerosis

4.1 Introduction

Multiple sclerosis (MS) is a neurodegenerative disease that affects the brain and spinal cord through inflammation and demyelination. Patients with MS experience a wide range of symptoms, including loss of coordination and vision impairment. The majority of these symptoms are physical disabilities, however individuals often experience cognitive impairment, such as decreases in processing speed.

In addition to the variety of symptoms that are associated with MS there are also multiple disease courses which are marked by varying patterns of progression. The majority of MS patients have the relapsing-remitting course, RRMS. In RRMS individuals experience attacks of new symptoms followed by periods of partial or full recovery. In this paper we will focus on the RRMS disease course.

Magnetic resonance imaging (MRI) is used are a common clinical tool in MS care to help determine disease course as well as to assess neurological damage. MRIs can be used to track brain atrophy and monitor lesions. There has been much research to investigate the relationship between MRI features and clinical outcomes. Studies have found associations of disability with MRI features, such as whole brain volume and T2-lesion volume [21, 17, 2]. In addition to finding correlations between clinical outcomes and MR imaging features, some researchers have looked into using these imaging features to predict disease progression with varying success [16, 29, 56].

In addition to the work above that uses features describing changes across the whole brain, there has been some work to understand the relationship of regional brain measurement and progression. A few studies have found associations between disease course and regional atrophy [71] as well as with clinical outcomes [31, 11, 13].

The aforementioned are studies are focused on determining relationships between regional atrophy and clinical outcomes. In this chapter we look to expand this research by using regional volumetric data to predict progression at 96 weeks for patients with RRMS. This is a difficult problem, because studies of MS are often small and given the volatile nature of progression in RRMS it is difficult to assess progression when data are collected over only a few years. Despite these challenges, we show that by clustering voxels to define subregions of interest we are able to achieve good predictive performance and outperform models that use no regional volumetric data.

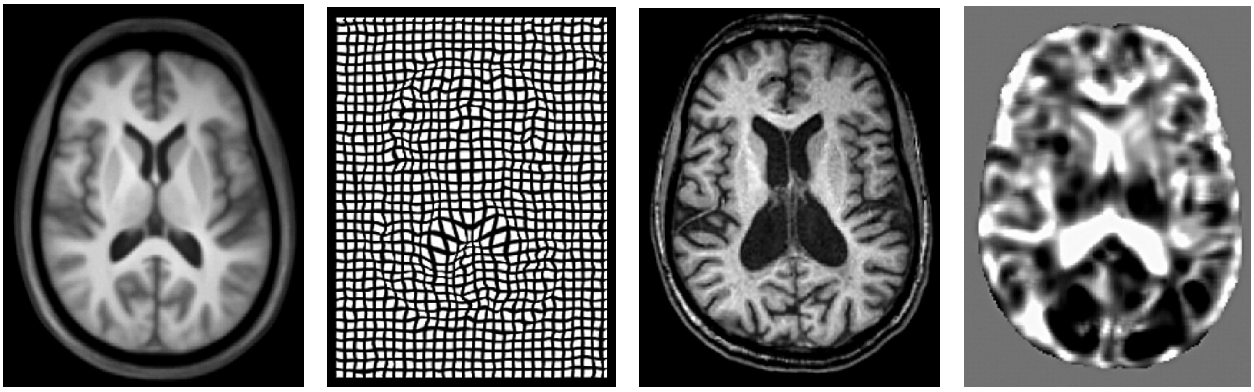
4.2 Data

The data in this chapter is from the control arm of the phase 3 clinical trial OPERA I (NCT01247324) [36]. The 411 patients in this arm received interferon beta-1a, which was at the time the standard of care for RRMS. These patients were followed for 96 weeks and a variety of demographic, clinical, and MRI data were collected throughout the trial.

We use the following baseline demographic, clinical, and and traditional MRI variables to predict 24 week confirmed disease progression:

- Demographic: age, sex, weight, ethnicity, race, and region
- Clinical:
  - Expanded disability scale score (EDSS): EDSS is an overall clinical score of MS disability. It is heavily weighted by physical disabilities.
  - Timed 25 foot walking test (25FWT): The 25FWT is a test of motor disability where patients are timed while walking 25 feet.
  - 9 hole peg test (9HPT): The 9HPT is a test of upper limb disability where patients place pegs in a set of 9 holes.
  - Single digit modality score (SSDMT): The SSDMT is a common test of cognitive disability.
- Traditional MRI
  - T2 lesion volume (T2VOL)
  - Whole brain volume (BNVOL)

We choose these variables based on inclusion in previous MS analyses. In addition to the variables listed above we have regional volumetric data of the brain extracted from MRI scans. This regional volumetric data was processed via deformation based morphometry (DBM), which aligned each patients baseline MRI scan to a common reference and compared the expansion or shrinkage of each voxel from a given patient to the reference [10]. The DBM data is in the form of a Jacobian map which describes the voxel-level transformation



(a) Reference image (b) Deformation grid (c) Patient image (d) Jacobian map

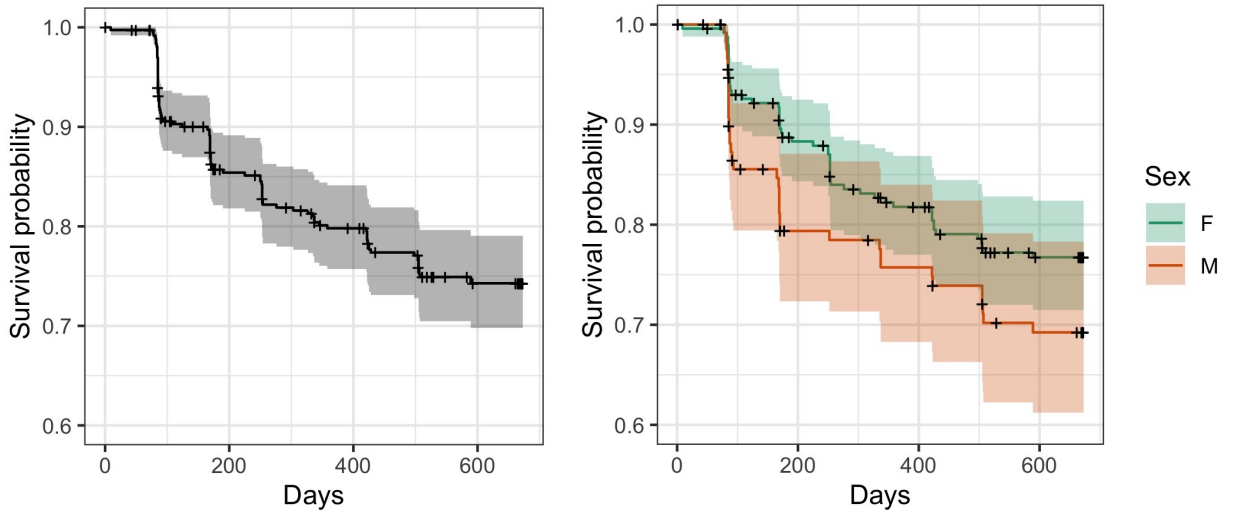
Figure 4.1: Schematic of deformation based morphometry (DBM) adapted from [49]. Each patient image (c) is aligned with the reference image (a). The deformation grid (b) illustrates how the voxels from the reference image are distorted to match the patient image. The result is a Jacobian map (d) where light colors indicate an expansion of the voxel in the patient and dark colors indicate a shrinkage of the voxel in the patient. For example, the cerebrospinal fluid in the center of the brain (the darkets area on the reference image) is much larger in the patient. This is indicated by the bright white areas in the center of the Jacobian map.

to the reference where a Jacobian value greater than 1 indicates a larger voxel in the patient compared to the reference and a Jacobian value less than 1 indicates the patient voxel is smaller than the corresponding voxel in the reference. An schematic of how these Jacobian maps are created is included Figure 4.1.

After dropping patients with missing baseline data in any of the variables listed above, we were left with 371 individuals. Of these remaining patients 246 are females and 125 are males. This is consistent with the breakdown of this disease in the general population as approximately two thirds of MS patients are female.

Due to the nature of this data we will model using survival analysis, where we fit a Cox proportional hazards model to predict disease progression. Of the 371 patients with full baseline data, 90 have composite confirmed disease progression at 24 weeks (CCDP24). CCDP24 is the outcome that we will be predicting in this study. It is defined by a patient having 24 weeks of sustained disability progression in EDSS, the 25FWT, and/or the 9HPT. We choose CCDP24 as the outcome for this study due to the relatively large number of events as compared to the other clinical outcomes. It should be noted that EDSS is primarily measuring physical disability, as is the 25FWT and 9HPT test. In future work, it may be interesting to use this data to model additional clinical outcomes. Potentially, these different models could provide important information about the relationship between regional brain volume and different symptoms.

The Kaplan Meier curve for CCDP24 in these patients is plotted in Figure 4.2. In the



(a) All patients

(b) Separately by sex

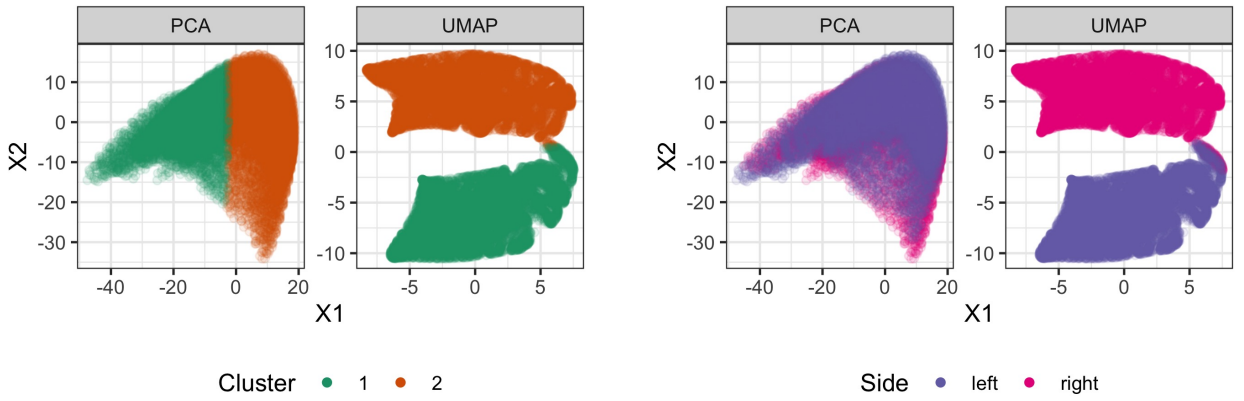
Figure 4.2: Kaplan-Meier curves for CCDP24 using (a) all patients and (b) separating by sex. In this dataset there is a higher proportion of progression in the male patients.

Figure 4.2 we show the survival for (a) all patients as well as (b) separating by sex. We can see that there are a higher proportion of cases of progression in males than females (35 out of 125 vs. 54 out of 246). However, using the log-rank test for differences in survival curves, we found no significant difference between the two sexes (Chi-sq value of 2.7 and p-value of 0.1). We show the sex-difference in progression, because we find in this data there are additional distinctions between sexes when predicting progression in the male and female groups separately. This may be due to random noise, but it is worth investigating in future research whether this difference holds in additional datasets.

4.3 Methods and results

As mentioned in the introduction the goal of this chapter is to predict disease progression using regional volumetric data dervied through DBM. Due to the large number of voxels in the raw DBM images, we can dramatically reduce our search space by summarizing the voxel level data within a region of interest (ROI). A ROI is defined by mapping images onto an atlas that registers voxels to pre-defined brain regions. The ROIs are made up of varying numbers of voxels and for each patient at each time point we can compute statistics such as the mean, standard deviation, and skewness over those voxels. In doing so we now have a relatively small set of features that describe the volume within each region.

