Carnegie Mellon University
Dietrich College of Humanities and Social Sciences
Dissertation

Submitted in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy

Title: High-dimensional statistical methods to model heterogeneity in genomic data
Presented by: Kevin Lin
Accepted by: Department of Statistics & Data Science
Readers:

KATHRYN ROEDER, ADVISOR	DATE
JING LEI, ADVISOR	DATE
MAX G'SELL	DATE
HAN LIU	DATE
ANJALI MAZUMDER	DATE
RYAN J. TIBSHIRANI	DATE
LARRY WASSERMAN	DATE

Approved by the Committee on Graduate Degrees:

RICHARD SCHEINES, DEAN	DATE

High-dimensional statistical methods to model heterogeneity in genomic data

Kevin Lin

April 9, 2020

A dissertation submitted in partial fulfillment
of the requirements for the Degree of Doctor of Philosophy

Department of Statistics & Data Science
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213

Thesis Committee:
Kathryn Roeder, Chair
Jing Lei, Chair
Max G'Sell
Han Liu
Anjali Mazumder
Ryan J. Tibshirani
Larry Wasserman

To the future me, who will recall fond memories when reading this.

Abstract

Often in genomic studies, understanding the heterogeneity among the samples can be helpful to address scientific questions directly, as well as to better understand how to model the data in downstream analyses. As an example of the former, geneticists are interested in understanding which regions of genome of tumor cells are erroneously too long or too short when compared to their control cells counterparts – a phenomenon known as copy number variation (CNV). Geneticists deploy comparative genomic hybridization (CGH) methods to collect data, which are analyzed by changepoint methods to detect heterogeneity among segments of the genome to directly address this scientific question. As an example of the latter, single-cell RNA-sequencing (RNA-seq) data give geneticists new opportunities to understand how individual cells express different genes at different intensities. In these studies, capturing the heterogeneity among cells is often the first step for improved downstream analyses.

In this thesis, we design various high-dimensional statistical methods to address the types of heterogeneity often found in genomic data. We provide a high-level overview of genomics in the first chapter. In the second chapter, we develop a method to determine, among a collection of different microarray expression datasets, a large subset of datasets that have similar covariance matrices, which is applied in an analysis pipeline to help detect genes associated with autism spectrum disorder (ASD). In the third and fourth chapters, we develop theoretical understandings of changepoint detection methods and quantify their detected changepoints' statistical significance, which are applied to CGH data to infer which segments of the genome display copy number variation. In the fifth chapter, we develop a non-linear dimension reduction method based on matrix factorization for one-parameter exponential-family distributions and study its theoretical properties. Our method is applied to study the cell developmental trajectories of oligodendrocytes – a particular cell type that plays an important role in the central nervous system.

Acknowledgments

There are too many people to thank for my tremendous growth, both academically and personally, over the last six years of my PhD. Nonetheless, I can try thank as many people as I can, for I cannot express enough gratitude to all the wonderful individuals who have made this journey over the last six years possible. Below are people I would like to thank in no particular order.

- To Kathryn Roeder, who has helped me grow so much ever since I was a first year student. I have developed an unending interest in genomics through her, and have learned so much about how mold and craft a curious idea into a fully-flushed out research paper. She has truly been an academic parent to me in so many ways. She has seen and helped me grow from viewing genomics as "an application where there's just another data matrix to throw into some statistical method" to a wonderful and complex field with many biological questions that can slowly demystified with the help of careful statistical analyses and visualizations.

- To Jing Lei, who has taught me so much about how to dissect statistical theory. Often times, I have found myself lost in the mechanical details, struggling to follow or execute a proof line-by-line. However, with his help, I have learned how to see through the forest and understand the broader theoretical narratives at play.

- To Han Liu, who has nurtured my statistical fascination when I was an undergraduate. He has inspired my curiosity of jointly studying high-dimensional statistics and optimization ever since the start of my PhD journey, and I would come full-circle to revisit these thoughts at the end of my PhD journey. He has always encouraged and pushed me to work on topics that I felt passionate about, even when I lacked the self-confidence to commit to them myself.

- To Ryan Tibshirani, who has been, in many ways, a co-advisor in spirit due to the numerous projects I have worked with him on. I am forever thankful for his patience

and individualized teachings when I started working with him in my PhD, allowing me to cross such a chasm of knowledge in such a short amount of time.

- To Larry Wasserman, who has shaped so much of my statistical instincts through his courses at the start of my PhD. His door was always open for me to barge in and ask random statistical questions, and I continued many unplanned discussions with him throughout the PhD.

- To Max G'Sell and Anjali Mazumder, who both have offered many suggestions and advice, both in terms of research and career, to me over the course of my PhD while providing feedback for my thesis.

- To Valerie Ventura, Brian Junker, and Alex Reinhart, who have taught me so much about how to craft a presentation, structure a paper, and design a codebase respectively. In my younger years, I had (erroneously) thought that research was only about learning new techniques and applying it to interesting problems. However, through the teachings and conversations of these three individuals, I have realized that knowing how to do research and how to communicate the research are just as important as the research itself. From Valerie, I have learned about the importance of the Assertion-Evidence style of presenting. From Brian, I have learned that all papers are opportunities to teach the reader. From Alex, I have learned the importance of unit testing.

- To Aaditya Ramdas, Edward Kennedy, and Yuting Wei who have taught such wonderful special-topics courses that I had the great opportunity of sitting in. Not only were their materials fascinating, they also presented in such charismatic and effective ways that I hope to incorporate their teaching techniques into my own teachings in the future.

- To all the professors in the CMU Statistics and Data Science department as a whole, especially Sivaraman Balakrishnan, Peter Freeman, Christopher Genovese, Joel Greenhouse, Jiashun Jin, Ann Lee, Rebecca Nugent, and Alessandro Rinaldo for their continued support, encouragement and humor throughout the PhD years.

- To the CMU Statistics and Data Science department staff, especially Kira Bokalders, Laura Butler, Beth Dongili, Paige Houser, Christopher Peter Makris, Carl Skipper, and Margie Smykla, for working behind-the-scenes to ensure that I was able to focus on my research with all my attention. They were always incredibly considerate and helpful whenever I had questions about logistics.

- To Bernie Devlin, Bert Klei, and Maria Jalbrzikowski, as well as Tim Barry, Ercument Cicek, Eugene Katesvich, Yue Li, Fuchen Liu, Cong Lu, Klea Panayidou, Minshi Peng, Yixuan Qiu, Jinjin Tian, Jiebiao Wang, Xuran Wang, Lu Xie, Ron Yurko, and

Lingxue Zhu, for absolutely entertaining and educational experiences in the biweekly lab meetings about statistics and genomics. It was such a great pleasure to have drawn knowledge and insight from each of you as I listened to their presentations and discussions.

- To the Eberly Center, especially Maggie Goss, Olga Navros, and Nuria Ballesteros Soria, for its wonderful Future Faculty Program that has opened my eyes to teaching perspectives and techniques that I never previously thought about. These three individuals provided such stellar feedback in their teaching observations of me, and I hope to incorporate their suggestions to my teaching in the future.

- To Robert Vanderbei, Philippe Rigollet, David Blei, Warren Powell, Robert Schapire, and Moses Charikar, who were all incredibly influential in fostering my curiosity of statistics/optimization/machine learning when I was an undergraduate. I return to the comfort of their lecture notes more times than I would have expected. Many of them spent an extraordinary time conversing to me about what graduate school was like, and I hope to display just as much care towards undergraduates in the future.

- To Junwei Lu, Ethan Wang, and Michael Bino, who taught me so much in my last year as an undergraduate and the first few years of my PhD when I visited Princeton often. Their humor and knowledge helped me to adapt seamlessly to my graduate life.

- To Anjalie Field, Mandy Coston, Lisa Lee, Channing Huang, and Tiggy and Eevee who formed my home away from home. Through them, I have been able to experience so much of Pittsburgh, and always had a reliable group to invite over to cook for. Despite the fact that they are in different departments from me, I'm constantly fascinated by their research projects outside of statistics.

- To Ksenia Korovina, whom I had the absolute joy of being roommates with and cooking for. I shared many conversations with her about spanning from random research ideas to random stories of graduate school life.

- To Xiao Hui Tai, who has brought so much joy and laughter throughout my PhD years. Legend has it that anyone could know the presence of both of us together solely based on our combined laughter.

- To Justin Hyun and Daren Wang, who I had the pleasure of collaborating on various projects. While research is never as straightforward as one would hope, I have learned and experienced so much through the countless hours I spent with them.

- To Peter Elliot and Nick Kim, who together with Alex Reinhart, entertained my many random thoughts that I would have either right before my meetings when I

was adrenaline-rushed or right after my meetings when I was mentally exhausted and simply wanted something random to discuss about.

- To Natalia Oliveira, Riccardo Fogliato, Vanessa Vidal, Bishal Karki, Jacqueline Liu, and Shamindra Shrotriya, who helped me explore the fascinating world of food and drinks, even when these experiences were sandwiched between discussions related to statistics. Through them, I learned so much about cooking and other cuisines that I had never previously been exposed to.

- To all the PhD students in the CMU Statistics and Data Science department as a whole, who have taught and support me so often throughout my PhD years. To YJ Choe, Taylor Pospisil, Collin Eubanks, and Bryan Hooi for being great officemates to entertain my random and unpredictable thoughts and sometimes found me napping in the office. To Ilmun Kim, Jaehyok Shin, and Heejong Bong for their thought-provoking opinions about the landscape of statistical research. To Pratik Patil who accompanied me in many classes and ping-ponged many statistical curiosities with me. To Robert Lunde, Yen-chi Chen, and Jisu Kim for their patience in teaching me nuanced statistical theory that I struggled to learn on my own. To Purvasha Chakravarti for her persistence and patience to tackle statistical problems with me, even when they were not needed for her own work. To Yufei Yi and Maria Jahja for their neverending humor and sass that never failed to brighten my day. To Kayla Frisoli, Yotam Hechtlinger, Lee Richardson, Matteo Bonvini, Brendan McVeigh, Ben LeRoy, Nic Dalmasso, Jining Qin, Alan Mishler, and Ivy Gu for their humor and hot-takes about how to improve statistical education.

- To Tina Shiang and Carmen Khoo, who were both deeply influential in getting me to fall in love with Zumba. While I was very nervous and filled with anxiety at the time, I am extremely appreciative of their efforts to drag me to my first Zumba classes.

- To Michelle Vislay, Liz Carter, Lettia DeNormandie, Cassie Eng, and Lisha White, whom I had to absolute and complete joy of doing Zumba with every Monday and Tuesday for the last three years. The times spent with them made Mondays and Tuesdays always the two days I would look forward to every week, and I can only hope of this type of luxury in the future.

- To Cassie Hsu and An Chu, my two closest friends who have supported me unquestionably throughout my entire PhD years, despite having to do virtually, to ensure I was able to always bounce back from any slump.