Doing this summarization with the atlas defined ROIs we found they provided limited ability to improve prediction of disease progression beyond what can be achieved using other



(a) All patients

(b) Separately by sex

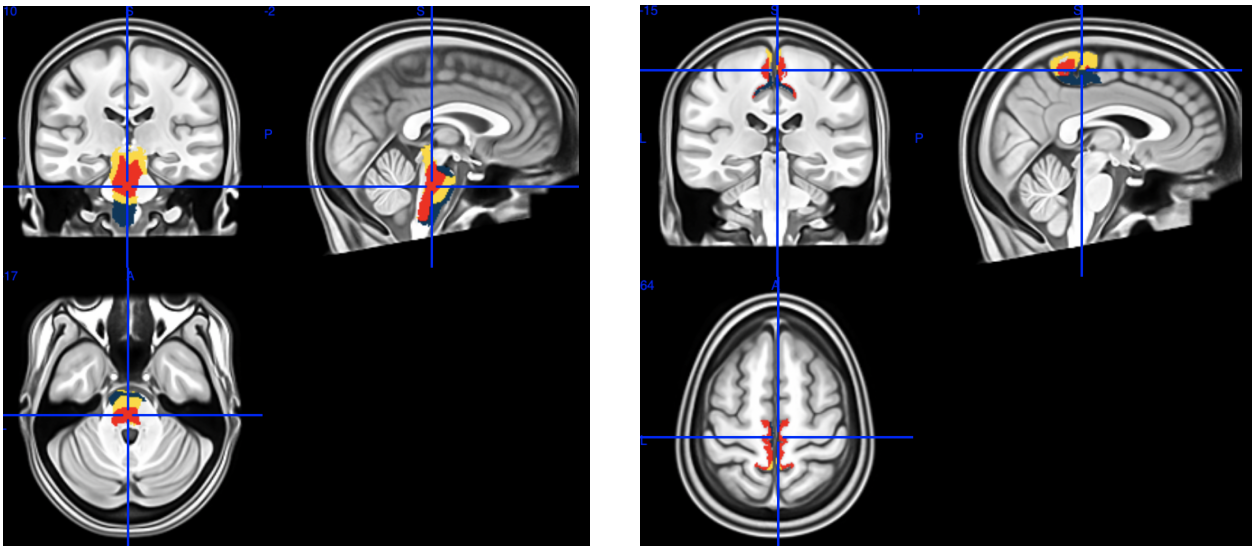
Figure 4.3: Projection of thalamus proper voxels onto the PCA and UMAP coordinates coloring by (a) clustering label from data-driven ROI definition and (b) side of brain.

demographic, clinical, and MRI features. We hypothesize that this is due to the atlas ROIs encompassing voxels with high within patient heterogeneity and thus when we summarize over all voxels in a ROI we may be losing the important information that is contained within that heterogeneity. For example, consider a patient with DBM values less than 1, e.g. low volume as compared to the reference, in a small portion of an ROI but DBM values slightly larger than 1, e.g. larger volume than the reference, in the remaining voxels. When we summarize over the ROI, the information in that small atrophied region will likely be lost in the noise. Hypothetically, if this were the case for all patients who progress and in patients who do not progress that atrophied area does not exist this will not be discovered.

Defining data-driven regions of interest

To overcome this hurdle, we defined data-driven ROIs, which split the original ROIs into multiple groups. These new groups are determined by clustering the voxels based on the distribution of Jacobian values across the patients. By doing this, voxels with similar patterns across patients will be grouped together and separate from other voxels in the same ROI with dissimilar patterns.

To do this clustering, for each of the 51 original ROIs, we apply two dimension reduction techniques to the $m_k \times n$ matrix X_k. Here m_k is the number of voxels in ROI k and n is the number of patients. We first do principal component analysis (PCA) on X_k to reduce the number of features used to represent each voxel to 20. Then we use uniform manifold projection (UMAP) [63] to further reduce the data, so that each voxel is represented by 4 features. UMAP is a non-linear dimension reduction method that is commonly used in single cell data analysis. It succinctly summarizes the information in the 20 PCs into a lower



(a) Brain stem (b) Paracentral

Figure 4.4: Projection of data-driven ROIs in the (a) brain stem and (b) paracentral. For both of these regions we subset the original ROIs into three groups indicated by the red, yellow and blue colors in the figures.

dimension, which is helpful for distance based clustering like k-means. In Figure 4.3 we show the first two PCs and UMAP coordinates in the thalamus proper as well as. The cluster labels are from applying k-means to the PCs and UMAPs respectively, while the left/right labels in (b) indicate the side of the brain. We can see that the UMAP coordinates produce an anatomically meaningful cluster structure that essentially separates the right lobe from the left, which is not the case if we use the PCs.

After reducing each ROI into a $m_k \times 4$ matrix we use k-means to cluster the data and choose the number of clusters which correspond to the highest average silhouette width [84]. The average silhouette width is a commonly used index for selecting a number of clusters between 2 and 10. Average silhouette width is a statistic that looks for the clustering which produces labels where the average distance between points in the same cluster being small when compared to the average distance between points in different clusters. This procedure, while it has been shown to work well in application, is ad hoc. We attempted other clustering approaches, such as DBCSAN [32] and HDBSCAN [62] that automatically select a cluster number, but we found they split the regions into too many groups.

In Figure 4.4 we show the clustering results in two atlas-based ROIs, the brain stem and the paracentral. In both regions we split the original into three data-driven sub regions. We can see that these data-driven cluster labels are grouping together voxels with close anatomical proximity. This is important because it shows that these clusters have potential for biological interpretation and indicates that the procedure is producing reasonable results.

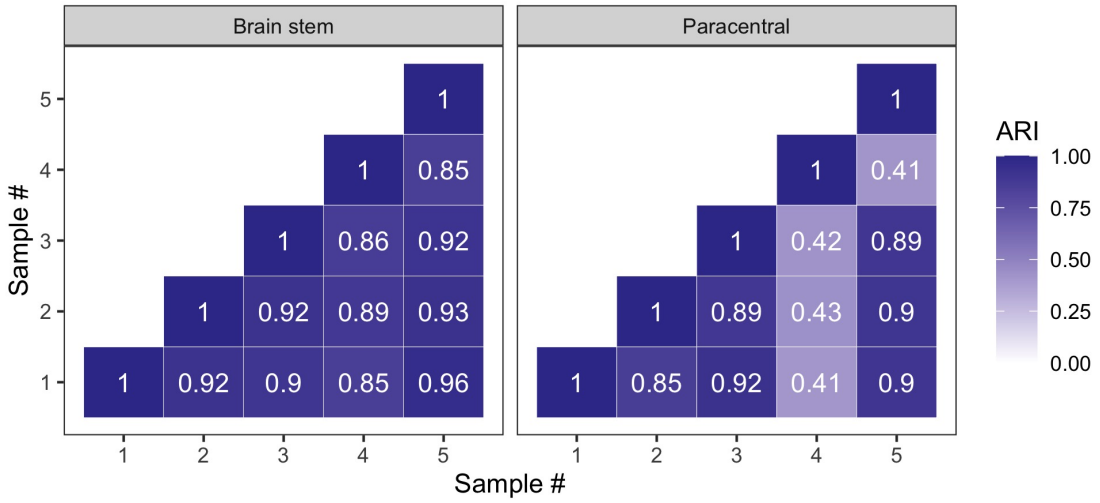


Figure 4.5: Adjusted rand index of voxel clustering in the brain stem and paracentral over five random samples of patients. We use the ARI to assess stability of the data-driven ROIs. A value close to 1 indicates strong agreement of clustering results, while a value close 0 indicates clustering results that are consistent with random partitions of the data.

As a further validation step for the derived data-driven ROIs, we compared clustering results over five random samples of patients in the paracentral and brain stem. In each sample we randomly selected 80% of the patients to be included in the dimension reduction and clustering. We used the adjusted rand index (ARI) to compare the clustering results over these five samples. The ARI is a common metric for assessing similarity between cluster labels that compares the number of pairs in the data that agree between two clusters (e.g. the points are in the same cluster in both labelings or in different clusters in both labelings) to the expected number of pairwise agreements under a random model [42]. ARI is a value between 0 and 1 where 1 indicates complete agreement between cluster results and a value close to 0 represents agreement consistent with randomly assigning a label to each voxel.

In Figure 4.5 we show the ARI over these five random samples in the brain stem and paracentral. We can see that there most values are close to 1 indicating strong similarity. There is one exception in the paracentral in the fourth row and column. In every other sample the region is split into 3 roughly equivalent regions, however in that sample the voxels corresponding to the yellow and blue regions in Figure 4.4 are further subdivided into two additional ROIs, creating 5 total clusters. In the brain stem we see consistent similarity across samples. Given that we only compared similarity across five samples and in two atlas-based ROIs it may be worth performing more thorough analysis with more samples and in more ROIs, however these results indicate good, albeit not perfect, stability of data-driven ROI labels. The stability promotes further confidence in the validity of the clustering results and potential for application in independent datasets.

Feature selection

After computing the mean, standard deviation, and skewness within each data-driven ROI for each patient at baseline we split the 51 original ROIs into 162 data-driven ROIs. For each of those we compute the mean, standard deviation and skewness, giving us 648 DBM features. We also tried computing robust statistics, such as the median and interquartile range, but found they performed similarity so that is excluded from this report. We also found that kurtosis was not a good predictor of progression in the original ROIs and thus it was left out of the analysis.

Due to the small sample size in the data, we perform an initial feature selection step to reduce the number of DBM features included in our predictive model. We fit 648 logistic regressions of our outcome of interest, CCDP24, for each DBM feature, while controlling for the demographic, clinical, and non-DBM MRI variables that we will include in our final model. As a reminder, the demographic variables are age, sex, region, ethnicity and race. We also control for the following baseline clinical variables, weight, EDSS, 9HPT, and 25FWT as well as traditional MRI features of T2 lesion volume and whole brain volume.

After fitting these logistic regressions we selected all DBM features with a p-value less than 0.025 on the coefficient in the model. This left us with 14 data-driven ROI features to use in our final model. Note, we also compare the model with data-drive ROI features to that with original ROI features. Doing the same feature selection method for the original ROIs we had 11 features. A table with the features, as well as the coefficients and p-values from the logistic regions, are included in the Appendix Tables C.1 and C.2.

We also checked the stability of the selected features by subsampling 80% of the data and re-fitting the logistic regression models for each feature in all the data-driven and original ROI. After repeating this 100 times we checked where each of the features ranked within each subsample. We did this for the feature selection controlling for all features in the final model as well as only controlling for age, sex, and years since onset of the disease.

Figure 4.6 shows the comparison between the top features from the data-driven ROIs and the atlas-based ROIs. On the y-axis we see the percentile rank of the feature and the x-axis shows the top six features corresponding to each ROI type. We can see that despite there being more than three times as many features with the data-driven ROIs we get similar, and often better, stability across these subsamples when controlling for both feature sets. This indicates that there is no increased instability in selected features despite the larger feature set with the data-driven ROI. Another takeaway from this Figure is that the feature selection is imperfect, and there are some selected features that would not be selected in many sub samples. This is due to the high levels of noise in this data.

Note, that the feature selection procedure selects two different sets of features for each ROI type. Appendix Tables C.1 and C.2 has the names of the selected features and the corresponding coefficients and p-values from the logistic regression.

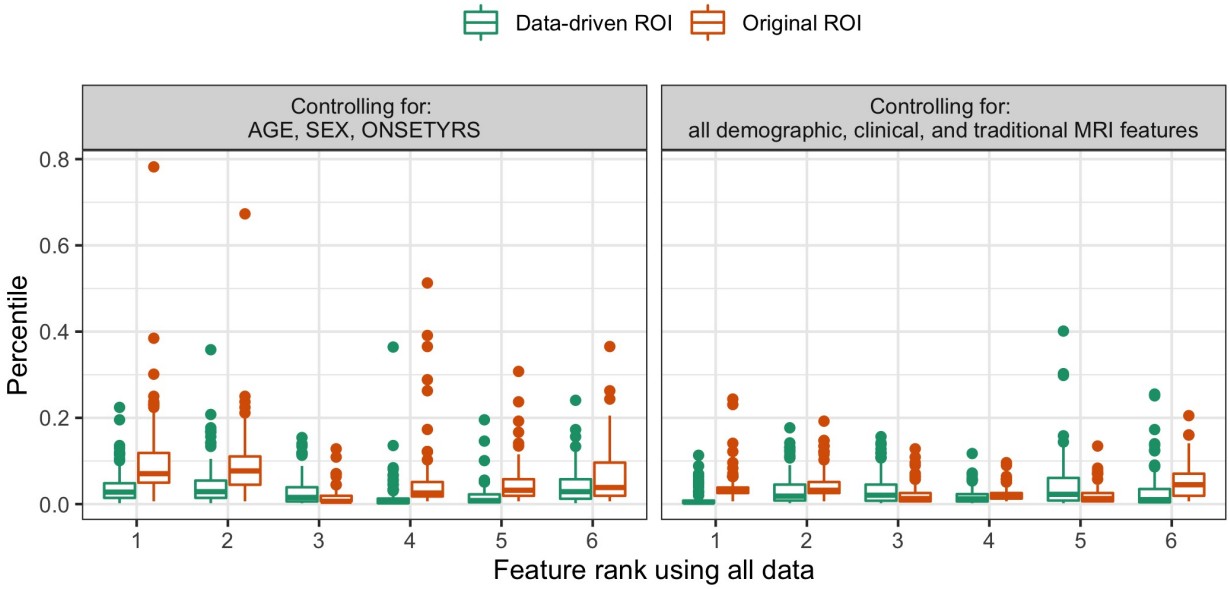


Figure 4.6: Percentile rank of each feature across 100 random samples of the data for the top 6 ranked features when using all data for feature selection. In the left panel we show the stability of the top-ranked features when controlling only for age, sex, and number of years since disease onset. In the right panel we control for all features listed in Section 4.2. We compare the data-driven ROI feature selection to the feature selection with the original ROI labels to ensure that the addition features are not leading to additional instability in the feature selection procedure.

Predicting progression

Following our feature selection procedure we fit a penalized Cox proportional hazards model to predict progression. We use a Cox PH model because this data is best modeled as time-to-event data due as a number of patients drop out of the study prior to the final 96 week follow up. We use the glmnet R package [34, 87] to fit a L2-penalized model.

Because we have an independent dataset from the OPERA II trial to validate our final model we compared potential models over multiple 80/20 train-test splits using the Concordance index (C-index) to determine model performance. For each train-test split we trained the penalized Cox PH model using 5-fold cross validation. The C-index compares the observed time-to-event to the predicted risk and counts the number of pairs of patients where those two values are consistent (e.g. we predict higher risk for a patient with observed progression prior to a second patient). It then compares the consistent, also called concordant pairs, to number of pairs of patients where the predicted risk is inconsistent, or discordant, with the observed timing.

In addition, to performing this analysis for the entire patient group we also did the feature

		No-DBM model	Original ROI model	Data-driven ROI model
All patients	CV	0.63	0.65	0.71
	Test	0.62	0.64	0.71
Female patients	CV	0.67	0.67	0.74
	Test	0.65	0.65	0.73
Male patients	CV	0.56	0.74	0.79
	Test	0.55	0.74	0.78

Table 4.1: Average C-index for the various models across the 100 train-test splits of the data. The CV values are the average of the average C-index over the 5 folds within each training split, while the test values are average C-index for each split on the held-out test set.

selection and training separately for males and females. The selected DBM variables that are used in features for both of the sexes are included in the Appendix for females in Tables C.3 and C.4 as well as the males in Tables C.5 and C.6.

The results over 100 train-test splits are shown below in Table 4.1 and Figure 4.7. Table 4.1 has the average of C-index from the cross validation (CV) and on the held out test set over these 100 samples for three models, (1) the model including all non-DBM features listed in Section 4.2, (2) the model using the non-DBM features as well as the selected DBM features from the original ROI labels, and (3) the model using the non-DBM features as well as the selected DBM features from the data-driven ROI labels. The cross validation We can see that on average the data-driven ROI model produces the highest CV C-index and test C-index in all patient groups.

Another thing to note from Table 4.1 is that the performance of the model without DBM features depends largely on the patients included in the model. When we model all patients or female patients only the performance is relatively good, with a C-index above 0.62. However, with the male patients the non-DBM features have much less predictive ability in this data. This is contrasted by the fact that the data-driven ROI performance is best in the male patient group, despite the lower sample size. This indicates, at least in the OPERA I trial, that regional volumetric data is better at predicting progression in male patients as compared to female patients. This observation warrants more investigation and validation in independent datasets.

Table 4.1 shows that on average the data-driven ROI model performs best, but we also compared the models directly within each train-test split by taking difference of the CV and test C-indices from the model with no DBM feature to the two models that include DBM features. The results are in Figure 4.7. Here we are plotting the average difference over the splits and the error bars represent plus or minus one standard deviation from that difference. All of the data-driven

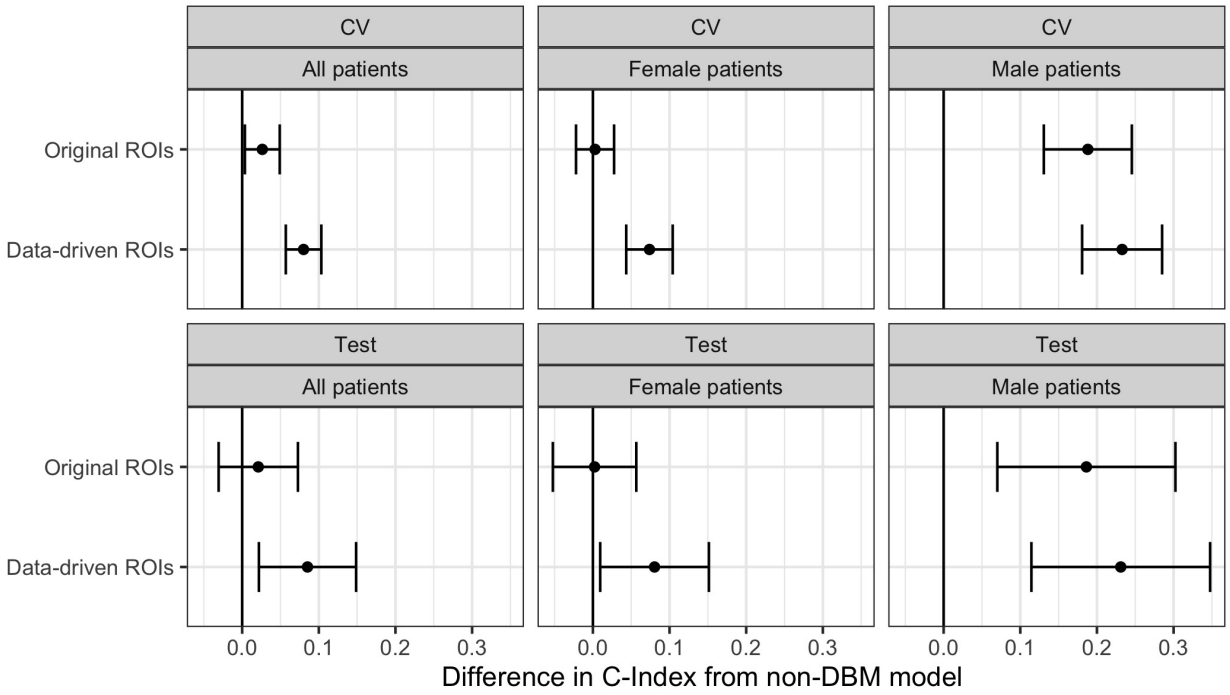
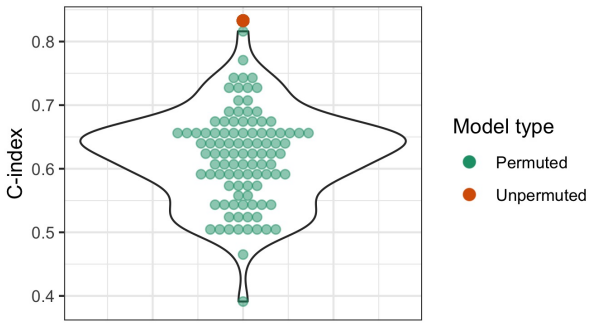


Figure 4.7: Comparison of C-index to the model that excludes DBM variables across 100 random 80/20 train test splits of the data. The top panel contains the difference between the average C-index from 5-fold CV within each sample. While the bottom panel has the difference between the C-index in the test data in each split. The error bars are one standard deviation of the differences across samples.

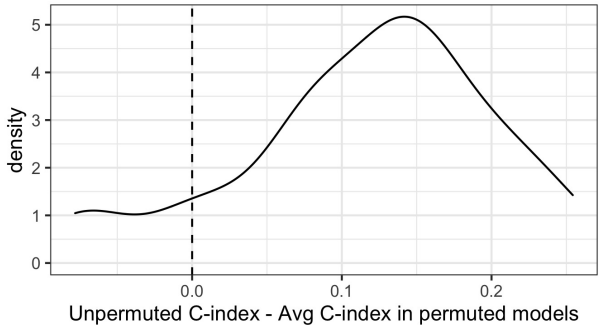
Testing for feature importance in prediction

We tested the importance of selected DBM features from the data-driven ROIs by randomly permuted their labels independently for each patient. In doing so we broke the connection between the individual DBM features and the data. The idea was that if we saw similar performance using the permuted model and the unpermuted model it would indicate that while the DBM data increases performance it is not due to the information within indivdual regions, rather from the overall value across regions.

We do this by randomly sampling 25 train-test splits and comparing the difference between 100 permuted models within each train-test split and the unpermuted model. The results are shown in Figure 4.8. In (a) we see the distribution of C-indices on the test set for the mutated models in a single train-test split. The orange point on this plot is the test set C-index for the unpermuted model. We can see that the unpermuted model outperforms the permuted model. In (b) we show the difference between the unpermuted model and the average permuted model across the 25 train-test splits. It is clear that permutation decreases



(a) Distribution of C-index



(b) Difference of permuted and unpermuted models

Figure 4.8: Comparing C-index from models where we randomly permute the DBM features for each patient to the unpermuted model. By doing these random permutations we break the relationship with the individual ROI labels. In (a) we look at the distribution of the permuted models in a single train/test split. In (b) we look at the difference between the average permuted C-index to the unpermuted model over 25 train/test splits.

the performance on the test set. This is showing that there is value in the selected ROIs.

4.4 Discussion and Future work

In this paper we were able to predict composite disease progression in multiple sclerosis using regional volumetric data extracted from MRIs along with other demographic, clinical, and traditional MRI features. We found using traditional atlas-based ROIs for the analysis was not able to provide any added benefit to prediction. However, by defining data-driven ROIs through voxel clustering we were able to derive regional volumetric features that added value to our predictive model.

This is an important observation indicating that there may be some important heterogeneity within atlas based ROIs that is lost when computing summary statistics over the regions. These data-driven regions initially appear to be relatively stable and have spatially consistent. This data-driven voxel clustering will require additional validation to ensure that it has biological meaning, but the prelimary results indicate it is a procedure that is successful. In future work we hope to better understand why this clustering procedure was beneficial as well as to determine if there is any biological interpretation of the derived regions.

In addition to validating the data-driven ROIs, further validation of the final model is needed by testing it on new data. OPERA II is a independent RRMS trial that is a natural fit to use for this purpose.

Finally, in this report we only utilize baseline regional volumetric data as a predictor of composite disease progression, however this is a rich dataset with many oppoturnities for

different model setups. For example, one could use regional atrophy data as a predictor or model different clinical outcomes to see if the relationship between regional volume and/or atrophy changes.

Bibliography

[1] Jennifer G Abelin et al. "Reduced-representation phosphosignatures measured by quantitative targeted MS capture cellular states and enable large-scale comparison of drug-induced phenotypes". In: *Molecular & Cellular Proteomics* 15.5 (2016), pp. 1622–1641.

[2] Martina Absinta et al. "Association of chronic active multiple sclerosis lesions with disability in vivo". In: *JAMA neurology* 76.12 (2019), pp. 1474–1483.

[3] Adrian Alexa, Jorg Rahnenfuhrer, et al. "topGO: enrichment analysis for gene ontology". In: *R package version* 2.0 (2010), p. 2010.

[4] Adrian Alexa and Jörg Rahnenführer. "Gene set enrichment analysis with topGO". In: *Bioconductor Improv* 27 (2009), pp. 1–26.

[5] Robert A Amezquita et al. "Orchestrating single-cell analysis with Bioconductor". In: *Nature methods* 17.2 (2020), pp. 137–145.

[6] Simon Anders and Wolfgang Huber. "Differential expression analysis for sequence count data". In: *Nature Precedings* (2010), pp. 1–1.

[7] Tallulah S Andrews and Martin Hemberg. "False signals induced by single-cell imputation". In: *F1000Research* 7 (2018).

[8] Tallulah S Andrews and Martin Hemberg. "M3Drop: dropout-based feature selection for scRNASeq". In: *Bioinformatics* 35.16 (2019), pp. 2865–2867.

[9] Dvir Aran et al. "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage". In: *Nature immunology* 20.2 (2019), pp. 163–172.

[10] John Ashburner et al. "Identifying global anatomical differences: Deformation-based morphometry". In: *Human brain mapping* 6.5-6 (1998), pp. 348–357.

[11] Rohit Bakshi et al. "Regional brain atrophy is associated with physical disability in multiple sclerosis: semiquantitative magnetic resonance imaging and relationship to clinical findings". In: *Journal of Neuroimaging* 11.2 (2001), pp. 129–136.

[12] MJ Ball et al. "Comparison of two triphasic contraceptives with different progestogens: effects on metabolism and coagulation proteins". In: *Contraception* 41.4 (1990), pp. 363–376.

[13] Marco Battaglini et al. "Voxel-wise assessment of progression of regional brain atrophy in relapsing-remitting multiple sclerosis". In: *Journal of the neurological sciences* 282.1-2 (2009), pp. 55–60.

[14] Francisco Beca and Kornelia Polyak. "Intratumor heterogeneity in breast cancer". In: *Novel biomarkers in the continuum of breast cancer* (2016), pp. 169–189.

[15] A Patrícia Bento et al. "The ChEMBL bioactivity database: an update". In: *Nucleic acids research* 42.D1 (2014), pp. D1083–D1090.

[16] Roberto Bergamaschi et al. "Predicting secondary progression in relapsing–remitting multiple sclerosis: a Bayesian analysis". In: *Journal of the neurological sciences* 189.1-2 (2001), pp. 13–21.

[17] Robert A Bermel and Rohit Bakshi. "The measurement and clinical relevance of brain atrophy in multiple sclerosis". In: *The Lancet Neurology* 5.2 (2006), pp. 158–170.

[18] Philip Brennecke et al. "Accounting for technical noise in single-cell RNA-seq experiments". In: *Nature methods* 10.11 (2013), pp. 1093–1095.

[19] Joanna K Chan, Vivian M Bruce, and Bruce E McDonald. "Dietary α-linolenic acid is as effective as oleic acid and linoleic acid in lowering blood cholesterol in normolipidemic men". In: *The American journal of clinical nutrition* 53.5 (1991), pp. 1230–1234.

[20] Thalia E Chan, Michael PH Stumpf, and Ann C Babtie. "Gene regulatory network inference from single-cell data using multivariate information measures". In: *Cell systems* 5.3 (2017), pp. 251–267.