- To Princeton eSports and its honorary family, especially Mona Zhang, April Hu, Dice Katsumata, Crystal Qian, Betty Liu, Jen Chew, Pat Park, Heling Zhao, Thomas Gilgenast, Sida Huang, Amy Tian, Tony Leng, Michael Yipeng Ye, Sam Cheng,

Matt Colen, and Vivien Cheng who have kept in close touch with me even after my undergraduate years via reunions and video games.

- To my family, who has helped me grow throughout my PhD and always encouraged me to look beyond the horizon to strive towards the long-term goals I have.
- To Day9, Dzeeff, 3Blue1Brown and Bon Appetit, which are Youtube channels produced by such wonderful content creators with amazing videos that I have consumed hours and hours on end. They have implicitly taught me so much about how to share educational content that they are individually so passionate about to an online audience.
- To LilyPichu, the most amazing and inspiring content creator I follow. Her humor and openness has fostered such a wonderful audience, and I hope to one day be able to share my interests and passions as well as she does.

Contents

Abstract **iii**

Contents **xi**

1 Introduction **1**
1.1 A brief primer on genomics . . . 1
1.2 Overview of covariance selection, applied to microarray expression data . . 4
1.3 Overview of one-dimensional changepoint detection, applied to comparative genomic hybridization data . . . 8
1.4 Overview of exponential-family embedding, applied to single-cell RNA-sequencing data . . . 12

2 Assessing heterogeneity – Covariance-based sample selection **17**
2.1 Introduction . . . 17
2.2 Data and model background . . . 19
2.3 Elementary analysis . . . 22
2.4 Methods: COBS (Covariance-based sample selection) . . . 24
2.5 Simulation study . . . 30
2.6 Application on BrainSpan study . . . 35
2.7 Conclusion and discussions . . . 40
2.A Code and dataset . . . 41
2.B Brain region details . . . 41
2.C Extension to the Stepdown method . . . 42
2.D Details of algorithms to find quasi-cliques . . . 43
2.E Formal description of simulation setup . . . 45
2.F Additional simulation results . . . 47
2.G Additional details on BrainSpan analysis . . . 49

3 Detecting heterogeneity – Fused lasso analysis **57**

3.1 Introduction . . . 57
3.2 Preliminary review of existing theory . . . 60
3.3 Sharp error analysis for the fused lasso estimator . . . 61
3.4 Extension to misspecified models . . . 65
3.5 Extension to exponential family models . . . 66
3.6 Approximate changepoint screening and recovery . . . 68
3.7 Summary . . . 71
3.A Proofs . . . 71
3.B Approximate changepoint recovery result, using post-processing . . . 86
3.C Comparison of Corollaries 20 and 21 to other results in the literature . . . 92
3.D Choosing a threshold level in the post-processing procedure . . . 94
3.E Numerical simulations to verify some of our theoretical results . . . 95

4 Detecting heterogeneity – Post-selection inference for changepoint significance 101
4.1 Introduction . . . 101
4.2 Preliminaries . . . 105
4.3 Inference for changepoint algorithms . . . 108
4.4 Practicalities and extensions . . . 113
4.5 Simulations . . . 115
4.6 Copy Number Variation (CNV) data application . . . 119
4.7 Conclusions . . . 120
4.A Code . . . 120
4.B Additional proofs . . . 120
4.C Additional algorithmic details . . . 124
4.D Model size selection using information criteria . . . 130

5 Modeling heterogeneity - Exponential-family embedding 135
5.1 Introduction . . . 136
5.2 Preliminary analysis . . . 137
5.3 Statistical model and background . . . 140
5.4 Method: eSVD (Exponential-family SVD) . . . 144
5.5 Statistical theory . . . 147
5.6 Numerical study . . . 151
5.7 Single-cell analysis . . . 154
5.8 Discussion . . . 159
5.A Code and reproducibility . . . 160
5.B Formal description of analysis pipeline . . . 160
5.C Discussion on estimator . . . 162
5.D Application of propositions to the curved Gaussian model . . . 168

5.E Additional simulation details/results 172
5.F Details on Slingshot and uncertainty tube 176
5.G Additional plots of results 177
5.H Proofs 177
5.I Auxiliary results and proofs 195

Bibliography **199**

One

Introduction

Genomic data such as DNA copy number variation, microarray expression, and RNA-sequencing data have led to many new discoveries about the genome over more than the three decades, but statistical methods to analyze such datasets have to account for non-trivial heterogeneity within the datasets in order to meaningfully estimate the desired parameter of interest. In this thesis, we broadly interpret heterogeneity to refer to the phenomenon that the genomic data we collect can *not* be trivially modeled as i.i.d. drawn from a fixed distribution. The source of the heterogeneity could arise from technical variation in new biotechnology (for example, the measurement uncertainty in the laboratory machines that measure gene expression) or from biological noise in cells or tissues with possibly different genomic makeup (for example, the gene expression differences that naturally arise from cells at different stages of development). Modeling and accounting for these different sources of heterogeneity has lead to developments in numerous areas in statistics such as batch correction (Sun et al., 2012), imputation (Chen and Zhou, 2018), low-dimensional embedding (Pierson and Yau, 2015), clustering (Zhu et al., 2019), feature selection (Witten et al., 2009), changepoint detection (Chen et al., 2017), graphical models (Fan et al., 2018a), and many more. Understanding the heterogeneity across among the samples can help address the scientific question directly, but also inform how to model the data in downstream analyses. In this thesis, we highlight a collection of four different papers, each its own chapter, that address a particular aspect of heterogeneity in genomic data that can be improved upon among this vast landscape of research.

All figures in this chapter are used only to demonstrate or visualize different concepts needed for the remaining chapters of the thesis.

1.1 A brief primer on genomics

The purpose of this chapter is to provide a reader with a concise, simplified and targeted overview of genomics so readers have the necessary biological background to approach the upcoming chapters in this thesis. At the risk of over-simplifying decades of genomic research, the so-called "central dogma" of biology states that DNA (the sequence of specific nucleic

acids that offsprings inherit from their two parents, and is shared among the vast majority of cells in an organism) transcribes into RNA, when translates into proteins. These proteins are synthesized to perform various roles within the body, such as aiding in the structure, function or regulation of the body's tissues and organs. This process is often succinctly depicted as

$$\text{DNA} \implies \text{RNA} \implies \text{Protein},$$

and illustrated as a schematic in Figure 1.1. Although this central dogma provides a compact summary of biology, understanding the specific biological factors that drive (or hinder) this process require sophisticated data to be collected, as well as sophisticated statistical methods to analyze said data. Different laboratory machines collect data via different biochemical processes to investigate different aspects of this process. While modeling how the central dogma works mechanically is well beyond the scope of this thesis (as it requires a deep understanding of biochemistry and biophysics), the aim of this thesis is to design methods that, when applied to genomic data, allow patterns within the data to reveal themselves. Not only do these method provide a qualitative assessment of biological questions (ex: how "similar" are these two cells in terms of their gene expression profiles, or how "significantly different" is the copy number in this segment of DNA different from an adjacent segment?), but they also provide a data-driven approach to inform the geneticist on what hypotheses to investigate next in future experiments (ex: do this cell type develop into two different cell types over time, or are copy number variations in these genes possibly causal of breast cancer?)

Genetic variation is one central factor that drives or hinder transcription and translation – that is, variation in the genome (i.e., the entire set of DNA in an organism) across all humans. Broadly speaking, this genetic variation comes in two types – structural variation (i.e., differences in the DNA sequence) or expression variation (i.e., differences in which genes[1] are expressed).

- **Structural variation**: Roughly speaking, structural variation asks *what* is different in the genome (i.e., how the specific sequence of nucleotides in the DNA differs) between individuals (or cells within an individual, if tumor cells arise). This variation can be caused by many mechanisms, such as mutations. Mutations are permanent alternations within a segment of the DNA which often occurs due to errors during cell replication (i.e., mitosis or meiosis). While most mutations are non-damaging, sometimes mutations can accumulate over time, especially if the mutation itself affects the cell replication process. The accumulation of such mutations in certain genes (designed segments within the genome that encode for RNA or a protein) can lead to

[1] In our thesis, we define a gene to be a particular continuous segment of DNA that encodes for RNA or a protein.

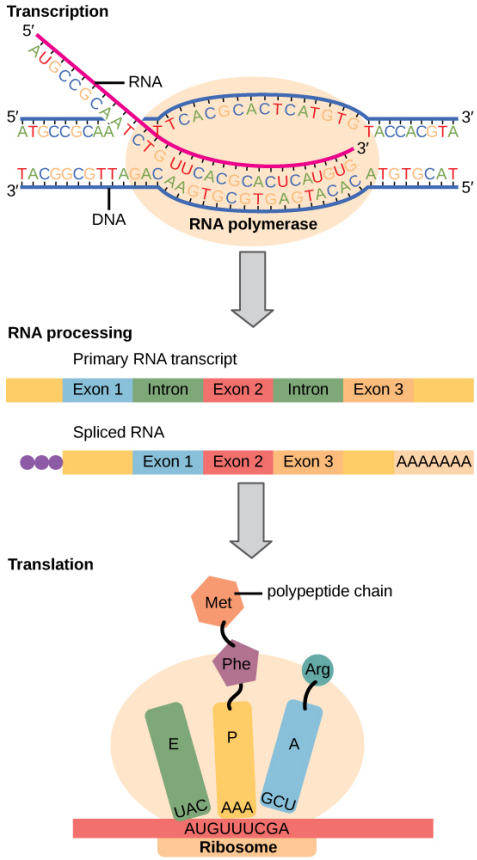


Figure 1.1: Figure taken from a chapter from Lumen's course "Boundless biology," available at `https://courses.lumenlearning.com/boundless-biology/chapter/the-genetic-code/`. Transcription is shown at top, where RNA is made from DNA. The letters A, T, C, G, and U represent different nucleotides. RNA processing is shown in the middle, where the introns (regions of mRNA that do not code for proteins) are removed in a process called RNA splicing, leaving on the exons (regions of mRNA that code for proteins). Translation is shown at the bottom, where the exons are used to dictate which amino acid sequences are built via a polypetide chain.

eventual development of autism spectral disorder in the individual, or the formation of tumor cells within certain tissues in the individual. This can lead to *de novo* loss-of-function mutations (discussed in §1.2) or copy number variation (discussed in §1.3).

- **Expression variation**: Roughly speaking, expression variation asks *how* the genes function differently across different cells, tissues, or individuals. The term "gene expression" often colloquially refers to the relative amount of RNA produced by a

particular gene. This varies naturally among cells due to differing cell types (such as different types of cells in the central nervous system, as discussed in §1.2 and §1.4), but can also vary among cells or individuals due to the structural variations mentioned above.

Geneticists often strive to understand these two forms of genetic variation, as these variations can cause differences in phenotypes (for example, cause a development in autism spectral disorder) or determines how different cells function within an organism. To achieve these goals, geneticists use a variety of laboratory machines to collect data across different samples, where the particular type of machine is chosen so that the genetic variation to be studied induces heterogeneity within the collected data. As statisticians, we hope to model the heterogeneity in the data to meaningfully reverse-engineer and advance our understanding of the genetic variation.

1.2 Overview of covariance selection, applied to microarray expression data

In Chapter 2, we focus on assessing the heterogeneity of microarray data measuring expression variation among brain tissues originating from different brain regions and different developmental age. The broader scientific question addressed in this chapter is to discover which genes are highly associative with autism spectrum disorder (ASD) when a damaging mutation occurs within them. ASD is a neurodevelopmental disorder that is characterized by impaired social functions and repetitive behavior. However, a lot of the statistical pipeline to address this question has already been developed in work like He et al. (2013); Liu et al. (2014, 2015). This chapter, broadly speaking, develops a statistical method to improve the sample size for this analysis by finding a homogeneous subset of samples among a heterogeneous dataset.

The work in this chapter resulted in the publication,

Lin, K. Z., Liu, H., and Roeder, K. (2020b). Covariance-based sample selection for heterogeneous data: Applications to gene expression and autism risk gene detection. *Journal of the American Statistical Association*, (To appear):1–22

1.2.1 Scientific background

Detecting which genes are highly associative of ASD when mutated can help future researchers better understand the genomic basis of ASD as well as design better treatment, but searching across the genome for so-called "autism risk genes" can be extremely timely and costly. A standard analysis to find autism risk genes involves sequencing the genome of trios (an individual with ASD as well as the two parents without ASD) and determining which

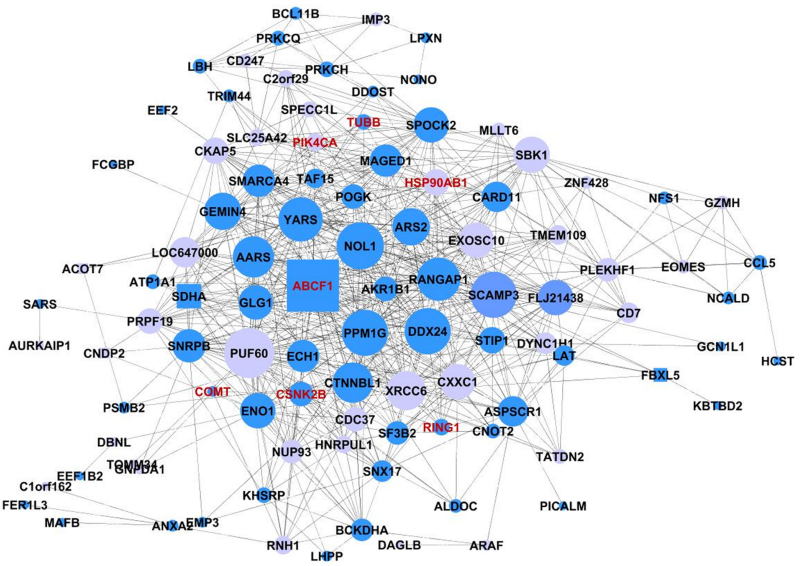


Figure 1.2: Figure taken from de Jong et al. (2012). This graph represents an exemplary gene co-expression network. Here, each node represents a gene (shown by its gene symbol), and an edge is present between two genes if the two genes are co-regulated or co-expressed biologically. This graph can be inferred by many ways, but in the thesis, we infer it directly from the microarray data measuring gene expression.

genes have a mutation in the individual with ASD that leads to severe disruption in how it is expressed (if any). This type of mutation is called *de novo* loss-of-function (dnLoF) mutations, which provides a great signal-to-noise ratio, but unfortunately, are extremely rare to observe in sequencing data. In fact, among thousands of trios sequenced, only a few dozens were genes were deemed as autism risk genes, and preliminary studies suggest there are still hundreds of genes left to be identified (Buxbaum et al., 2012).

More modern analyses rely on pooling other forms of genomic information aside from dnLoF mutations to infer likely autism risk genes. For example, using only sequencing data, TADA (He et al., 2013) models other types of mutations or transmitted variation to infer which genes are likely autism risk genes. Furthermore, a sequence of papers extended this analysis by developing DAWN (Liu et al., 2014, 2015), a method which additionally uses microarray data measuring gene expression to infer a gene co-expression network. An example of a gene co-expression network is shown in Figure 1.2. This graph is shown to bolster the power of previous analyses such as TADA since genes that are co-regulated with autism risk genes are likely to be autism risk genes themselves.

The heterogeneity we tackle in Chapter 2 occurs within this microarray data within

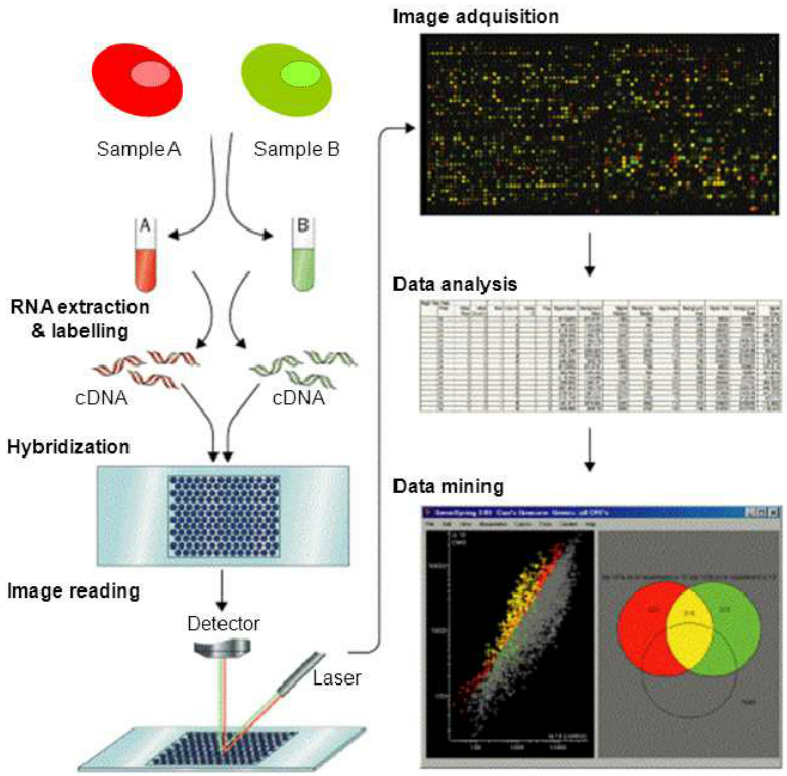


Figure 1.3: Figure taken from Lamas et al. (2012). This figure demonstrates how microarray technology works. Specifically, this figure shows a two-channel microarray, where two different samples of genetic material (Sample A and Sample B) are separately processed via RNA extraction and labeling. They are combined and placed into different probes of the microarray, where hybridization hames within each probe. This process releases light processed by a laser, where the light varies from red to green in different intensities. The light color and intensity measures the relative gene expression between the two samples. The resulting image is shown on the top right. This data then is processed in a statistical analysis.

the DAWN framework. Microarray data designed to infer the expression variation, broadly speaking, relies on reading light intensities arising from a chemical reaction to infer the amount of mRNA that is produced by a particular gene. One example of this technology is schematically explained in Figure 1.3. This light intensity is colloquially referred to as "gene expression" as genes that are highly expressed produce a higher amount of mRNA. Such datasets is exemplified in Figure 1.4. The amount of expression depends on many biological factors such as what type of tissue is analyzed or the developmental age of the cells in that tissue. Hence, to make the gene co-expression network as useful as possible to infer autism risk genes, Liu et al. (2014) uses microarray data measuring the gene expression of brain

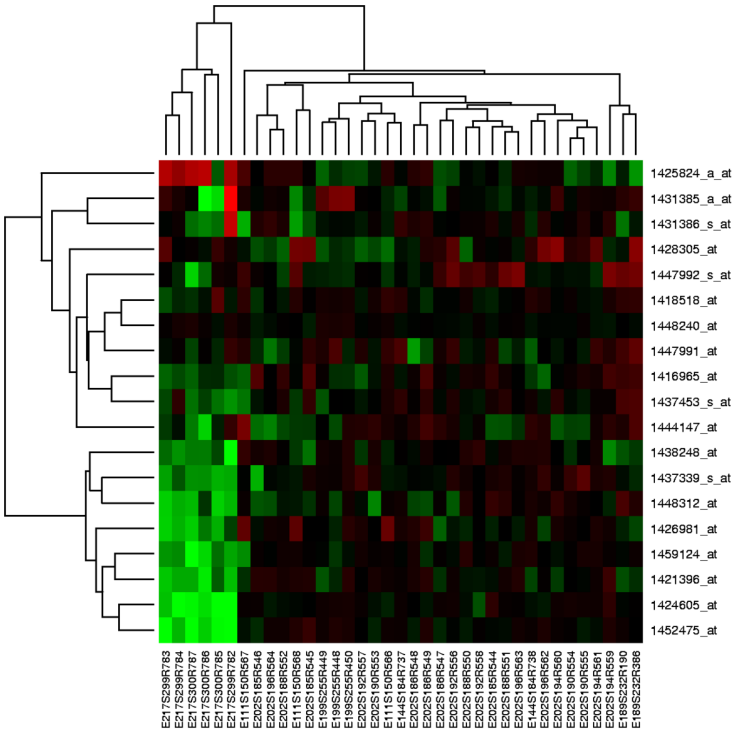


Figure 1.4: Figure taken from Wikipedia (page for "Gene expression profiling"). This figure shows a colorized example of a possible microarray dataset measuring gene expression, where each row i represents a different sample and each column j represents a different gene (here, named using their Ensembl IDs (ENSG)). Roughly speaking, this matrix is formed after restructuring the imaging data exemplified in the top right panel of Figure 1.3. The saturation of each entry in this figure represents the magnitude of the value in the microarray dataset, while the color of each entry represents its sign. Hierarchical clusterings of the rows and the columns are also shown.

tissues, ranging from different regions of the brain and different developmental ages. This dataset is first published in Kang et al. (2011) and is commonly known as the BrainSpan dataset. This choice of analyzing brain tissue is natural, as ASD is a neurological disorder. However, since gene expression varies wildly with the brain region as well as the tissue's developmental age, early analyses in Liu et al. (2014, 2015) focus primarily on microarray data originating from a particular choice of brain region and developmental age, and discard the remaining data.

1.2.2 Statistical novelty and heterogeneity

In Chapter 2, instead of bluntly discarding microarray data originating from other brain regions or developmental ages, we design a statistical method to model this source of heterogeneity. We call this procedure Covariance-based Sample Selection (COBS). This is useful for determining which microarray datasets are "similar enough" to that from our initial choice of brain region and developmental age. Here, we design our metric of similarity between two microarray datasets based on their covariance matrices, as we are planning to estimate the gene co-expression network using a Gaussian graphical model in our downstream analysis. After determining which microarray datasets are more-or-less homogeneous from this statistical perspective, we can aggregate all such datasets together to improve the sample size of our downstream analysis.

1.2.3 Scientific results

After deploying COBS to analyze the BrainSpan dataset, we obtain a set of microarray data with a larger sample size. This in turn provides a better estimation of the Gaussian graphical model, which is used in the DAWN framework to detect more autism risk genes. We provide various diagnostics to demonstrate the validity of our method and results.