[21] DT Chard et al. "Brain atrophy in clinically early relapsing–remitting multiple sclerosis". In: *Brain* 125.2 (2002), pp. 327–337.

[22] B Chen and AJ Butte. "Leveraging big data to transform target selection and drug discovery". In: *Clinical Pharmacology & Therapeutics* 99.3 (2016), pp. 285–297.

[23] Bin Chen et al. "Reversal of cancer gene expression correlates with drug efficacy and reveals therapeutic targets". In: *Nature communications* 8 (2017), p. 16022.

[24] Shuonan Chen and Jessica C Mar. "Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data". In: *BMC bioinformatics* 19.1 (2018), pp. 1–21.

[25] Jie Cheng and Lun Yang. "Comparing gene expression similarity metrics for connectivity map". In: *2013 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE. 2013, pp. 165–170.

[26] Jie Cheng et al. "Systematic evaluation of connectivity map for disease indications". In: *Genome medicine* 6.12 (2014), pp. 1–8.

[27] John Y.L. Chiang. "Bile acid metabolism and signaling in liver disease and therapy". In: *Liver Research* 1.1 (2017), pp. 3–9. ISSN: 2542-5684. DOI: https://doi.org/10.1016/j.livres.2017.05.001. URL: https://www.sciencedirect.com/science/article/pii/S2542568417000071.

[28] Tabula Muris Consortium et al. "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris". In: *Nature* 562.7727 (2018), pp. 367–372.

[29] I Dekker et al. "Predicting clinical progression in multiple sclerosis after 6 and 12 years". In: *European journal of neurology* 26.6 (2019), pp. 893–902.

[30] Joel T Dudley et al. "Computational repositioning of the anticonvulsant topiramate for inflammatory bowel disease". In: *Science translational medicine* 3.96 (2011), 96ra76–96ra76.

[31] Arman Eshaghi et al. "Progression of regional grey matter atrophy in multiple sclerosis". In: *Brain* 141.6 (2018), pp. 1665–1677.

[32] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[33] Kristen Fortney et al. "Prioritizing therapeutics for lung cancer: an integrative meta-analysis of cancer gene signatures and chemogenomic data". In: *PLoS Comput Biol* 11.3 (2015), e1004068.

[34] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1.

[35] A Ari Hakimi et al. "An integrated metabolic atlas of clear cell renal cell carcinoma". In: *Cancer cell* 29.1 (2016), pp. 104–116.

[36] Stephen L Hauser et al. "Ocrelizumab versus interferon beta-1a in relapsing multiple sclerosis". In: *New England Journal of Medicine* 376.3 (2017), pp. 221–234.

[37] Haley Hieronymus et al. "Gene expression signature-based chemical genomic prediction identifies a novel class of HSP90 pathway modulators". In: *Cancer cell* 10.4 (2006), pp. 321–330.

[38] Dominique Hillaire-Buys et al. "Evidence for an inhibitory A1 subtype adenosine receptor on pancreatic insulin-secreting cells". In: *European journal of pharmacology* 136.1 (1987), pp. 109–112.

[39] Marek Hlavac. "stargazer: beautiful LATEX, HTML and ASCII tables from R statistical output". In: (2015).

[40] Katherine A Hoadley et al. "Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin". In: *Cell* 158.4 (2014), pp. 929–944.

[41] Tom Hu. *GeneFishing R code*. URL: https://github.com/tomwhoooo/GeneFishingPy.

[42] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218.

[43] Vân Anh Huynh-Thu et al. "Inferring regulatory networks from expression data using tree-based methods". In: *PloS one* 5.9 (2010), e12776.

[44] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. "Single-cell RNA sequencing technologies and bioinformatics pipelines". In: *Experimental & molecular medicine* 50.8 (2018), pp. 1–14.

[45] Murat Iskar et al. "Drug-induced regulation of target expression". In: *PLoS computational biology* 6.9 (2010), e1000925.

[46] Hillary Johnston-Cox et al. "The A2b adenosine receptor modulates glucose homeostasis and obesity". In: *PloS one* 7.7 (2012), e40584.

[47] Steven E Kahn et al. "Evidence of cosecretion of islet amyloid polypeptide and insulin by β-cells". In: *Diabetes* 39.5 (1990), pp. 634–638.

[48] Peter V Kharchenko, Lev Silberstein, and David T Scadden. "Bayesian approach to single-cell differential expression analysis". In: *Nature methods* 11.7 (2014), pp. 740–742.

[49] Junghoon Kim et al. "Structural consequences of diffuse traumatic brain injury: a large deformation tensor-based morphometry study". In: *Neuroimage* 39.3 (2008), pp. 1014–1026.

[50] Seongho Kim. "ppcor: an R package for a fast calculation to semi-partial correlation coefficients". In: *Communications for statistical applications and methods* 22.6 (2015), p. 665.

[51] Steven D Kunkel et al. "mRNA expression signatures of human skeletal muscle atrophy identify a natural compound that increases muscle mass". In: *Cell metabolism* 13.6 (2011), pp. 627–638.

[52] Justin Lamb et al. "The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease". In: *science* 313.5795 (2006), pp. 1929–1935.

[53] *Library of Integrated Network-Based Cellular Signatures (LINCS). NIH LINCS Program.* `http://www.lincsproject.org`.

[54] Ke Liu. *GeneFishing R code.* URL: `https://github.com/iLukegogogo/GeneFishing`.

[55] Ke Liu et al. "GeneFishing to reconstruct context specific portraits of biological processes". In: *Proceedings of the National Academy of Sciences* 116.38 (2019), pp. 18943–18950.

[56] Alexis A Lizarraga and Bianca Weinstock-Guttman. "Multiple sclerosis in 2019: predicting progression". In: *The Lancet Neurology* 19.1 (2020), pp. 12–14.

[57] John Lonsdale et al. "The genotype-tissue expression (GTEx) project". In: *Nature genetics* 45.6 (2013), pp. 580–585.

[58] Aaron Lun et al. "Package 'scran'". In: (2017).

[59] Aaron TL Lun, Karsten Bach, and John C Marioni. "Pooling across cells to normalize single-cell RNA sequencing data with many zero counts". In: *Genome biology* 17.1 (2016), pp. 1–14.

[60] Guillermo Luxán et al. "Endothelial EphB4 maintains vascular integrity and transport function in adult heart". In: *elife* 8 (2019), e45863.

[61] Davis J McCarthy et al. "Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R". In: *Bioinformatics* 33.8 (2017), pp. 1179–1186.

[62] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *Journal of Open Source Software* 2.11 (2017), p. 205.

[63] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[64] I Mertens et al. "Among inflammation and coagulation markers, PAI-1 is a true component of the metabolic syndrome". In: *International journal of obesity* 30.8 (2006), pp. 1308–1314.

[65] Philipp Mertins et al. "Proteogenomics connects somatic mutations to signalling in breast cancer". In: *Nature* 534.7605 (2016), p. 55.

[66] Thomas Moerman et al. "GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks". In: *Bioinformatics* 35.12 (2019), pp. 2159–2161.

[67] Asher Mullard. "Parsing clinical success rates". In: *Nature Reviews Drug Discovery* 15.7 (2016), pp. 447–448.

[68] Aliyu Musa et al. "A review of connectivity map and computational approaches in pharmacogenomics". In: *Briefings in bioinformatics* 19.3 (2018), pp. 506–523.

[69] Andrew Y Ng, Michael I Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems*. 2002, pp. 849–856.

[70] Vera van Noort et al. "Novel drug candidates for the treatment of metastatic colorectal cancer through global inverse gene-expression profiling". In: *Cancer research* 74.20 (2014), pp. 5690–5699.

[71] Elisabetta Pagani et al. "Regional brain atrophy evolves differently in patients with multiple sclerosis according to clinical phenotype". In: *American Journal of Neuroradiology* 26.2 (2005), pp. 341–346.

[72] Willem-Jan Pannekoek, Anneke Post, and Johannes L Bos. "Rap1 signaling in endothelial barrier control". In: *Cell adhesion & migration* 8.2 (2014), pp. 100–107.

[73] Sunmin Park et al. "Central prolactin modulates insulin sensitivity and insulin secretion in diabetic rats". In: *Neuroendocrinology* 95.4 (2012), pp. 332–343.

[74] Ross G Pinsky. "Law of large numbers for increasing subsequences of random permutations". In: *Random Structures & Algorithms* 29.3 (2006), pp. 277–295.

[75] Aditya Pratapa et al. "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data". In: *Nature methods* 17.2 (2020), pp. 147–154.

[76] Xiaoyan A Qu and Deepak K Rajpal. "Applications of Connectivity Map in drug discovery and development". In: *Drug discovery today* 17.23-24 (2012), pp. 1289–1298.

[77] Pearl Quijada et al. "Coordination of endothelial cell positioning and fate specification by the epicardium". In: *Nature communications* 12.1 (2021), pp. 1–18.

[78] Aviv Regev et al. "Science forum: the human cell atlas". In: *elife* 6 (2017), e27041.

[79] Ajaya Kumar Reka et al. "Identifying inhibitors of epithelial-mesenchymal transition by connectivity map–based systems approach". In: *Journal of Thoracic Oncology* 6.11 (2011), pp. 1784–1792.

[80] Jason A Reuter, Damek V Spacek, and Michael P Snyder. "High-throughput sequencing technologies". In: *Molecular cell* 58.4 (2015), pp. 586–597.

[81] Tadeja Rezen et al. "Interplay between cholesterol and drug metabolism". In: *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* 1814.1 (2011), pp. 146–160.

[82] Seung Bae Rho, Boh-Ram Kim, and Sokbom Kang. "A gene signature-based approach identifies thioridazine as an inhibitor of phosphatidylinositol-3'-kinase (PI3K)/AKT pathway in ovarian cancer cells". In: *Gynecologic oncology* 120.1 (2011), pp. 121–127.

[83] Nathan Ross et al. "Fundamentals of Stein's method". In: *Probability Surveys* 8 (2011), pp. 210–293.

[84] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[85] Lina Schiffer et al. "Human steroid biosynthesis, metabolism and excretion are differentially reflected by serum and urine steroid metabolomes: A comprehensive review". In: *The Journal of Steroid Biochemistry and Molecular Biology* 194 (2019), p. 105439. ISSN: 0960-0760. DOI: https://doi.org/10.1016/j.jsbmb.2019.105439. URL: https://www.sciencedirect.com/science/article/pii/S0960076019302791.

[86] David B Seligson et al. "Global histone modification patterns predict risk of prostate cancer recurrence". In: *Nature* 435.7046 (2005), pp. 1262–1266.

[87] Noah Simon et al. "Regularization paths for Cox's proportional hazards model via coordinate descent". In: *Journal of statistical software* 39.5 (2011), p. 1.

[88] Marina Sirota et al. "Discovery and preclinical validation of drug indications using compendia of public gene expression data". In: *Science translational medicine* 3.96 (2011), 96ra77–96ra77.

[89] Michael A Skinnider, Jordan W Squair, and Leonard J Foster. "Evaluating measures of association for single-cell transcriptomics". In: *Nature methods* 16.5 (2019), pp. 381–386.

[90] Wendy Weijia Soon, Manoj Hariharan, and Michael P Snyder. "High-throughput sequencing for biology and medicine". In: *Molecular systems biology* 9.1 (2013), p. 640.

[91] Alicia T Specht and Jun Li. "LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering". In: *Bioinformatics* 33.5 (2017), pp. 764–766.

[92] Joshua M Stuart et al. "A gene-coexpression network for global discovery of conserved genetic modules". In: *science* 302.5643 (2003), pp. 249–255.

[93] Aravind Subramanian et al. "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles". In: *Cell* 171.6 (2017), pp. 1437–1452.

[94] Aravind Subramanian et al. "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550.

[95] Léon-Charles Tranchevent et al. "Candidate gene prioritization with Endeavour". In: *Nucleic acids research* 44.W1 (2016), W117–W121.

[96] Aviad Tsherniak et al. "Defining a cancer dependency map". In: *Cell* 170.3 (2017), pp. 564–576.

[97] Bram Van de Sande et al. "A scalable SCENIC workflow for single-cell gene regulatory network analysis". In: *Nature Protocols* 15.7 (2020), pp. 2247–2276.

[98] Erwin L Van Dijk et al. "Ten years of next-generation sequencing technology". In: *Trends in genetics* 30.9 (2014), pp. 418–426.

[99] M Von Eynatten et al. "Retinol-binding protein 4 is associated with components of the metabolic syndrome, but not with insulin resistance, in men with type 2 diabetes or coronary artery disease". In: *Diabetologia* 50.9 (2007), pp. 1930–1937.

[100] Allon Wagner, Aviv Regev, and Nir Yosef. "Revealing the vectors of cellular identity with single-cell genomics". In: *Nature biotechnology* 34.11 (2016), pp. 1145–1160.

[101] Allon Wagner et al. "Drugs that reverse disease transcriptomic signatures are more effective in a mouse model of dyslipidemia". In: *Molecular systems biology* 11.3 (2015), p. 791.

[102] Guiping Wang et al. "Expression-based in silico screening of candidate therapeutic compounds for lung adenocarcinoma". In: *PloS one* 6.1 (2011), e14573.