1.3 Overview of one-dimensional changepoint detection, applied to comparative genomic hybridization data

In Chapters 3 and 4, we focus on as detecting heterogeneity within array comparative genomic hybridization data (aCGH) to determine regions of copy number variation in the genome (a particular type of structural variation often thought to be caused by mutations). The broader scientific question addressed in this chapter is to understand how copy number variation is associated with the developmental of tumor cells in an individual. Statistically, aCGH data is often analyzed by using a one-dimensional changepoint detection algorithm such as fused lasso (Tibshirani and Wang, 2008) or variants of binary segmentation (Olshen et al., 2004), but the statistical properties of such algorithms has only been studied thoroughly with the last decade. These chapters, broadly speaking, continue this line of statistical work by first investigating the changepoint detection convergence rates and then investigating the statistical uncertainty of the size of the detected changepoints within the post-selection inference framework.

The work in these chapters result in the following publication and preprint respectively,

Lin, K. Z., Sharpnack, J., Rinaldo, A., and Tibshirani, R. J. (2017). A sharp error analysis for the fused lasso, with application to approximate changepoint screening. In *Advances in Neural Information Processing Systems*, pages 6884–6893

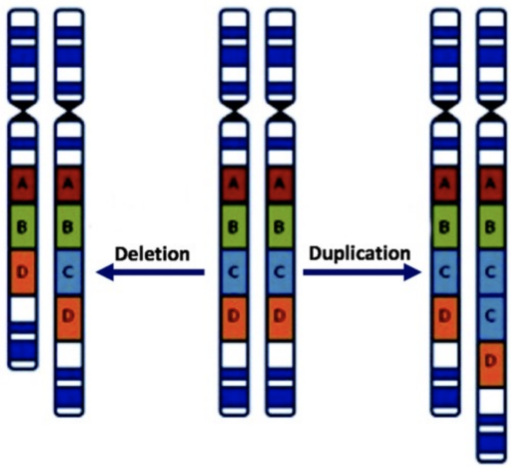


Figure 1.5: Figure taken from a blog post `https://medium.com/intothegenomics/cnvs-copy-number-variants-context-detection-methods-and-exploratory-data-analysis-with-python-986de6a58072`. In the middle, the both copies of the genome within an individual's cells are shown (one inherited from the father, the other inherited from the mother). Here, the letters A through D represent four arbitrary segments of of the genome. A deletion (also called a loss) in copy number variation is shown on the left, where one segment of genome (here, C) is erroneously deleted, possibly from only one of the two copies. A duplication (also called a gain) in copy number variation is shown on the right where one segment of genome is erroneously duplicated, possibly from only one of the two copies.

Hyun, S., Lin, K. Z., G'Sell, M., and Tibshirani, R. J. (2018b). Post-selection inference for changepoint detection algorithms with application to copy number variation data. *arXiv preprint arXiv:1812.03644*

1.3.1 Scientific background

In this section, we are focus on a particular structural variation called copy number variation, where long continuous segments of DNA are erroneously deleted or duplicated. Under normal circumstances, all cells in a human have the same genome across all 23 chromosomes. However, copy number variation is often caused by mutations that occur during cell replication (i.e., mitosis or meiosis), and is often attributed to the development of cancer tumors. Examples of copy number variations are illustrated in Figure 1.5.

While §1.2 discuss using microarray technologies to infer expression variation, this section uses microarray technologies to infer structural variation instead. These microarrays use a process called comparative genomic hybridization (CGH) to determine the relative gains and losses in the genome between a reference control cells (i.e., "normal cells") and a collection

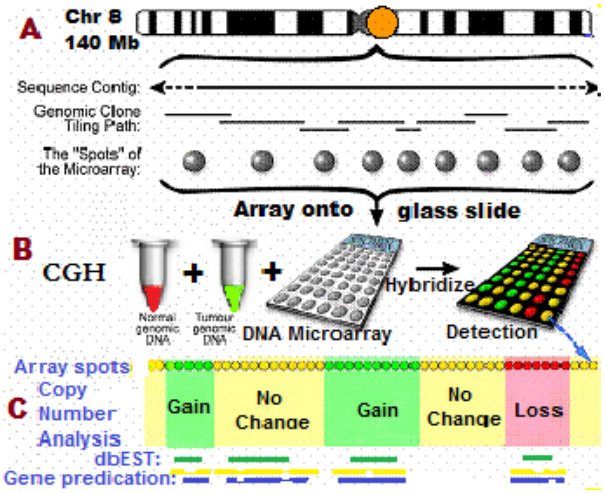


Figure 1.6: Figure taken from Alzeyadi (2013). A cartoon of chromosome 8 is shown in Panel A, where the gray spheres represents regions in the DNA where the microarray's probes are designed to target. The setup of a typical CGH analysis is shown in Panel B, where the reference control cells are shown in the red solution and the tumor cells are shown in the green solution. Both solutions are mixed together across all the probes of a microarray, where the light intensities resulting the chemical reaction are measured. Based on the light intensities, which infer the relative prevalence of each region of genome between the reference and tumor cells, the copy number analysis then reconstructs which regions of the genome in the tumor cells had a gain (i.e., duplication) or loss (i.e., deletion), shown in Panel C.

of tumor cells. These microarrays are different from the ones discussed in §1.2 since, instead of the probe inferring the amount of mRNA produced by different segments of DNA, the probes here are designed to hybridize directly with different segments of DNA. Roughly speaking, if there are duplications at the genome region that the probe is designed for, there are opportunities for that region to hybridize, yielding more reactions to happen that be picked up by the microarray machine. This induces heterogeneity in the data, which we aim to model and detect. One example of this technology is schematically explained in Figure 1.6.

However, aCGH data is often very noisy. One reason is technical noise, since there is noise during the hybridization or imaging steps that occur during data collection. However, another reason is due to the fact that collection of tumor cells used in CGH analyses often is not "pure". That is, it often contains a mixture of tumor cells and normal cells, inducing another source of noise in the collected data. An example of such data is shown in Figure 1.7. Given such a dataset, which is a vector in $\mathbb{R}^n$, geneticists often use a changepoint detection method to model the observed aCGH data as noisy observations from a piecewise constant

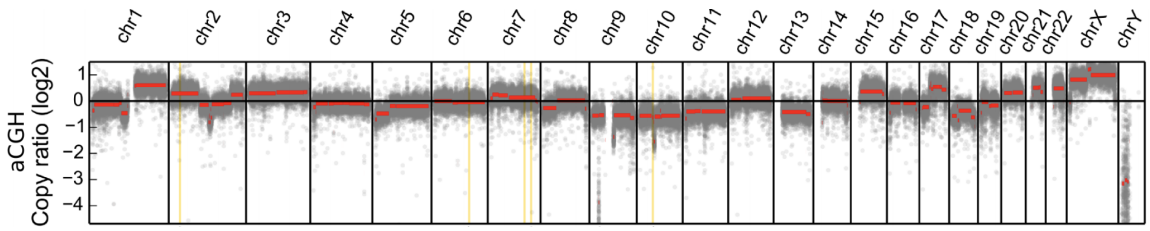


Figure 1.7: Figure taken from Talevich and Shain (2018). This figure show an example of a possible aCGH dataset (i.e., a microarray dataset measuring copy number variation via CGH). The data is represented a vector in $\mathbb{R}^n$, where entry i (shown as a gray point) represents the relative light intensity measured by the microarray machine at probe i, and all the probes are sorted based on their position across all 22 autosomal chromosomes and both sex chromosomes. Here, the x-axis shows the position of these probes across these chromosomes, while the y-axis shows the measured ratio in copy number between the reference and tumor cells. The red line shows the fitted piecewise constant function, estimated by a changepoint detection method.

function. This is a natural idea, since the different pieces of this function represent continuous segments of the genome that have been duplicated or deleted. However, the changepoints estimated by these methods are random, and there is need for statistical theory to quantify how reliable these estimates are.

1.3.2 Statistical novelty and heterogeneity

In Chapter 3, we abstract the heterogeneity in aCGH data mathematically, and investigate the theoretical properties of the estimated changepoints in a generic setting. Specifically, we prove an intimate relation between the convergence rate of estimating the underlying piecewise constant vector and the convergence rate of estimating the changepoints themselves. We apply this relation to study the changepoint properties of fused lasso primarily (Tibshirani et al., 2005), but show that is applies to other one-dimensional changepoint detection methods more generally.

In Chapter 4, we focus on aCGH data directly, where we are additionally interested in how to assess the statistical uncertainty of the estimated magnitude of the change between neighboring segments of the piecewise constant function. We assess this uncertainty within the post-selection framework, where we design various methods to output p-values that quantify the estimated changepoint's statistical significance. These different p-values reflect different null hypothesis being tested. These methods can be used to either investigate specific changepoints estimated in aCGH data, or can be applied in mass to multiple aCGH datasets as a screening tool to remove spurious changepoints.

1.3.3 Empirical results

In Chapter 4, we show that our post-selection inference method can be successfully applied to a common benchmark aCGH dataset first published in Snijders et al. (2001). We show that our method can successfully identify spurious changepoints whose p-values are larger than a pre-specified significance level, and the remaining changepoints correspond to true duplications or deletions in the genome, verified by karyotyping.

1.4 Overview of exponential-family embedding, applied to single-cell RNA-sequencing data

In Chapter 5, we focus on quantifying the heterogeneity within single-cell RNA-sequencing (RNA-seq for short) data that measures the gene expression of different oligodendrocytes. The broader scientific question addressed in this chapter is to analyze how oligodendrocytes develop over time, cells part of the central nervous system which provide support and insulation to axons. Statistically and computationally, this task first requires embedding each oligodendrocyte into a lower-dimensional space based on the measured single-cell RNA-seq data. This chapter, broadly speaking, develops a non-linear dimension reduction method based one-parameter exponential-family distributions to perform this embedding task.

This work in this chapter resulted in the preprint,

> Lin, K. Z., Lei, J., and Roeder, K. (2020a). Exponential-family embedding with application to cell developmental trajectories for single-cell RNA-seq data

1.4.1 Scientific background

While Sections 1.2 and 1.3 discuss using microarray technologies to infer expression variation or structural variation respectively, this section relies on using sequencing technologies to infer the expression variation. This revolution of sequencing technologies to replace microarray technologies is often called "next generation sequencing" (NGS) since sequencing technologies count the amount of RNA produced by a gene more-or-less directly, whereas microarray technologies rely on measuring light intensity as a proxy for the amount of RNA produced. This technology is schematically explained in Figure 1.8. We will call the data produced by this type of technology collectively as RNA-seq data. This is often also called "count data" as the data matrix contains entries which measures how many times there was a read within gene j in sample i. Here, each sample often refers to many cells from a certain tissue within an organism. From a statistician's perspective however, the data is visualized very similarly to microarray data, as in Figure 1.6, as both technologies result in a matrix that represent each gene as a different column and each sample as a different row.

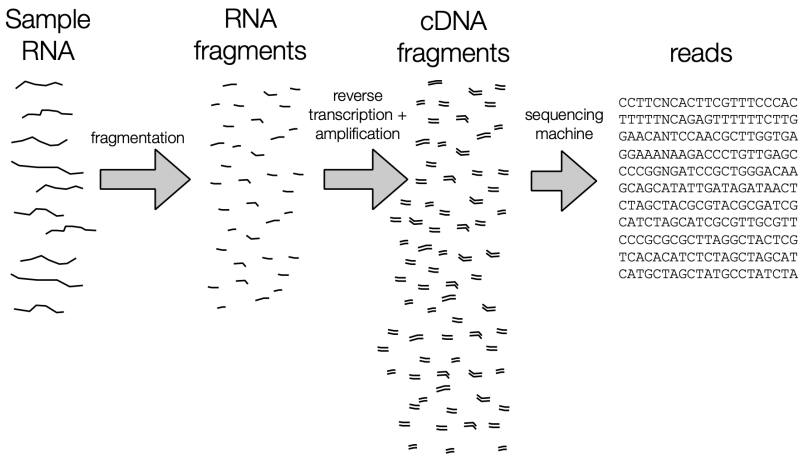


Figure 1.8: Figure taken from Ferdous and Ullah (2017). This figure demonstrates how RNA-sequencing technology works generally. The RNA strands (presumably produced by the genome of the cells in the sample of the interest) is shown on the left. The sequencing machine then fragments the RNA into smaller strands, after which a reverse transcription (where cDNA is matched to the RNA fragments) and amplification (where the RNA is copied multiple times, so cDNA can match to more strands) occurs. Finally, the sequencing machine reads all the cDNA strands produced in this process, shown on the right. Afterwards (not shown in the diagram), each read is matched onto the genome, which determines which gene gets an additional count.

Within the last decade however, a new breakthrough allowed geneticists to apply this technology on the cellular level. This is often called single-cell RNA-seq data, where the sequencing technology is applied to individual cells. That is, the collected data is still count data, but now sample i represents a specific cell in an organism, as opposed to many cells from a certain tissue. One example of this technology is schematically explained in Figure 1.9.

By measuring individual cells, geneticists can investigate biological questions at a much finer resolution. For example, one can ask about the cell developmental lineage of a particular cell type – that is, how do cells develop and specialize over an organism's lifespan. In this chapter, we focus specifically on single-cell RNA-seq data that measure the gene expressions of oligodendrocytes, first published in Marques et al. (2016). Understanding how oligodendrocytes develop and specialize over time is an important task, as the failure for oligodendrocytes to specialize can lead to various disorders such as multiple sclerosis (MS). This process of development and specialization is called the developmental lineage, and a cartoon of oligodendrocytes' lineage is shown in Figure 1.10. Geneticists estimate these lineages using single-cell RNA-seq by assuming that the gene expression profiles smoothly evolves across the lineage. To do this, geneticists first embed each cell into a meaningful

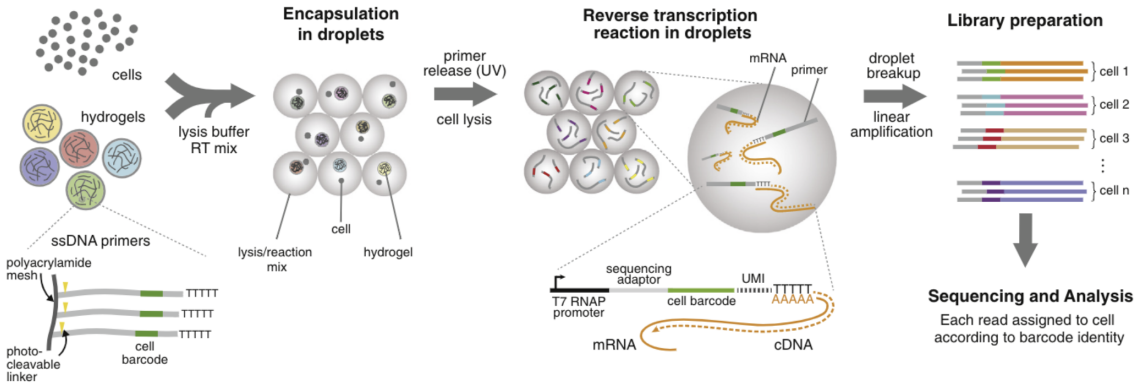


Figure 1.9: Figure taken from Klein et al. (2015). This figure demonstrates of how a particular single-cell RNA-sequencing technology works. Specifically, this figures shows a droplet-barcoding schematic on how to obtain single-cell RNA-seq data. The cells and hydrogels are shown on the left, where each hydrogel contains material to form a cell barcode. After combining each cell to a different hydrogel (each with containing material for a different cell barcode) in a droplet, reverse transcription happens separately in each droplet. Specifically, as mRNA and cDNA is formed within each droplet, the cDNA retains the cell barcode. Then, the cDNA from all the droplets are collected together, and then sequenced all together, as shown on the right. Because each cell's unique barcode remains on its corresponding cDNA, the machine can determine which reads belong to which cells when sequencing.

lower-dimension space based on the single-cell RNA-seq data, where nearby points in this space represents cells that are similar developmentally, and then apply various algorithms to infer one-dimensional curves within this embedding space. These estimated curves would represent the inferred developmental lineage. An example visualizing such results is shown in Figure 1.11.

1.4.2 Statistical novelty and heterogeneity

In Chapter 5, instead of using the SVD embedding (which is arguably the most common embedding), we develop an exponential-family SVD (eSVD), an embedding with respect to an arbitrary one-parameter exponential-family distribution. This allows us to embed the single-cell RNA-seq data in a non-linear fashion. Our embedding builds upon the existing literature in matrix factorization, where we design our estimator to estimate a low-rank matrix of natural parameters within a particular hierarchical dot product model. This low-rank structure captures of the notion of heterogeneity in this chapter, as it implies that each cell is associated with a different low-dimensional latent vector. While many such embeddings of this flavor already exist, we design our method such that it is computationally simpler than existing convex methods but still retains desirable statistical properties such as identifiability and consistency.

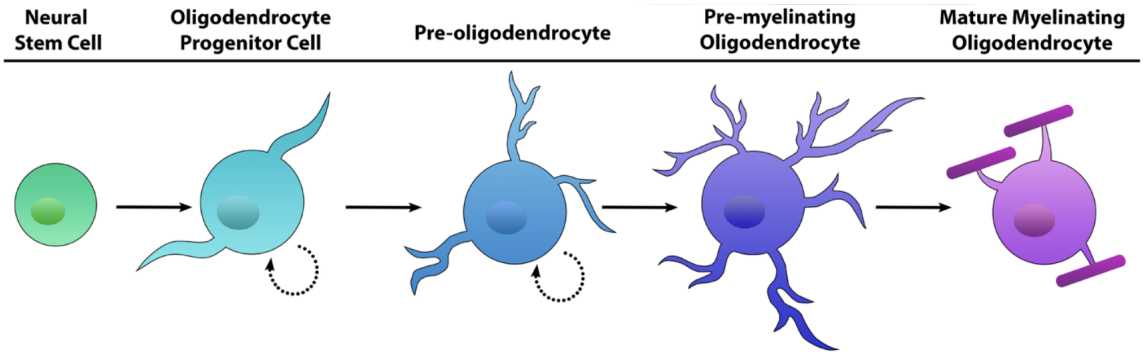


Figure 1.10: Figure taken from Newville et al. (2017). This cartoon demonstrates five possible categories of different oligodendrocytes, developing from a neural stem cell shown on the left to a mature myelinating oligodendrocyte shown on the right. However, in reality, such lineage might not be as linear, as neural stem cell might develop to many different types of mature oligodendrocytes.

1.4.3 Scientific results

We apply eSVD using a curved Gaussian distribution (where the standard deviation is proportional to the mean) to analyze the oligodendrocytes by embedding each cell into a low-dimensional space. We then apply Slingshot (Street et al., 2018) (a competitor algorithm to Monocle (Trapnell et al., 2014)) on the cells in this low-dimensional space to estimate the cell developmental lineages. We find that the oligodendrocytes develop into two different types of mature oliodendrocytes.

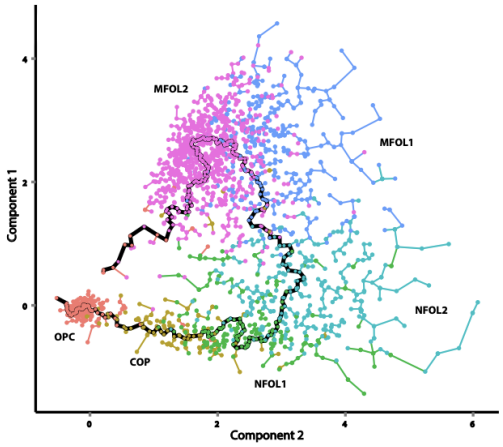


Figure 1.11: Figure taken from Marques et al. (2016). This figure exemplifies the task of estimating the cell lineage across six different cell types. Here, each of the cell measured in the single-cell RNA-seq dataset is embedded into two-dimensional space using independent component analysis (ICA), where the different colors represent the six different cell types. Then, the authors use a particular cell lineage estimator called Monocle (Trapnell et al., 2014) to estimate the lineage, represented by the thick black line that roughly interpolates the center of the cell types.

Two

Assessing heterogeneity – Covariance-based sample selection

Paper summary: Risk for autism can be influenced by genetic mutations in hundreds of genes. Based on findings showing that genes with highly correlated gene expressions are functionally interrelated, "guilt by association" methods such as DAWN have been developed to identify these autism risk genes. Previous research analyzes the BrainSpan dataset, which contains gene expression of brain tissues from varying regions and developmental periods. Since the spatiotemporal properties of brain tissue is known to affect the gene expression's covariance, previous research have focused only on a specific subset of samples to avoid the issue of heterogeneity. This leads to a potential loss of power when detecting risk genes. In this article, we develop a new method called COBS (COvariance-Based sample Selection) to find a larger and more homogeneous subset of samples that share the same population covariance matrix for the downstream DAWN analysis. To demonstrate COBS's effectiveness, we utilize genetic risk scores from two sequential data freezes obtained in 2014 and 2020. We show COBS improves DAWN's ability to predict risk genes detected in the newer data freeze when utilizing the risk scores of the older data freeze as input.

The work in this chapter was done jointly with Han Liu and Kathryn Roeder, and has been accepted to JASA Applications and Case Studies under the title, "Covariance-based sample selection for heterogeneous data: Applications to gene expression and autism risk gene detection."

2.1 Introduction

The genetic cause of autism spectrum disorder (ASD), a neurodevelopmental disorder that affects roughly 1-2% individuals in the United States, remains an open problem despite decades of research (Autism and Investigators, 2014). ASD is characterized primarily by impaired social functions and repetitive behavior (Kanner et al., 1943; Rutter, 1978). To

better understand this disorder, scientists identify specific genes that are liable for increasing the chance of developing ASD when damaged or mutated (Sanders et al., 2015). These are genes are called risk genes. While breakthroughs in genomic technologies and the availability of large ASD cohorts have led to the discovery of dozens of risk genes, preliminary studies suggest there are hundreds of risk genes still unidentified (Buxbaum et al., 2012). In this work, we build upon the current statistical methodologies to further improve our ability to identify risk genes.