[103] Yong Wang and Nicholas E Navin. "Advances and applications of single-cell sequencing technologies". In: *Molecular cell* 58.4 (2015), pp. 598–609.

[104] YX Rachel Wang, Michael S Waterman, and Haiyan Huang. "Gene coexpression measures in large heterogeneous samples using count statistics". In: *Proceedings of the National Academy of Sciences* 111.46 (2014), pp. 16371–16376.

[105] YX Rachel Wang et al. "Generalized correlation measure using count statistics for gene expression data with ordered samples". In: *Bioinformatics* 34.4 (2017), pp. 617–624.

[106] John N Weinstein et al. "The cancer genome atlas pan-cancer analysis project". In: *Nature genetics* 45.10 (2013), p. 1113.

[107] Qing Wen et al. "A gene-signature progression approach to identifying candidate small-molecule cancer therapeutics with connectivity mapping". In: *BMC bioinformatics* 17.1 (2016), pp. 1–11.

[108] Anja Wilmes et al. "Application of integrated transcriptomic, proteomic and metabolomic profiling for the delineation of mechanisms of drug induced cell stress". In: *Journal of proteomics* 79 (2013), pp. 180–194.

[109] Aaron K Wong, Arjun Krishnan, and Olga G Troyanskaya. "GIANT 2.0: genome-scale integrated analysis of gene networks in tissues". In: *Nucleic acids research* 46.W1 (2018), W65–W70.

[110] Ke Xu et al. "Blood vessel tubulogenesis requires Rasip1 regulation of GTPase signaling". In: *Developmental cell* 20.4 (2011), pp. 526–539.

[111] Lun Yang and Pankaj Agarwal. "Systematic drug repositioning based on clinical side-effects". In: *PloS one* 6.12 (2011), e28025.

[112] Yuting Ye and Peter J. Bickel. *Binomial Mixture Model With U-shape Constraint*. 2021. arXiv: `2107.13756 [stat.ME]`.

[113] Guangchuang Yu et al. "clusterProfiler: an R package for comparing biological themes among gene clusters". In: *Omics: a journal of integrative biology* 16.5 (2012), pp. 284–287.

[114] Luke Zappia, Belinda Phipson, and Alicia Oshlack. "Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database". In: *PLoS computational biology* 14.6 (2018), e1006245.

[115] Haoyang Zeng et al. "Convolutional neural network architectures for predicting DNA–protein binding". In: *Bioinformatics* 32.12 (2016), pp. i121–i127.

[116] Bin Zhang and Steve Horvath. "A general framework for weighted gene co-expression network analysis". In: *Statistical applications in genetics and molecular biology* 4.1 (2005).

[117] Hui Zhang et al. "Integrated proteogenomic characterization of human high-grade serous ovarian cancer". In: *Cell* 166.3 (2016), pp. 755–765.

[118] Shu-Dong Zhang and Timothy W Gant. "A simple and robust method for connecting small-molecule drugs using gene-expression signatures". In: *BMC bioinformatics* 9.1 (2008), pp. 1–10.

Appendix A

Supplemental material for "Local rank-pattern based reverse correlation"

A.1 Gene-wise statistics theory

In order to prove Theorem 2 from the main text, we need to show the result holds for $\boldsymbol{Z} = \sigma(\boldsymbol{x})$, when $\boldsymbol{Z}$ is a uniform random permutation of $\{1, \ldots, n\}$. This follows from the assumption that $\boldsymbol{x}$ has an exchangeable distributions and $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent.

For notational simplicity in the proof, we define a new coordinate system. For any $i \in \{1, \ldots, n\}$, we consider a window of size $2l-1$ that concentrates on i, denoted as $\mathbf{W}_i$. When i lies in the middle of the sequence $\{1, \ldots, n\}$, the center of $\mathbf{W}_i$ is i. When i locates close to the two ends of the sequence, i.e., 1 and n, $\mathbf{W}_i$ can stretch outside the sequence. We rearrange $\mathbf{W}_i$ to follow the circular pattern, as defined in by equation (1) in the main text. Since $Z_1, \ldots, Z_n$ are identically distributed under the null hypothesis that no gene is affected by the drug, $\{Z_1, \ldots, Z_h\}$ (or $\{Z_{n-h+1}, Z_{n-h+2}, \ldots, Z_n\}$) are identically distributed as $\{Z_{t+1}, \ldots, Z_{t+h}\}$ for any $t \geq 1$, $\leq h \leq n$ and $t + h \leq n$. Therefore, we can specifically define the window as

$$\mathbf{W}_i = \begin{cases} [i-l+1, i+l-1] & \text{if } l \leq i \leq n-l+1 \\ [1, 2l-1] & \text{if } i < l \\ [n-2(l-1), n] & \text{if } i > n-l+1 \end{cases} \tag{A.1}$$

We mention that the definition of the window in (A.1) is different from that of (1). The former rearranges the part of the window, which stretches outside of the sequence $\{1, \ldots, n\}$, to the other end of the window, while the latter rearranges this part to the other end of the sequence ([Ye] Need a picture for illustration). However, the two definitions are equivalent in distribution. We use the former definition is to simplify the analysis.

Additionally, let the shift s in this new coordinate system be $s = (\mathbf{W}_i[2l-1] + \mathbf{W}_i[1])/2 - i$, which is the shift quantity of the position i form the center of the window $\mathbf{W}_i$. We consider

two types location-specific counts of the decreasing pairs in the window $\mathbf{W}_i$ that involves Z_i. The first one is an unweighted statistic:

$$V_{i,l,s} = \underbrace{\sum_{j=1}^{l-1-s} \mathbb{I}(Z_{i-j} > Z_i)}_{A_{i,l,s}} + \underbrace{\sum_{j=1}^{l-1+s} \mathbb{I}(Z_i > Z_{i+j})}_{B_{i,l,s}} \tag{A.2}$$

The second one is weighted by the Manhattan distance between the two locations in a decreasing pair:

$$V^{(w)}_{i,l,s} = \underbrace{\sum_{j=1}^{l-1-s} \mathbb{I}(Z_{i-j} > Z_i) \cdot w_j}_{A^{(w)}_{i,l,s}} + \underbrace{\sum_{j=1}^{l-1+s} \mathbb{I}(Z_i > Z_{i+j}) \cdot w_j}_{B^{(w)}_{i,l,s}}, \tag{A.3}$$

where $w_j = |l-1-j| + \mathbb{I}(j \leq l-1)$. Note that under the assumptions from Theorem 2, we have that $\boldsymbol{Z}$ is a uniform random permutation of $\{1, \ldots, n\}$. When that is true, we know $V^{(w)}_{i,l,s} = G_i(l)$, and thus we can prove Theorem 2 by showing that $V^{(w)}_{i,l,s}$ has the same properties.

To begin we provide properties of the un-weighted version, $V_{i,l,s}$. Note that $V_{i,l,s}$ is simply counting the number of decreasing pairs containing gene i without adding the additional weights that result from the sliding windows.

Theorem 3 *For the asymptotics property of $V_{i,l,s}$, it follows that*

1. *When $s = o(\sqrt{l})$, $V_{i,l,s}$ is asymptotically a Gaussian-like random variable.*
2. *When $s = O(l^\alpha)$ with $\alpha > \frac{1}{2}$, $V_{i,l,s}$ is asymptotically a uniform distribution random variable.*
3. *When $s = O(\sqrt{l})$, $V_{i,l,s}$ is asymptotically a random variable from a mixture of a Gaussian-like distribution and a uniform distribution.*

For the finite-sample property, we have

1. *If $s = 0$,*

$$\mathbb{P}(V_{i,l,s} - l + 1 \geq v) \leq \exp(-\frac{v^2}{l-1}) \cdot \mathbb{I}(v > 0) + \mathbb{I}(v \leq 0) \tag{A.4}$$

2. *If $s \neq 0$,*

$$\mathbb{P}(V_{i,l,s} - l + 1 \geq v) \leq (\frac{|s| - v}{2|s|})_+ + \frac{\sqrt{l-1}\pi}{4|s|} \left\{ erf\left(\frac{v + |s|}{\sqrt{l-1}}\right) - erf\left(\frac{(v-|s|)_+}{\sqrt{l-1}}\right) \right\}, \tag{A.5}$$

where $(\cdot)_+ = \max(0, \cdot)$, $erf(\cdot)$ *is the error function, i.e.,*

$$erf(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2) dt.$$

Remark 1 *A quick calculation shows that*

$$\begin{aligned} &\mathbb{E}V_{i,l,s} = l - 1 \\ &var(V_{i,l,s}) = \tfrac{s^2+l-1}{3}. \end{aligned}$$

From the variance expression, we can have some sense why the asymptotic distribution of $V_{i,l,s}$ *changes when the scale of* s *increases from* $o(\sqrt{l})$ *to* $o(l^\alpha)$ $(\alpha > \frac{1}{2})$*. The most difficult case to analyze is* $s = O(\sqrt{l})$*, where we can not easily determine how the two components of the mixture distribution interact. Another noteworthy thing is that we do no have an exact Gaussian distribution even in the asymptotics sense. Instead, we have a distribution that looks like the Gaussian distribution. This phenomenon occurs because any two indicators in* $V_{i,l,s}$ *are highly correlated (they all contain the* Z_i *term). Common tools like Stein's methods can no longer be used here.*

Theorem 4 *For the asymptotics property of* $V_{i,l,s}^{(w)}$*, it follows that*

1. *When* $s = o(l^{\frac{3}{4}})$*,* $V_{i,l,s}^{(w)}$ *is asymptotically a Gaussian-like random variable.*
2. *When* $s = O(l^\alpha)$ *with* $\alpha > \frac{3}{4}$*,* $V_{i,l,s}^{(w)}$ *is asymptotically a uniform distribution random variable.*
3. *When* $s = O(l^{\frac{3}{4}})$*,* $V_{i,l,s}^{(w)}$ *is asymptotically a random variable from a mixture of a Gaussian-like distribution and a uniform distribution.*

For the finite-sample property, define $\tilde{\sigma}^2 = \frac{l(l-1)(2l-1)}{24}$*. we have*

1. *If* $s = 0$*,*

$$\mathbb{P}(V_{i,l,s}^{(w)} - \frac{(l-1)^2 + l}{2} \geq v) \leq \exp(-\frac{v^2}{\tilde{\sigma}^2}) \cdot \mathbb{I}(v > 0) + \mathbb{I}(v \leq 0) \tag{A.6}$$

2. *If* $s \neq 0$*,*

$$\mathbb{P}(V_{i,l,s}^{(w)} - \frac{l(l-1)}{2} \geq v | Z_i = z) \leq \exp\left\{\frac{-(v - \eta(s,z))^2}{\tilde{\sigma}^2}\right\} \mathbb{I}(v \geq \eta(s,z)) + \mathbb{I}(v < \eta(s,z)), \tag{A.7}$$

where $\eta(s, z) = \frac{1}{2}(2z - 1)(s^2 + |s|)$*.*

Proof of Theorem 3

Lemma 5 *Suppose* $\tilde{Z}_i \overset{i.i.d}{\sim} Uniform(0,1)$, $i = 1, \ldots, n$. *Follow all the definitions above for* $\tilde{V}_{i,l,s}$ *and* $\tilde{V}^{(w)}_{i,l,s}$ *using* $\tilde{Z}'_i s$. *Then*

$$\begin{aligned} \tilde{V}_{i,l,s} &\overset{d}{=} V_{i,l,s}. \\ \tilde{V}^{(w)}_{i,l,s} &\overset{d}{=} V^{(w)}_{i,l,s}. \end{aligned}$$

This is called Renyi representation.

Proof By Lemma 5, we can assume $Z_i \overset{i.i.d}{\sim} Uniform(0,1)$ without loss of generality. Conditional on $Z_i = z$, it is easy to see that

$$\left.\begin{array}{l} \mathbb{I}(Z_{i-j} > Z_i) \overset{i.i.d}{\sim} Bernoulli(1-z) \\ \mathbb{I}(Z_i > Z_{i+j}) \overset{i.i.d}{\sim} Bernoulli(z) \end{array}\right\} \perp\!\!\!\perp,$$

where $\perp\!\!\!\perp$ represents the independence. By Central Limite Theorem, it follows that

$$\begin{array}{l} \frac{A_{i,l,s}-(l-1-s)(1-z)}{\sqrt{l-1-s}} \xrightarrow{d} N(0, z(1-z)) \text{ as } l-1-s \to \infty \\ \frac{B_{i,l,s}-(l-1+s)z}{\sqrt{l-1+s}} \xrightarrow{d} N(0, z(1-z)) \text{ as } l-1-s \to \infty \end{array}.$$

In other words, we have

$$\left.\begin{array}{l} A_{i,l,s} \approx (l-1-s)(1-z) + \sqrt{(l-1-s)z(1-z)} \cdot W_1 \\ B_{i,l,s} \approx (l-1+s)z + \sqrt{(l-1-s)z(1-z)} \cdot W_2 \end{array}\right\} \perp\!\!\!\perp,$$

where $W_1 \perp\!\!\!\perp W_2$ are two standard Gaussian random variables. It implies that

$$V_{i,l,s} = A_{i,l,s} + B_{i,l,s} \approx l - 1 + s(2z-1) + \sqrt{(l-1) \cdot 2z(1-z)} W_3,$$

where W_3 is a standard Gaussian random variable.

1. If $s = o(\sqrt{l})$, $V_{i,l,s} \approx l - 1 + \sqrt{(l-1) \cdot 2z(1-z)} \cdot W_3$. Then consider the characteristic function of normalized $V_{i,l,s}$ and integrate over $z \in [0,1]$, we have

$$\begin{aligned} f(t) &= \mathbb{E} \exp\left(\mathrm{i}t \frac{V_i - (l-1)}{\sqrt{(l-1)/3}} \right) \to \int_0^1 \exp(-3(1-z)zt^2) dz \\ &= \frac{2\sqrt{3}}{3t} \exp\left(-\frac{3t^2}{4}\right) \int_0^{\frac{\sqrt{3}t}{2}} \exp(v^2) dv \\ &= 1 - \frac{t^2}{2} + \frac{t^4}{20/3} + O(t^6), \end{aligned}$$

where we normalize $V_{i,l,s}$ by substracting its mean $(l-1)$ and dividing $\sqrt{\frac{l-1}{3}}$. On the other hand, the characteristics function of the standard normal is

$$\mathbb{E}\exp\left(\mathrm{i}tZ\right) = \exp\left(-t^2/2\right) = 1 - \frac{t^2}{2} + \frac{t^4}{8} + O(t^6).$$

We can see that the asymptotic distribution of the normalized $V_{i,l,s}$ is close to the standard normal, but not exactly the same.

2. If $s = O(l^\alpha)$ with $\alpha > \frac{1}{2}$, then

$$V_{i,l,s} \approx l - 1 + s(2z-1),$$

where $\sqrt{l-1}N(0, 2z(1-z))$ can be ignored compared to the scale of s. Specifically, we have

$$\frac{V_{i,l,s} - l + 1}{s} \overset{d}{=} 2Z_i - 1 \sim Uniform(-1,1), \text{ a.s. } l \to \infty.$$

3. If $s = O(\sqrt{l})$, then neither $s(2z-1)$ nor $\sqrt{l-1}N(0, z(1-z))$ can be ignored. In this case, the normalied $V_{i,l,s}$ is a mixture of the Gaussian-like distribution random variable and the uniform distribution random variable, i.e.,

$$\frac{V_{i,l,s} - l + 1}{\sqrt{s^2 + l - 1}} \approx \frac{s(2z-1)}{\sqrt{s^2+l-1}} + \frac{\sqrt{(l-1)\cdot 2z(1-z)}\cdot W_3}{\sqrt{s^2+l-1}}.$$

Note that the sum of the weights on the two components is not one. This is because the two components are correlated.

Finally, we give the conservative p-values for $V_{i,l,s}$. Since all the indicators are upper bounded by 1 and lower bounded by 0, by using Hoeffding bound, we can easily obtain that

$$\mathbb{P}(V_{i,l,s} - (l-1) \geq v | Z_i = z) \leq \exp\left(-\frac{(2sz - v - s)^2}{l-1}\right) \cdot \mathbb{I}(v > s(2z-1)) + \mathbb{I}(v \leq s(2z-1))$$

1. If $s = 0$, the unconditional p-value is

$$\begin{aligned}
\mathbb{P}(V_{i,l,s} - (l-1) \geq v) &= \int_0^1 \mathbb{P}(V_{i,l,s} - l + 1 \geq v | Z_i = z) \cdot 1 dz \\
&\leq \mathbb{I}(v \leq 0) + \mathbb{I}(v > 0) \cdot \int_0^1 \exp\left(-\frac{v^2}{l-1}\right) dz \\
&= \exp(-v^2/(l-1)) \cdot \mathbb{I}(v > 0) + \mathbb{I}(v \leq 0)
\end{aligned}$$

2. If $s > 0$, we have

$$\begin{aligned}
\mathbb{P}(V_{i,l,s} - l + 1 \geq v) &= \int_0^1 \mathbb{P}(V_{i,l,s} - l + 1 \geq v | Z_i = z) dz \\
&\leq \int_0^{\min(\frac{v+s}{2s},1)} \exp\left(-\frac{(2sz - v - s)^2}{l-1}\right) dz + 1 - \min(\frac{v+s}{2s}, 1) \\
&= (\frac{-v+|s|}{2|s|})_+ + \frac{\sqrt{(l-1)\pi}}{4|s|}\left\{erf\left(\frac{v+|s|}{\sqrt{l-1}}\right) - erf\left(\frac{(v-|s|)_+}{\sqrt{l-1}}\right)\right\}.
\end{aligned}$$

3. If $s < 0$, we can prove the same result as $s > 0$ with a similar argument.

■

Proof of Theorem 4

Proof Given Lemma 5, we refer $V_{i,l,s}^{(w)}$ to $\tilde{V}_{i,l,s}^{(w)}$ in the following proof. It is easy to show that

$$\begin{aligned}
\mathbb{E}V_{i,l,s}^{(w)} &= \frac{1}{2}[(l-1)^2 + l - 1] \\
\mathbb{E}[V_{i,l,s}^{(w)} | Z_i = z] &= \frac{1}{2}[((l-1)^2 + (l-1)) + (2z-1)(s^2 + |s|)] \\
var[V_{i,l,s}^{(w)}] &= \frac{1}{18} \cdot l(l-1)(2l-1) + \frac{1}{12} \cdot (s^2 + |s|) \\
var[V_{i,l,s}^{(w)} | Z_i = z] &= \frac{1}{3} \cdot z(1-z) \cdot l(l-1)(2l-1)
\end{aligned}$$

By Linderberg CLT, we have

$$\frac{V_{i,l,s}^{(w)} - \frac{1}{2}[((l-1)^2 + l - 1) + (2z-1)(s^2 + |s|)]}{\sqrt{\frac{1}{3} \cdot l(l-1)(2l-1)}} \xrightarrow{d} N(0, z(1-z)) \qquad \text{given } Z_i = z.$$

Therefore,

$$V_{i,l,s}^{(w)} \approx \frac{1}{2}((l-1)^2 + l - 1) + \frac{1}{2}(2z-1)(s^2 + |s|) + \sqrt{\frac{1}{3} \cdot l(l-1)(2l-1)} \cdot N(0, z(1-z)).$$

1. When $s = o(l^{\frac{3}{4}})$, we can ignore the term $\frac{1}{2}(2z-1)(s^2 + |s|)$, and have

$$V_{i,l,s}^{(w)} \approx \frac{1}{2}((l-1)^2 + l - 1) + \sqrt{\frac{1}{3} \cdot l(l-1)(2l-1)} \cdot N(0, z(1-z)).$$

   As analyzed in Theorem 3, $3\sqrt{2} \cdot [V_{i,l,s}^{(w)}) - \frac{1}{2}((l-1)^2 + l - 1)]/\sqrt{l(l-1)(2l-1)}$ is distributed as a Gaussian-like distribution.

2. When $s = O(l^\alpha)$ with $\alpha > \frac{3}{4}$, then

$$V_{i,l,s}^{(w)} \approx \frac{1}{2}((l-1)^2 + l) + \frac{1}{2}(2z-1)(s^2+|s|).$$

In this case, $V_{i,l,s}^{(w)}$ is uniformly distributed as shown in Theorem 3.

3. When $s = O(l^{\frac{3}{4}})$, either $\frac{1}{2}(2z-1)(s^2+|s|)$ or $\sqrt{\frac{1}{3}l(l-1)(2l-1)} \cdot N(0, z(1-z))$ can be ignored. So the normalized $V_{i,l,s}^{(w)}$ is a mixture of Gaussian-like distribution and a uniform distribution.

Next, we consider the conservative p-value of $V_{i,l,s}^{(w)}$. Define $2\tilde{\sigma}^2 = \sum_{j=1}^{l-1-s} w_j^2 + \sum_{j=1}^{l-1+s} w_j^2 = \frac{(l-1)l(2l-1)}{3}$. Then by Hoeffding bound, we have

$$\mathbb{P}(V_{i,l,s}^{(w)} - \frac{l(l-1)}{2} \geq v | Z_i = z) \quad \leq \quad \exp\left\{\frac{-(v-\eta(s,z))^2}{\tilde{\sigma}^2}\right\} \mathbb{I}(v \geq \eta(s,z)) + \mathbb{I}(v < \eta(s,z)),$$

where $\eta(s,z) = \frac{1}{2}(2z-1)(s^2+|s|)$.

1. If $s = 0$, it follows that

$$\begin{aligned}
\mathbb{P}(V_{i,l,s}^{(w)} - \frac{l(l-1)}{2} \geq v) &= \int_0^1 \mathbb{P}(V_{i,l,s}^{(w)} - \frac{l(l-1)}{2} \geq v | Z_i = z) \cdot 1 dz \\
&\leq \mathbb{I}(v \leq 0) + \mathbb{I}(v > 0) \cdot \int_0^1 \exp\left\{-\frac{v^2}{\tilde{\sigma}^2} dz\right\} \\
&= \mathbb{I}(v \leq 0) + \mathbb{I}(v > 0) \cdot \exp\left\{-\frac{v^2}{\tilde{\sigma}^2} dz\right\} \qquad \text{(A.8)}
\end{aligned}$$

2. If $s > 0$, it follows that

$$\begin{aligned}
& \mathbb{P}(V_{i,l,s}^{(w)} - \frac{l(l-1)}{2} \geq v) \\
= & \int_0^1 \mathbb{P}(V_{i,l,s}^{(w)} - \frac{l(l-1)}{2} \geq v | Z_i = z) dz \\
\leq & \int_{(v-\frac{1}{2}(s^2+|s|))_+}^{v+\frac{1}{2}(s^2+|s|)} \frac{\exp\{-t^2/\tilde{\sigma}^2\}}{s^2+|s|} + (\frac{1}{2} - \frac{v}{s^2+|s|})_+ \\
= & (\frac{1}{2} - \frac{v}{s^2+|s|})_+ + \frac{\sqrt{\pi} \cdot \tilde{\sigma}}{2(s^2+|s|)} \left\{ erf\left(\frac{v + \frac{1}{2}(s^2+|s|)}{\tilde{\sigma}}\right) - erf\left(\frac{(v - \frac{1}{2}(s^2+|s|))_+}{\tilde{\sigma}}\right)\right\}.
\end{aligned}$$

3. If $s < 0$, we can prove the same result as $s > 0$ with a similar argument.

■

A.2 Top compounds

Rank	Perturbagen name	sLoCor	Rank	Perturbagen name	sLoCor
1	BRD-K64040351	5.78	16	BRD-K26772093	4.81
2	BRD-K26359746	5.76	17	BRD-K95716640	4.80
3	BRD-K12126940	5.67	18	SA-85148	4.78
4	SA-103150	5.48	19	BRD-K73008154	4.77
5	SA-1459172	5.35	20	BRD-K62014585	4.77
6	BRD-M79326715	5.32	21	BRD-K88819433	4.74
7	BRD-A72837804	5.17	22	BRD-K24427624	4.73
8	BRD-K73205140	5.12	23	SA-419090	4.69
9	BRD-K57393707	5.08	24	BRD-K11424789	4.68
10	BRD-K64687628	5.00	25	BRD-K43048602	4.66
11	BRD-K33486590	4.98	26	BRD-K31439714	4.64
12	BRD-K36591038	4.97	27	BRD-K14704318	4.63
13	BRD-K95521279	4.87	28	BRD-K96270963	4.62
14	BRD-K55681368	4.85	29	BRD-K47664103	4.61
15	BRD-K60796852	4.82	30	BRD-K22862802	4.60

Table A.1: Top perturbagens based on sLoCor for BRCA.

Rank	Perturbagen name	sLoCor	$\log(IC_{50})$
75	camptothecin	4.19	2.32
109	PHA-793887	4.00	3.11
231	entinostat	3.64	3.63
239	topotecan	3.62	2.66
247	alvocidib	3.61	1.85
291	SN-38	3.54	2.27
316	mitoxantrone	3.50	1.26
347	doxorubicin	3.46	2.48
370	raloxifene	3.42	-0.10
404	MG-132	3.40	2.60
477	trichostatin-a	3.31	1.78
514	gemcitabine	3.26	2.17
567	amonafide	3.19	3.57
722	amsacrine	3.02	2.34
737	vorinostat	3.01	3.18

Table A.2: Top perturbagens with IC_{50} based on sLoCor for BRCA.