We focus on statistical methods that use gene co-expression networks to help identify risk genes. These networks are estimated from brain tissue's gene expression data. Since these gene co-expression networks provide insight into genes that regulate normal biological mechanisms in fetal and early brain development, it was hypothesized that risk genes that alter these mechanisms should be clustered in these networks (Šestan et al., 2012). Early findings confirmed this hypothesis (Parikshak et al., 2013; Willsey et al., 2013). These results led to the development of the Detection Association With Networks (DAWN) algorithm which uses a "guilt by association" strategy – implicating new risk genes based on their connectivity to previously identified risk genes (Liu et al., 2014, 2015). However, the previous DAWN analyses suffer from statistical limitations that we will investigate and resolve in this article.

We challenge previous analyses' assumptions regarding the homogeneity of the covariance matrix in gene expression data. Previous DAWN analyses assume that gene expression samples from the same brain tissue type share the same covariance matrix. This assumption was influenced by the findings in Kang et al. (2011) and Willsey et al. (2013), which showed that gene co-expression patterns differ among different brain regions and developmental periods on average. Statistically, this means that the covariance matrix among the gene expressions may differ with respect to the spatiotemporal properties of the brain tissue. Hence, previous DAWN analyses (Liu et al., 2014, 2015) use only samples from a particular brain tissue type chosen by the findings in Willsey et al. (2013). However, no further statistical analysis is performed to check for homogeneity of this specific subset of samples. In addition, since previous DAWN analyses limit themselves to a subset of gene expression samples, many other samples assumed to be heterogeneous are excluded. This leads to a potential loss of power when identifying risk genes.

To overcome these limitations, we develop a method called COBS (COvariance-Based sample Selection), a two-staged procedure in order to select a subset of gene expression samples in a data-driven way that is more homogeneous and larger in sample size than the fixed subset used previously. In the first stage, we take advantage of the recent developments in high-dimensional covariance testing (Cai et al., 2013; Chang et al., 2017) to determine whether if the gene expression from two different brain tissues share the same population covariance matrix. We combine this with a multiple-testing method called Stepdown that

accounts for the dependencies among many hypothesis tests (Romano and Wolf, 2005; Chernozhukov et al., 2013). In the second stage, after determining which pairs of brain tissues have statistically indistinguishable covariance matrices, we develop a clique-based procedure to select which brain tissues to use in the downstream DAWN analysis. We show that COBS selects brain tissues within the BrainSpan dataset that align with current scientific knowledge and also leads to an improved gene network estimate for implicating risk genes. This article addresses the numerous algorithmic challenges needed to implement this idea.

In Section 2 in this chapter, we describe the data and statistical model for heterogeneity in the covariance matrix. In Section 3, we provide a visual diagnostic to investigate the homogeneity assumptions of previous DAWN analyses. In Section 4, we describe the different stages of COBS to find a subset of homogeneous samples within a dataset. In Section 5, we illustrate the properties of COBS on synthetic datasets. In Section 6, we apply our procedure on gene expression data to show that, when combined with DAWN, we have an improved gene network that can better implicate risk genes. Section 7 provides an overall summary and discussion.

2.2 Data and model background

Due to the challenge of obtaining and preserving brain tissue, datasets recording the gene expression patterns of brain tissue are rare. The BrainSpan project contributes one of the largest microarray expression datasets available (the "BrainSpan dataset" henceforth), sampling tissues from 57 postmortem brains with no signs of large-scale genomic abnormalities (Kang et al., 2011). Many studies have favored this dataset because its 1294 microarray samples capture the spatial and temporal changes in gene expression that occur in the brain during the entirety of development (De Rubeis et al., 2014; Dong et al., 2014; Cotney et al., 2015). While our paper focuses on this particular microarray expression dataset, our method would apply to other gene expression datasets such as RNA sequencing data.

The heterogeneity of gene expression due to the spatiotemporal differences in brain tissues presents statistical challenges. As documented in Kang et al. (2011), the region and developmental period of the originating brain tissue contribute more to the heterogeneity than other variables such as sex and ethnicity. To understand this heterogeneity, we use the following schema to model the BrainSpan dataset. Each of the 1294 microarray samples is categorized into one of 16 *spatiotemporal windows*, or *windows* for short, depending on which brain region and developmental period the brain tissue is derived from. Within each window, all microarray samples originating from the same brain are further categorized into the same *partition*. There are 212 partitions in total. Figure 2.1 summarizes how many partitions and microarray samples belong in each window in the BrainSpan dataset. This schema allows us to model the microarray samples more realistically since the gene co-expression

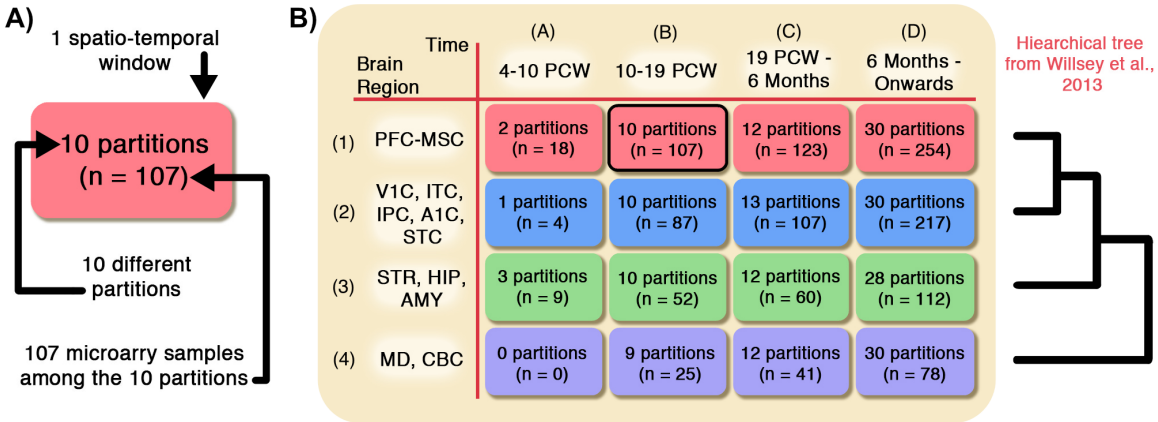


Figure 2.1: (A) 107 microarray samples grouped by the originating 10 brains. This forms 10 different partitions. Since all these partitions originate from the same brain region and developmental period, they are further grouped into the same window. (B) The 57 postmortem brains belong to 4 different developmental periods (columns). Here, PCW stands for post-conceptual weeks. Each brain is dissected and sampled at 4 different brain regions (rows). In total, over the 212 partitions, there are 1294 microarray samples, each measuring the expression of over 13,939 genes. Window 1B (outlined in black) is the window that previous work (Liu et al., 2015) focus on, and the hierarchical tree from Willsey et al. (2013) is shown to the right. Additional details about the abbreviations are given in Appendix 2.B.

patterns vary greatly on average from window to window (Willsey et al., 2013). Additionally, Willsey et al. (2013) find that among all the windows, the known risk genes in Window 1B are most tightly co-expressed. Window 1B is highlighted in Figure 2.1 and contains the 107 microarray samples from the prefrontal cortex and primary motor-somatosensory cortex from 10 to 19 post-conceptual weeks. Due to this finding, previous DAWN analyses focus on all 107 samples from 10 partitions, assuming that these samples were homogeneous without further statistical investigation, and discard the remaining 1187 samples, (Liu et al., 2014, 2015). We seek to improve upon this heuristical sample selection procedure, first by formalizing a statistical model.

2.2.1 Statistical model

We use a mixture model that assumes that microarry samples from the same partition are homogeneous while samples from different partitions could be heterogeneous. For the pth partition, let $X_1^{(p)}, \ldots X_{n_p}^{(p)} \in \mathbb{R}^d$ denote n_p i.i.d. samples, and let $w(p)$ denote the window that partition p resides in. These n_p samples are drawn from either a distribution with covariance Σ, or another distribution with a different covariance matrix Σ_p. Our notation emphasizes that the distributions in consideration are not necessarily Gaussian, and Σ is the covariance matrix shared among all partitions, while Σ_p may vary from partition

to partition. A fixed but unknown parameter $\gamma_{w(p)} \in [0, 1]$ controls how frequently the partitions in window w are drawn from these two distributions, meaning it controls the amount of heterogeneity. For each partition p, this mixture model is succinctly described as,

$$\begin{aligned} I^{(p)} &\sim \text{Bernoulli}(\gamma_{w(p)}), \\ X_1^{(p)}, \ldots, X_{n_p}^{(p)} &\overset{i.i.d.}{\sim} \begin{cases} D(\Sigma) & \text{if } I^{(p)} = 1 \\ D(\Sigma_p) & \text{otherwise,} \end{cases} \end{aligned} \tag{2.1}$$

where $D(\Sigma)$ denotes an arbitrary distribution with covariance matrix Σ, and $I^{(p)}$ is the latent variable that determines whether or not the samples in partition p have covariance Σ or Σ_p. With this model setup, our task is to determine the set of partitions that originate from the covariance matrix Σ, which we will call

$$\mathcal{P} = \Big\{ p \ : \ I^{(p)} = 1 \Big\}. \tag{2.2}$$

The findings of Kang et al. (2011) and Willsey et al. (2013) inform us on how much heterogeneity to expect within a window via $\gamma_{w(p)}$. While analyses such as Liu et al. (2015) assume that all the samples in Window 1B are homogeneous, it is noted in Kang et al. (2011) that sampling variability in brain dissection and in the proportion of white and gray matter in different brain tissues can cause variability in the gene co-expression patterns. This means that scientifically, we do not expect all the partitions in Window 1B to be homogeneous (i.e., $\gamma_{w(p)} = 1$). Furthermore, Willsey et al. (2013) find a hierarchical clustering among the four brain regions. This is illustrated in Figure 2.1, where the gene co-expression patterns in the brain regions represented in first row are most similar to those in the second row and least similar to those in the fourth row. The authors also find a smooth continuum of gene expression patterns across different developmental periods, represented as the columns of the table in Figure 2.1. Hence, we expect $\gamma_{w(p)}$ to decrease smoothly as the window w becomes more dissimilar to Window 1B, in both the spatial and temporal direction.

2.2.2 Connections to other work

Other work use models similar to (2.1) on microarray expression data to tackle the different co-expression patterns among different tissues and subjects, but their methods differ from ours. One direction is to directly cluster the covariance matrices of each partition (Ieva et al., 2016). However, this approach does not account for the variability in the empirical covariance matrix, unlike our hypothesis-testing based method. Another approach is to explicitly model the population covariance matrix for each partition as the summation of a shared component and a partition-specific heterogeneous component. This is commonly used in batch-correction procedures where the analysis removes the heterogeneous component from each partition (Leek and Storey, 2007). However, we feel such an additive model is

too restrictive for analyzing the BrainSpan dataset, as we do not believe there is a shared covariance matrix across all windows of the brain. Instead, our approach will find specific set of partitions with statistically indistinguishable covariance matrices.