Rank	Perturbagen name	sLoCor	Rank	Perturbagen name	sLoCor
1	BRD-K30025108	4.89	16	BRD-K74782274	3.71
2	BRD-K56366698	4.48	17	BRD-K29582182	3.71
3	BRD-K69681876	4.34	18	BRD-K82483257	3.62
4	BRD-K54711075	4.25	19	BRD-K30373477	3.62
5	BRD-K32014638	4.18	20	BRD-K41761879	3.61
6	BRD-K19898724	4.13	21	BRD-K07689498	3.59
7	BRD-K56647759	4.07	22	BRD-K46999092	3.56
8	BRD-K03608142	4	23	BRD-K68730107	3.52
9	BRD-K72827150	3.97	24	BRD-K40695780	3.51
10	BRD-K42624823	3.89	25	BRD-K90004840	3.50
11	BRD-K13234528	3.89	26	ropivacaine	3.46
12	BRD-K23077259	3.87	27	BRD-K15916856	3.45
13	BRD-K64716493	3.83	28	BRD-K55288106	3.44
14	BRD-K70125131	3.81	29	BRD-K67485291	3.44
15	BRD-K74594501	3.73	30	BRD-K71452159	3.44

Table A.3: Top perturbagens based on sLoCor for COAD.

sLoCor rank	Perturbagen name	sLoCor	$\log(IC_{50})$
152	dactinomycin	2.76	0.78
168	triptolide	2.71	1.19
190	doxorubicin	2.65	2.36
282	amonafide	2.48	3.67
303	daunorubicin	2.45	2.43
336	mitoxantrone	2.40	1.29
389	vorinostat	2.33	3.59
710	topotecan	2.04	2.57
726	bosutinib	2.03	3.65
789	camptothecin	1.98	2.41
906	etoposide	1.89	3.28
949	PD-184352	1.86	2.12
1220	NVP-TAE684	1.74	3.68
1436	gemcitabine	1.65	1.00
1484	ellipticine	1.63	4.17

Table A.4: Top perturbagens with IC_{50} based on sLoCor for COAD.

Rank	Perturbagen name	sLoCor	Rank	Perturbagen name	sLoCor
1	BRD-K91617567	5.29	16	BRD-K86638751	4.19
2	BRD-K66383562	5.15	17	SA-1973507	4.19
3	BRD-A14014306	4.81	18	BRD-K60019754	4.14
4	BRD-K55851396	4.64	19	BRD-K67035256	4.13
5	BRD-K46088987	4.46	20	BRD-K28934562	4.13
6	BRD-K09701578	4.45	21	BRD-K67774729	4.09
7	BRD-K94034047	4.44	22	BRD-K71701252	4.08
8	BRD-K81164606	4.43	23	BRD-K92627732	4.05
9	BRD-K51674646	4.42	24	BRD-K47533918	4.04
10	malonoben	4.38	25	BRD-K73008154	4.04
11	BRD-K55082668	4.36	26	BRD-K32581454	4.02
12	BRD-K47977047	4.34	27	BRD-K07339740	4.01
13	tyrphostin-A9	4.31	28	VU-0418939-2	4.00
14	BRD-K03652504	4.30	29	VU-0418946-1	3.99
15	BRD-K30025108	4.23	30	BRD-K23319301	3.97

Table A.5: Top perturbagens based on sLoCor for LIHC.

Rank	Perturbagen name	sLoCor	$\log(\mathrm{IC}_{50})$
193	camptothecin	3.12	2.22
199	topotecan	3.10	2.92
216	ryuvidine	3.08	0.08
311	brazilin	2.88	4.08
422	etoposide	2.72	3.69
438	amonafide	2.70	3.59
647	entinostat	2.51	4.48
685	MG-132	2.47	2.51
751	doxorubicin	2.42	2.70
753	bortezomib	2.42	1.40
812	cycloheximide	2.36	3.40
1250	vorinostat	2.11	3.22
1264	daunorubicin	2.11	2.85
1372	mercaptopurine	2.05	3.90
1453	bufalin	2.01	2.81

Table A.6: Top perturbagens with IC_{50} based on sLoCor for LIHC.

A.3 Top genes

Gene name	LoCor gene score	Achilles median effect
CCNA2	1.44	-1.40
CDC20	1.25	-1.66
PLK1	1.19	-1.63
CCNB2	1.16	0.03
AURKA	1.11	-1.24
KIF20A	1.07	-0.59
SMC4	1.01	-1.23
TOP2A	0.96	-1.65
CCNF	0.92	-0.34
LYRM1	0.91	0.11
KIF2C	0.91	-0.48
UBE2C	0.88	-0.54
CCNB1	0.82	-0.81
BIRC5	0.81	-1.41
POLE2	0.81	-1.64

Table A.7: Top genes for BRCA.

Gene name	LoCor gene score	Achilles median effect
MYC	1.13	-1.79
PSMG1	0.95	-0.85
DNMT1	0.92	-0.49
TSEN2	0.89	-0.85
EPHA3	0.88	-0.20
CCDC85B	0.85	-0.30
TCEA2	0.85	-0.07
CTSD	0.84	0.19
CSNK1E	0.82	-0.01
TM9SF3	0.80	-0.04
STUB1	0.79	-0.29
CDC25B	0.79	-0.31
SLC2A6	0.79	-0.23
GRB10	0.79	0.06
RPL39L	0.79	0.08

Table A.8: Top genes for COAD.

Gene name	LoCor gene score	Achilles median effect
CCNA2	1.35	-1.39
GADD45A	1.30	0.10
PLK1	1.11	-1.72
AURKA	1.10	-1.09
CCNF	1.09	-0.18
CDC20	1.05	-1.52
KIF20A	1.02	-0.42
KIF14	0.99	-0.55
TOP2A	0.95	-1.61
CCNB2	0.92	0.09
WDR7	0.86	-1.10
JUN	0.82	-0.27
GADD45B	0.80	-0.25
PSRC1	0.77	-0.28
MCM3	0.77	-0.84

Table A.9: Top genes for LIHC.

Appendix B

Supplemental material for "GeneFishing in scRNA-sequencing data and its applications"

B.1 Enrichment analysis results

Tables in this section were created using `stargazer` [39]. All enrichment results are from `clusterProfiler` [113] for KEGG pathway enrichment and `topGO` [3] for GO term enrichement.

Liver - cholesterol metabolic process bait

Bait set 1

Using the liver with 14 bait genes selected from the cholesterol metabolic process GO term we fished out 453 additional genes using a CFR cutoff of 0.969. Table B.1 and B.2 show the pathways and GO terms that are enriched for these 453 genes compared to the other genes in the data.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu04610	Complement and coagulation cascades	34/279	61/3447	0
mmu00830	Retinol metabolism	26/279	55/3447	0
mmu00140	Steroid hormone biosynthesis	24/279	48/3447	0
mmu04976	Bile secretion	24/279	52/3447	0
mmu00591	Linoleic acid metabolism	11/279	19/3447	0.00000
mmu05204	Chemical carcinogenesis - DNA adducts	18/279	52/3447	0.00000
mmu04141	Protein processing in endoplasmic reticulum	28/279	130/3447	0.00003
mmu03320	PPAR signaling pathway	17/279	59/3447	0.0001
mmu02010	ABC transporters	11/279	27/3447	0.0001
mmu04979	Cholesterol metabolism	13/279	39/3447	0.0001

Table B.1: 10 most enriched KEGG pathways for GeneFishing with 14 cholesterol metabolic process GO term genes in the liver data.

GO.ID	Term	Significant	Annotated	p.value
GO:0042738	exogenous drug catabolic process	16	31	3.0e-12
GO:0010951	negative regulation of endopeptidase act...	20	81	1.4e-10
GO:0019373	epoxygenase P450 pathway	12	21	3.5e-10
GO:0006805	xenobiotic metabolic process	19	62	7.9e-09
GO:0006958	complement activation, classical pathway	11	23	2.3e-08
GO:0007596	blood coagulation	31	74	1.9e-07
GO:0072378	blood coagulation, fibrin clot formation	6	7	3.1e-07
GO:0055114	oxidation-reduction process	66	528	7.7e-07
GO:0051918	negative regulation of fibrinolysis	5	5	7.8e-07
GO:0015722	canalicular bile acid transport	5	5	7.8e-07

Table B.2: 10 most enriched GO terms for GeneFishing with 14 cholesterol metabolic process GO term genes in the liver data.

Bait set 2

We found an additional bait set of 11 cholesterol metabolic process genes, where we fished out 532 additional genes with a CFR cutoff of 0.951. The enrichment results are very similar to the results above because there are 413 genes in common between the two GeneFishing results and we recover 13 of the 14 bait genes from the first bait set.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu04610	Complement and coagulation cascades	31/306	61/3447	0
mmu00140	Steroid hormone biosynthesis	26/306	47/3447	0
mmu00830	Retinol metabolism	27/306	55/3447	0
mmu04976	Bile secretion	22/306	52/3447	0
mmu00591	Linoleic acid metabolism	12/306	19/3447	0.00000
mmu05204	Chemical carcinogenesis - DNA adducts	18/306	52/3447	0.00001
mmu03320	PPAR signaling pathway	19/306	60/3447	0.00002
mmu02010	ABC transporters	12/306	27/3447	0.00003
mmu04931	Insulin resistance	19/306	65/3447	0.00004
mmu04141	Protein processing in endoplasmic reticulum	29/306	130/3447	0.00004

Table B.3: 10 most enriched KEGG pathways for GeneFishing with 11 cholesterol metabolic process GO term genes in the liver data.

GO.ID	Term	Significant	Annotated	p.value
GO:0042738	exogenous drug catabolic process	16	31	3.3e-11
GO:0010951	negative regulation of endopeptidase act...	21	82	1.4e-09
GO:0019373	epoxygenase P450 pathway	12	21	2.2e-09
GO:0055085	transmembrane transport	71	473	1.5e-08
GO:0001676	long-chain fatty acid metabolic process	30	68	4.2e-08
GO:0007596	blood coagulation	30	74	5.2e-08
GO:0006805	xenobiotic metabolic process	20	62	7.2e-08
GO:0042632	cholesterol homeostasis	18	64	2.3e-07
GO:0006641	triglyceride metabolic process	25	74	4.7e-07
GO:0072378	blood coagulation, fibrin clot formation	6	7	7.8e-07

Table B.4: 10 most enriched GO terms for GeneFishing with 11 cholesterol metabolic process GO term genes in the liver data.

Pancreas - insulin secretion bait

Using all pancreas cells with 12 bait genes selected from the insulin secretion GO term we fished out 158 additional genes using a CFR cutoff of 0.912. Table B.5 and B.6 show the pathways and GO terms that are enriched for these 158 genes compared to the other genes in the data.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu04950	Maturity onset diabetes of the young	5/72	15/4304	0.001
mmu04917	Prolactin signaling pathway	6/72	50/4304	0.013
mmu04080	Neuroactive ligand-receptor interaction	10/72	163/4304	0.018
mmu04960	Aldosterone-regulated sodium reabsorption	4/72	26/4304	0.034

Table B.5: KEGG pathway enrichment for GeneFishing in all 1,419 pancreas cells.

GO.ID	Term	Significant	Annotated	p.value
GO:0042593	glucose homeostasis	9	166	4.5e-05
GO:0007601	visual perception	7	73	0.00011
GO:0001973	G protein-coupled adenosine receptor sig...	3	7	0.00011
GO:1904659	glucose transmembrane transport	4	75	0.00131
GO:0007605	sensory perception of sound	6	85	0.00174
GO:0002678	positive regulation of chronic inflammat...	2	5	0.00219
GO:0060384	innervation	3	19	0.00272
GO:0035556	intracellular signal transduction	34	1,471	0.00300
GO:0048149	behavioral response to ethanol	2	6	0.00325
GO:0048312	intracellular distribution of mitochondr...	2	6	0.00325

Table B.6: 10 most enriched GO terms for GeneFishing in all 1,419 pancreas cells.

Heart and aorta - antigen presentation bait

Using the heart and aorta with 11 bait genes selected from the antigen presentation GO term we fished out 337 additional genes using a CFR cutoff of 0.933. Table B.9 and B.10 show the pathways and GO terms that are enriched for these 337 genes compared to the other genes in the data.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu04613	Neutrophil extracellular trap formation	20/166	71/3824	0
mmu04380	Osteoclast differentiation	22/166	91/3824	0
mmu05150	Staphylococcus aureus infection	13/166	28/3824	0
mmu04650	Natural killer cell mediated cytotoxicity	19/166	74/3824	0
mmu04662	B cell receptor signaling pathway	17/166	61/3824	0
mmu04062	Chemokine signaling pathway	24/166	126/3824	0
mmu05140	Leishmaniasis	15/166	50/3824	0.00000
mmu05152	Tuberculosis	21/166	106/3824	0.00000
mmu04666	Fc gamma R-mediated phagocytosis	15/166	70/3824	0.00000
mmu04670	Leukocyte transendothelial migration	16/166	83/3824	0.00001

Table B.7: 10 most enriched KEGG pathways for GeneFishing with 11 antigen presentation GO term genes in the heart and aorta data.

GO.ID	Term	Significant	Annotated	p.value
GO:0045087	innate immune response	61	374	2.2e-17
GO:0002250	adaptive immune response	46	240	3.0e-10
GO:0006954	inflammatory response	68	413	3.3e-10
GO:0030593	neutrophil chemotaxis	18	67	1.6e-09
GO:0006911	phagocytosis, engulfment	17	44	6.3e-09
GO:0001774	microglial cell activation	14	34	6.8e-09
GO:0032715	negative regulation of interleukin-6 pro...	10	29	4.5e-08
GO:0032755	positive regulation of interleukin-6 pro...	13	55	6.8e-08
GO:0070374	positive regulation of ERK1 and ERK2 cas...	19	122	9.3e-08
GO:0032760	positive regulation of tumor necrosis fa...	11	41	1.7e-07

Table B.8: 10 most enriched GO terms for GeneFishing with 11 antigen presentation GO term genes in the heart and aorta data.