2.3 Elementary analysis

In this section, we develop a visual diagnostic to investigate if the 10 partitions in Window 1B used in previous work (Liu et al., 2014, 2015) are as homogeneous as these previous analyses assume. Using a hypothesis test for equal covariances, our diagnostic leverages the following idea: we divide the partitions into two groups and apply a hypothesis test to the samples between both groups. If all the partitions were truly drawn from distributions with equal covariances, then over many possible divisions, the empirical distribution of the resulting p-values should be roughly uniform. We can visualize this distribution by using a QQ-plot. The less uniform the p-values look, the less we are inclined to interpret our partitions to be all drawn from distributions with equal covariances. The following algorithm summarizes this diagnostic.

Algorithm 1: Covariance homogeneity diagnostic

1. Loop over trials $t = 1, 2, \ldots, T$:
   a) Randomly divide the selected partitions in the set $\widehat{\mathcal{P}}$ into two sets, $\widehat{\mathcal{P}}^{(1)}$ and $\widehat{\mathcal{P}}^{(2)}$, such that $\widehat{\mathcal{P}}^{(1)} \cup \widehat{\mathcal{P}}^{(2)} = \widehat{\mathcal{P}}$ and $\widehat{\mathcal{P}}^{(1)} \cap \widehat{\mathcal{P}}^{(2)} = \emptyset$.
   b) For each partition $p \in \widehat{\mathcal{P}}^{(1)}$, center the samples $X_1^{(p)}, \ldots, X_{n_p}^{(p)}$. Then aggregate all samples in $\widehat{\mathcal{P}}^{(1)}$ to form the set of samples
   $$\mathcal{X} = \bigcup_{p \in \widehat{\mathcal{P}}^{(1)}} \{X_1^{(p)}, \ldots, X_{n_p}^{(p)}\}.$$
   Similarly, form the set of samples $\mathcal{Y}$ from the set of partitions $\widehat{\mathcal{P}}^{(2)}$.
   c) Compute the p-value for a hypothesis test that tests whether or not the samples in $\mathcal{X}$ and $\mathcal{Y}$ have the same covariance matrix.
2. Produce a QQ-plot of the resulting T p-values to see if empirical distribution of the p-values is close to a uniform distribution.

We remind the reader that the above procedure is a diagnostic. This is not necessarily a recipe for a goodness-of-fit test since the T p-values are not independent, which makes it difficult to analyze its theoretical properties without a carefully designed global null test. However, as we will demonstrate in later sections of this article, this diagnostic is nonetheless able to display large-scale patterns in our dataset.

2.3.1 Specification of covariance hypothesis test

To complete the above diagnostic's description, we describe the procedure to test for equality of covariance matrices. Following the model (2.1), let $\mathcal{X} = \{X_1, \ldots, X_{n_1}\}$ and $\mathcal{Y} = \{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{n_2}\}$ be n_1 and n_2 i.i.d. samples from d-dimensional distribution with covariance Σ_X and Σ_Y respectively, both with an empirical mean of 0. We define $\mathbb{X} \in \mathbb{R}^{n_1 \times d}$ and $\mathbb{Y} \in \mathbb{R}^{n_2 \times d}$ as the matrices formed by concatenating these samples row-wise. Define the empirical covariance matrices as $\widehat{\Sigma}_X = \mathbb{X}^\top \mathbb{X}/n_1$, and $\widehat{\Sigma}_Y = \mathbb{Y}^\top \mathbb{Y}/n_2$, where we denote the individual elements of these matrices as $\widehat{\Sigma}_X = [\widehat{\sigma}_{X,ij}]_{1 \le i,j \le d}$ and likewise for $\widehat{\Sigma}_Y$.

We now discuss the hypothesis test for equal covariance, $H_0 : \Sigma_X = \Sigma_Y$, that we will consider in this article based on the test statistic defined in Chang et al. (2017) which extends Cai et al. (2013). In these works, the authors note that if $\Sigma_X = \Sigma_Y$, then the maximum element-wise difference between Σ_X and Σ_Y is 0. Hence, Chang et al. (2017) defines the test statistic $\widehat{T}$ as the maximum of squared element-wise differences between $\widehat{\Sigma}_X$ and $\widehat{\Sigma}_Y$, normalized by its variance. Specifically,

$$\widehat{T} = \max_{ij} \left(\widehat{t}_{ij}\right) \quad \text{where } \widehat{t}_{ij} = \frac{\left(\widehat{\sigma}_{X,ij} - \widehat{\sigma}_{Y,ij}\right)^2}{\widehat{s}_{X,ij}/n_1 + \widehat{s}_{Y,ij}/n_2}, \quad i,j \in 1, \ldots, d, \tag{2.3}$$

where $\widehat{s}_{X,ij} = \sum_{m=1}^{n_1} (\mathbb{X}_{mi}\mathbb{X}_{mj} - \widehat{\sigma}_{X,ij})^2/n_1$ is the empirical variance of the variance-estimator $\widehat{\sigma}_{X,ij}$, and $\widehat{s}_{Y,ij}$ is defined similarly.

Then, Chang et al. (2017) constructs an empirical null distribution of $\widehat{T}$ under $H_0 : \Sigma_X = \Sigma_Y$ using the multiplier bootstrap (Chernozhukov et al., 2013). On each of the $b \in \{1, \ldots, B\}$ trials, the multiplier bootstrap computes a bootstrapped test statistic $\widehat{T}^{(b)}$ by weighting each of the $n_1 + n_2$ observations by a standard Gaussian random variable drawn independently of all other variables, denoted collectively as $(g_1^{(b)}, \ldots, g_{n_1}^{(b)}, g_{n_1+1}^{(b)}, \ldots, g_{n_1+n_2}^{(b)})$. Specifically, we construct the bootstrap statistic for the bth trial as

$$\widehat{T}^{(b)} = \max_{ij} \left(\widehat{t}_{ij}^{(b)}\right) \quad \text{where } \widehat{t}_{ij}^{(b)} = \frac{\left(\widehat{\sigma}_{X,ij}^{(b)} - \widehat{\sigma}_{Y,ij}^{(b)}\right)^2}{\widehat{s}_{X,ij}/n_1 + \widehat{s}_{Y,ij}/n_2}, \quad i,j \in 1, \ldots, d, \tag{2.4}$$

where

$$\widehat{\sigma}_{X,ij}^{(b)} = \sum_{m=1}^{n_1} g_m^{(b)} \frac{\mathbb{X}_{mi}\mathbb{X}_{mj} - \widehat{\sigma}_{X,ij}}{n_1}, \quad \text{and} \quad \widehat{\sigma}_{Y,ij}^{(b)} = \sum_{m=1}^{n_2} g_{n_1+m}^{(b)} \frac{\mathbb{Y}_{mi}\mathbb{Y}_{mj} - \widehat{\sigma}_{Y,ij}}{n_2}.$$

We compute the p-value by counting the proportion of bootstrap statistics that are larger than the test statistic,

$$\text{p-value} = \frac{\left|\{b : |\widehat{T}^{(b)}| \ge |\widehat{T}|\}\right|}{B}.$$

Chang et al. (2017) prove that this test has asymptotically $1 - \alpha$ coverage under the null hypothesis as long as the all distributions in the distribution family D in (2.1) have sub-Gaussian and sub-exponential tails, even in the high-dimensional regime where $d \gg \max(n_1, n_2)$.

2.3.2 Application to BrainSpan

Equipped with a complete description of the diagnostic, we apply it to the BrainSpan dataset. Among the 10 partitions in Window 1B, we divide the partitions into two groups uniformly at random 250 times, and compute a p-value using Method 1 (with normalization) for each division using 200 bootstrap trials. The QQ-plot of the resulting p-values are shown in Figure 2.2A, where we see that the p-values are biased towards 0. This implies the 10 partitions in Window 1B are heterogeneous since they do not seem to all share the same covariance matrix. Furthermore, we apply this diagnostic to all partitions in the BrainSpan dataset with 5 or more samples. This results in using only 125 of the 212 partitions shown in Figure 2.1. The resulting p-values become more biased towards 0 (Figure 2.2B), implying the dataset as a whole is more heterogeneous than the partitions in Window 1B. In the next section, we develop a method to resolve this issue by finding the largest subset of partitions possible among the 125 partitions in the BrainSpan dataset that share the same covariance matrix.

2.4 Methods: COBS (Covariance-based sample selection)

While we have discussed a method to test for equivalent covariance matrices between any two partitions in §2.3, we cannot directly apply this method to select a large number of homogeneous partitions in the BrainSpan dataset without suffering a loss of power due to multiple testing. Since there are $r = 125$ partitions with more than 5 samples, applying the hypothesis test to each pair of partitions results in $\binom{r}{2} = 7750$ dependent p-values. These p-values are dependent since each of the r partitions is involved in $r - 1$ hypothesis tests. Hence, standard techniques such as a Bonferroni correction are too conservative when accounting for these dependencies, likely leading to a loss of power.

To properly account for this dependency, we introduce our new method called COBS, which comprises of two parts. First, we use a Stepdown method in Subsection 2.4.1 that simultaneously tests all $\binom{r}{2}$ hypothesis tests for equal covariance matrices, which builds upon the bootstrap test introduced previously in §2.3. After determining which of the $\binom{r}{2}$ pairs of partitions do not have statistically significant differences in their covariance matrices, we develop a clique-based method in Subsection 2.4.2 to select a specific set of partitions $\widehat{\mathcal{P}}$.

2.4.1 Stepdown method: multiple testing with dependence

We use a Stepdown method developed in Chernozhukov et al. (2013) to control the family-wise error rate (FWER). We tailor the bootstrap-based test in Subsection 2.3.1 to our specific

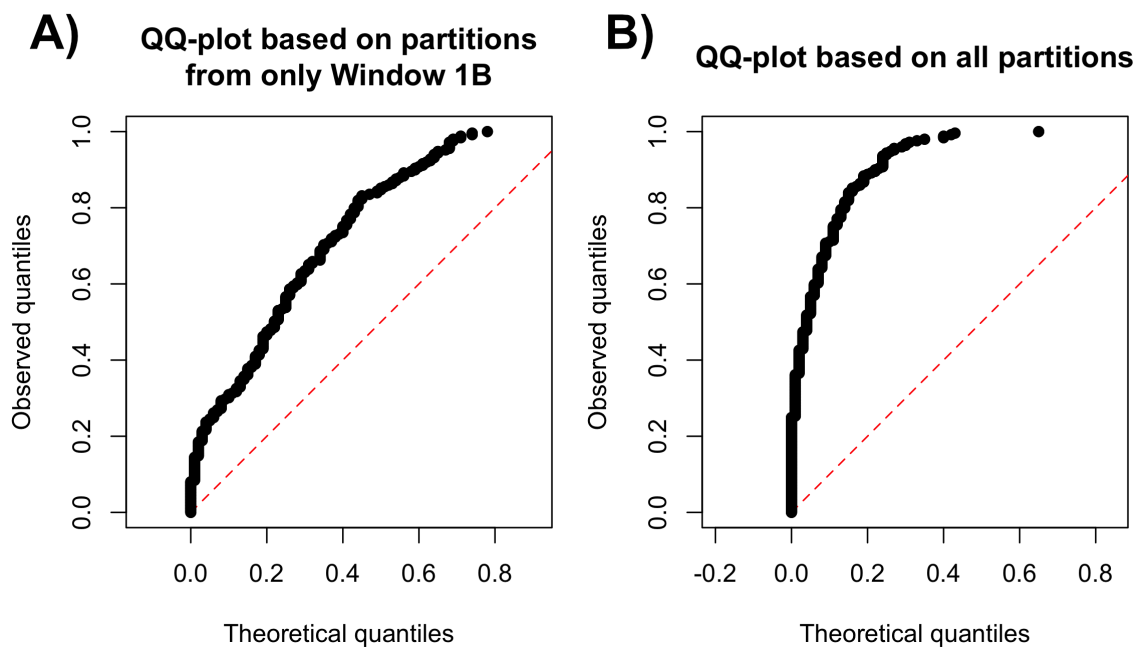


Figure 2.2: QQ-plots of the 250 p-values generated when applying our diagnostic to the BrainSpan dataset. (A) The diagnostic using only the partitions in Window 1B, showing a moderate amount of heterogeneity. (B) The diagnostic using all 125 partitions in the BrainSpan dataset, showing a larger amount of heterogeneity.