Heart and aorta - cardiac muscle cell contraction bait

Using the heart and aorta with 30 bait genes selected from the cardiac muscle cell contraction GO term we fished out 583 additional genes using a CFR cutoff of 0.931. Table B.9 and B.10 show the pathways and GO terms that are enriched for these 583 genes compared to the other genes in the data.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu05415	Diabetic cardiomyopathy	37/266	154/3809	0
mmu01200	Carbon metabolism	24/266	72/3809	0
mmu00020	Citrate cycle (TCA cycle)	14/266	25/3809	0
mmu04714	Thermogenesis	34/266	152/3809	0.00000
mmu04260	Cardiac muscle contraction	18/266	46/3809	0.00000
mmu05412	Arrhythmogenic right ventricular cardiomyopathy	16/266	46/3809	0.00000
mmu04020	Calcium signaling pathway	25/266	109/3809	0.00000
mmu00190	Oxidative phosphorylation	23/266	99/3809	0.00000
mmu05012	Parkinson disease	33/266	189/3809	0.00001
mmu01210	2-Oxocarboxylic acid metabolism	7/266	10/3809	0.00002

Table B.9: 10 most enriched KEGG pathways for GeneFishing with 30 cardiac muscle cell contraction GO term genes in the heart and aorta data.

GO.ID	Term	Significant	Annotated	p.value
GO:0006099	tricarboxylic acid cycle	13	25	6.7e-10
GO:0045214	sarcomere organization	15	34	1.2e-07
GO:0006103	2-oxoglutarate metabolic process	8	13	2.6e-07
GO:0014874	response to stimulus involved in regulat...	6	7	4.4e-07
GO:0055003	cardiac myofibril assembly	8	14	5.7e-07
GO:0055013	cardiac muscle cell development	29	69	8.7e-07
GO:0010880	regulation of release of sequestered cal...	6	8	1.7e-06
GO:0002027	regulation of heart rate	17	45	2.0e-06
GO:0048747	muscle fiber development	13	38	2.1e-06
GO:1903779	regulation of cardiac conduction	7	12	2.5e-06

Table B.10: 10 most enriched GO terms for GeneFishing with 30 cardiac muscle cell contraction GO term genes in the heart and aorta data.

Heart and aorta - endothelial cell proliferation bait

Using the heart and aorta with 21 bait genes selected from the endothelial cell proliferation GO term we fished out 359 additional genes using a CFR cutoff of 0.971. Table B.11 and B.12 show the pathways and GO terms that are enriched for these 359 genes compared to the other genes in the data.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu04015	Rap1 signaling pathway	17/131	128/3821	0.0002
mmu04072	Phospholipase D signaling pathway	12/131	85/3821	0.003
mmu04360	Axon guidance	12/131	109/3821	0.020
mmu04670	Leukocyte transendothelial migration	10/131	83/3821	0.022
mmu04014	Ras signaling pathway	13/131	133/3821	0.022
mmu04926	Relaxin signaling pathway	10/131	87/3821	0.022
mmu04923	Regulation of lipolysis in adipocytes	6/131	33/3821	0.022
mmu05200	Pathways in cancer	22/131	329/3821	0.040
mmu04514	Cell adhesion molecules	9/131	85/3821	0.050
mmu04724	Glutamatergic synapse	7/131	55/3821	0.050

Table B.11: 10 most enriched KEGG pathways for GeneFishing with 21 endothelial cell proliferation GO term genes in the heart and aorta data.

GO.ID	Term	Significant	Annotated	p.value
GO:0001525	angiogenesis	47	335	2.0e-06
GO:0061028	establishment of endothelial barrier	10	36	2.1e-05
GO:1905653	positive regulation of artery morphogene...	4	6	3.5e-05
GO:0048013	ephrin receptor signaling pathway	6	18	4.7e-05
GO:0043267	negative regulation of potassium ion tra...	5	23	7.8e-05
GO:1902396	protein localization to bicellular tight...	4	7	7.9e-05
GO:0048845	venous blood vessel morphogenesis	4	9	0.00027
GO:0045765	regulation of angiogenesis	28	204	0.00035
GO:0043114	regulation of vascular permeability	9	29	0.00056
GO:0003348	cardiac endothelial cell differentiation	3	5	0.00059

Table B.12: 10 most enriched GO terms for GeneFishing with 21 endothelial cell proliferation GO term genes in the heart and aorta data.

Heart and aorta - fibroblast proliferation bait

Using the heart and aorta with 8 bait genes selected from the fibroblast proliferation GO term we fished out 270 additional genes using a CFR cutoff of 0.972. Table B.13 and B.14 show the pathways and GO terms that are enriched for these 270 genes compared to the other genes in the data.

ID	Description	GeneRatio	BgRatio	p.adjust
mmu04974	Protein digestion and absorption	17/108	44/3829	0
mmu04512	ECM-receptor interaction	12/108	51/3829	0.00000
mmu04151	PI3K-Akt signaling pathway	18/108	207/3829	0.001
mmu04510	Focal adhesion	12/108	143/3829	0.020
mmu05146	Amoebiasis	8/108	69/3829	0.020
mmu05204	Chemical carcinogenesis - DNA adducts	5/108	26/3829	0.020
mmu00982	Drug metabolism - cytochrome P450	5/108	27/3829	0.021
mmu00980	Metabolism of xenobiotics by cytochrome P450	5/108	30/3829	0.028
mmu04933	AGE-RAGE signaling pathway in diabetic complications	8/108	78/3829	0.028
mmu05165	Human papillomavirus infection	14/108	211/3829	0.038

Table B.13: 10 most enriched KEGG pathways for GeneFishing with 8 fibroblast proliferation GO term genes in the heart and aorta data.

GO.ID	Term	Significant	Annotated	p.value
GO:0030199	collagen fibril organization	15	33	1.4e-14
GO:0030198	extracellular matrix organization	46	167	1.4e-12
GO:0007155	cell adhesion	58	759	3.0e-09
GO:0044849	estrous cycle	4	6	1.4e-05
GO:0048286	lung alveolus development	8	35	3.2e-05
GO:0048251	elastic fiber assembly	4	7	3.2e-05
GO:0001657	ureteric bud development	8	54	3.4e-05
GO:0002053	positive regulation of mesenchymal cell ...	6	22	4.7e-05
GO:0010575	positive regulation of vascular endothel...	5	15	7.2e-05
GO:0001568	blood vessel development	39	477	0.00013

Table B.14: 10 most enriched GO terms for GeneFishing with 8 fibroblast proliferation GO term genes in the heart and aorta data.

Appendix C

Supplemental material for "Using regional volumetric data to predict clinical progression in multiple sclerosis"

Region	Cluster number	Summary statistic	Coefficient	SE	p-value
Pars orbitalis	1	Std	-11	3.3	0.001
Rostral middle frontal	1	Skewness	-0.48	0.13	0.003
Fusiform	1	Skewness	0.61	0.21	0.004
3rd ventricle	2	Skewness	0.60	0.21	0.005
Cerebellum (exterior)	1	Skewness	0.63	0.23	0.006
Posterior cingulate	1	Skewness	0.97	0.36	0.007
Pars orbitalis	2	Std	-7.9	3.0	0.01
Superior frontal	4	Std	14	5.5	0.01
Rostral anterior cingulate	2	Std	6.0	2.4	0.01
Posterior cingulate	9	Skewness	-0.57	0.23	0.01
Rostral anterior cingulate	4	Mean	3.1	1.3	0.02
Superior temporal	1	Std	-11.7	4.9	0.02
Posterior cingulate	7	Skewness	0.84	0.40	0.02
Caudate	2	Std	10.6	4.7	0.03

Table C.1: Selected features for the data-driven ROIs. The cluster number is the number from the clustering. The numbers themselves are not meaningful, but they are included to distinguish the various clusters within a single original ROI label. For example, in the pars orbitalis the standard deviation from two clusters (1 and 2) are included in the selected feature set.

Region	Summary statistic	Coefficient	SE	p-value
Middle temporal	Std	-21	5.8	0.0002
Lingual	Mean	-23	6.4	0.0003
Pars orbitalis	Std	-12	3.3	0.0004
Cerebellum (exterior)	Mean	-14	4.3	0.001
Fusiform	Mean	-14	4.4	0.002
3rd ventricle	Skewness	0.7	0.24	0.003
Superior temporal	Std	-17	6.1	0.003
Cerebellar vermal lobules VI-VII	Mean	-4.7	1.9	0.01
4th ventricle	Mean	-4.6	1.9	0.01
Cerebellar vermal lobules VIII-X	Mean	-4.4	1.9	0.02
Rostral middle frontal	Std	-8.9	3.9	0.02

Table C.2: Selected features for the original ROIs.

Region	Cluster number	Summary statistic	Coefficient	SE	p-value
3rd ventricle	2	Skewness	094	0.30	0.002
Posterior cingulate	9	Skewness	-1.0	0.32	0.002
Rostral anterior cingulate	2	Skewness	1.7	0.58	0.004
Brain stem	1	Mean	-5.5	2.2	0.01
Pallidum	2	Skewness	1.1	.044	0.01
Posterior cingulate	2	Skewness	-0.90	0.36	0.01
Inferior temporal	2	Skewness	0.83	0.33	0.01
Amygdala	2	Mean	-6.6	2.8	0.02
Cerebellum (exterior)	2	Mean	-14	6.0	0.02
CSF	10	Std	25	11	0.02
4th ventricle	1	Mean	-4.3	2.0	0.02

Table C.3: Selected features for the data-driven ROIs in the female patients. Again, cluster numbers are included.

Region	Summary statistic	Coefficient	SE	p-value
3rd ventricle	Skewness	1.0	0.32	0.002
Cerebellum (exterior)	Mean	-14	6.4	0.03
Hippocampus	Std	-15	7.1	0.03
Amygdala	Mean	-6.7	3.2	0.04
Inferior temporal	Skewness	0.67	0.33	0.04
4th ventricle	Mean	-4.5	2.3	0.05

Table C.4: Selected features for the original ROIs in the female patients. Note, the cutoff of p-value for feature selection was increased from 0.025 to 0.05 as to include more than one DBM feature in the model.

Region	Cluster number	Summary statistic	Coefficient	SE	p-value
Pars orbitalis	1	Std	-25.95	7.34	0.0004
Basal forebrain	1	Skewness	1.47	0.46	0.002
Posterior cingulate	1	Skewness	2.24	0.72	0.002
Rostral middle frontal	1	Skewness	-0.77	0.26	0.003
Lateral orbitofrontal	2	Std	-10.01	3.37	0.004
Lateral orbitofrontal	3	Std	-10.66	3.68	0.004
Middle temporal	1	Std	-19.43	6.72	0.005
Posterior cingulate	7	Skewness	2.01	0.72	0.01
Pars orbitalis	2	Std	-14.15	5.30	0.01
Supramarginal	1	Skewness	1.25	0.47	0.01
Pars orbitalis	1	Skewness	-1.17	0.44	0.01
Rostral middle frontal	2	Skewness	-0.86	0.34	0.01
Superior frontal	4	Std	23.24	9.23	0.01
Posterior cingulate	4	Std	25.84	10.31	0.01
Rostral middle frontal	2	Std	-16.99	6.98	0.01
Pallidum	2	Skewness	-1.35	0.57	0.02
Fusiform	1	Std	-17.83	7.69	0.02
Inferior temporal	2	Std	-17.84	7.71	0.02
Posterior cingulate	7	Std	19.75	8.56	0.02
Postcentral	1	Std	13.32	5.84	0.02
Accumbens area	2	Skewness	0.96	0.43	0.02

Table C.5: Selected features for the data-driven ROIs in the male patients. Again, cluster numbers are included.

Region	Summary statistic	Coefficient	SE	p-value
pars.orbitalis	Std	-21.91	6.51	0.0008
Middle temporal	Std	-35.55	11.25	0.002
Lateral orbitofrontal	Mean	-13.50	4.30	0.002
Lingual	Mean	-37.80	12.76	0.003
Fusiform	Mean	-29.83	10.53	0.005
Lateral orbitofrontal	Std	-8.89	3.32	0.01
Isthmus cingulate	Mean	-23.94	8.98	0.01
Rostral middle frontal	Skewness	-0.79	0.30	0.01
Inferior temporal	Std	-23.93	9.33	0.01
Pars orbitalis	Mean	-16.99	6.64	0.01
Supramarginal	Skewness	1.42	0.56	0.01
Fusiform	Skewness	0.98	0.39	0.01
Parahippocampal	Mean	-14.94	6.07	0.01
Inferior temporal	Mean	-19.08	7.96	0.02
Basal forebrain	Skewness	1.17	0.49	0.02
Rostral middle frontal	Std	-16.86	7.18	0.02
Accumbens area	Skewness	1.32	0.56	0.02
Cerebellar vermal lobules I-V	Mean	-11.82	5.08	0.02
Putamen	Skewness	-1.28	0.56	0.02

Table C.6: Selected features for the original ROIs in the male patients.