setting in the algorithm below. We denote $\widehat{T}_{(i,j)}$ as the test statistic formed using (2.3) to test if the covariance of samples between partition i and partition j are equal. Similarly, let $\widehat{T}^{(b)}_{(i,j)}$ denote the corresponding bootstrap statistics on the bth bootstrap trial. Here, quantile$(\{x_1, \ldots, x_n\}; 1-\alpha)$ represents the empirical $(1-\alpha) \cdot 100\%$ quantile of the vector $(x_1, \ldots, x_n)$.

Algorithm 2: Stepdown method

1. Initialize the list enumerating all $\binom{r}{2}$ null hypotheses corresponding to the set of partition pairs, $\mathcal{L} = \Big\{(1,2), \ldots, (r-1, r)\Big\}$.

2. Calculate $\widehat{T}_\ell$ for each $\ell \in \mathcal{L}$, as stated in (2.3).

3. Loop over steps $t = 1, 2, \ldots$:

   a) For each bootstrap trial $b = 1, \ldots, B$:

i. Generate $N = \sum_p n_p$ i.i.d. standard Gaussian random variables, one for each sample in each partition, and compute $\widehat{T}_\ell^{(b)}$ for all $\ell \in \mathcal{L}$, as stated in (2.4).

ii. Compute

$$\widehat{T}^{(b)} = \max\Big\{\widehat{T}_\ell^{(b)} : \ell \in \mathcal{L}\Big\}. \tag{2.5}$$

b) Remove any $\ell \in \mathcal{L}$ if

$$\widehat{T}_\ell \geq \text{quantile}\Big(\{\widehat{T}^{(1)}, \ldots, \widehat{T}^{(b)}\}; 1-\alpha\Big).$$

If no elements are removed from $\mathcal{L}$, return the null hypotheses corresponding to $\mathcal{L}$. Otherwise, continue to step $t+1$.

Using techniques in Romano and Wolf (2005) and Chernozhukov et al. (2013), it can be proven that this method has the following asymptotic FWER guarantee,

$$\mathbb{P}\Big(\text{no true null hypothesis among } \mathcal{H} \text{ null hypotheses are rejected}\Big) \geq 1 - \alpha + o(1) \tag{2.6}$$

under the same assumptions posed in Chang et al. (2017). The reason Algorithm 2 is able to control the FWER without a Bonferroni correction is because the null distribution in the Stepdown method is properly calibrated to account for the joint dependence among the $\binom{r}{2}$ tests. Specifically, when $\binom{r}{2}$ tests are individually performed as in Subsection 2.3.1, the test statistics (2.3) are dependent, but the bootstrapped null distributions do not account for this dependence. Hence, accounting for the dependence via a Bonferroni correction after-the-fact can lead to a substantial loss in power. However, in the Stepdown procedure, the bootstrapped null distributions retain the dependencies jointly since they are generated from the same N Gaussian random variables in each trial. See Chernozhukov et al. (2013) (Comment 5.2) for a further discussion.

Robustness concerns. In practice, due to the maximum function in the test statistic $\widehat{T}_\ell$ displayed in (2.3), the Stepdown method could possibly erroneously reject a hypothesis due to the presence of outliers. One way to circumvent this problem to purposely shrink the value of the test statistic $\widehat{T}_\ell$ while leaving the bootstrapped statistics $\widehat{T}_\ell^{(b)}$ in (2.4) the same. Specifically, we can replace $\max_{ij}(\widehat{t}_{ij})$ in (2.3) with the quantile$(\{\widehat{t}_{ij}\}_{ij}; 1-\epsilon)$, where ϵ is a positive number extremely close to 0. This has the desired effect of "discarding" the large values in $\{\widehat{t}_{ij}\}_{ij}$. Observe that this procedure would potentially lead to a slight loss in power, but the inferential guarantee in (2.6) still holds since there can only be strictly less rejections.

Computational concerns. While we use the test statistics (2.3) when describing the Stepdown method, we note that this method applies to a broader family of test statistics. In Appendix 2.C, we discuss in detail one alternative to the test statistic in (2.3) that can dramatically reduce up the computation complexity of the Stepdown method. However, we defer this to the appendix because in our specific problem setting of testing equality of covariances, it does not seem to perform well empirically.

2.4.2 Largest quasi-clique: selecting partitions based on testing results

After applying the covariance testing with the Stepdown method described in the previous subsection, we have a subset of null hypotheses from $\mathcal{H}$ that we accepted. In this subsection, we develop a clique-based method to estimate $\mathcal{P}$, the subset of partitions that share the same covariance matrix defined in (2.2), from our accepted null hypotheses.

We conceptualize the task of selecting partitions as selecting vertices from a graph that form a dense subgraph. Let $H_{0,(i,j)}$ denote the null hypothesis that the population covariance matrices for partition i and j are equal. Let $G = (V, E)$ be an undirected graph with vertices V and edge set E such that

$$V = \{1, \ldots, r\}, \quad E = \{(i,j) : H_{0,(i,j)} \text{ is accepted by the Stepdown method}\}. \tag{2.7}$$

Since each of the $\binom{|\mathcal{P}|}{2}$ pairwise tests among the partitions in $\mathcal{P}$ satisfies the null hypotheses, the vertices corresponding to $\mathcal{P}$ would ideally form the largest clique in graph G. However, this ideal situation is unlikely to happen. Instead, due to the probabilistic nature of our theoretical guarantee in (2.6), there are likely to be a few missing edges in G among the vertices corresponding to $\mathcal{P}$. Hence, a natural task is to instead find the largest *quasi-clique*, a task that has been well-studied by the computer science community (see Tsourakakis (2014) and its references within). We say a set of k vertices form a γ-quasi-clique if there are at least $\gamma \cdot \binom{k}{2}$ edges among these k vertices for some $\gamma \in [0, 1]$. The largest γ-quasi-clique is the largest vertex set that forms a γ-quasi-clique . We justify the choice to search for this γ-quasi-clique since, by our model (2.1), the prevalent covariance matrix among the r partitions is the desired covariance matrix Σ we wish to estimate. Here, γ is an additional tuning parameter, but we set $\gamma = 0.95$ by default throughout this entire paper.

Unfortunately, many algorithms that could be used to find the largest γ-quasi-clique do not satisfy a certain monotone property in practice, which hinders their usability. Specifically, consider an algorithm $\mathcal{A}$ that takes in a graph G and outputs a vertex set, denoted by $\mathcal{A}(G)$, and for two graphs G' and G, let $G' \subseteq G$ denote that G' is a subgraph of G. We say that algorithm $\mathcal{A}$ has the *monotone property* if

$$G' \subseteq G \quad \Rightarrow \quad |\mathcal{A}(G')| \leq |\mathcal{A}(G)|, \quad \text{for any two graphs } G, G'. \tag{2.8}$$

We are not aware of such a property being important in the quasi-clique literature, but it is a natural property to inherit from the multiple testing community. That is, a multiple testing

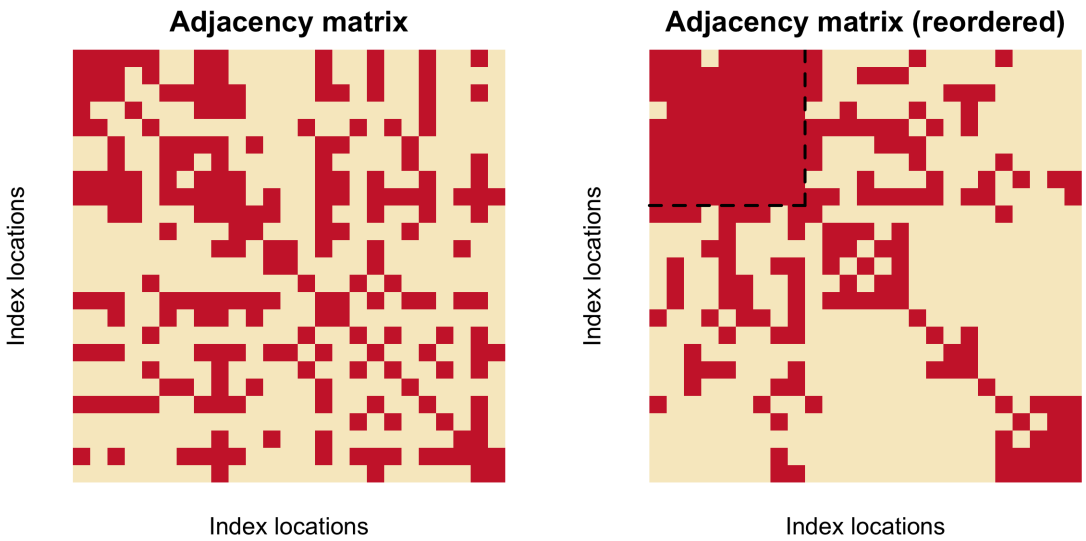


Figure 2.3: (A) Visualization of an (example) adjacency matrix that can be formed using (2.7), where the ith row from top and column from the left denotes the ith vertex. A red square in position (i, j) denotes an edge between vertex i and j, and a pale square denotes the lack of an edge. (B) Illustration of the desired goal. The rows and columns are reordered from Figure A, and the dotted box denotes the vertices that were found to form a γ-quasi-clique.

procedure has the monotone property if increasing the signal-to-noise ratio (i.e., decreasing the p-values) yields more rejections (see (Hahn, 2018) and references within). Similarly in the quasi-clique setting, it is natural to expect that increasing the signal-to-noise ratio (i.e., removing edges in G) yields less partitions selected. The monotone property is crucial in practice since it can be shown that the chosen FWER level α and the graph G defined in (2.7) have the following relation,

$$\alpha \geq \alpha' \quad \Rightarrow \quad G \subseteq G',$$

where G and G' are the graphs formed by FWER level α and α' respectively. Hence, an algorithm that does not exhibit the property in (2.8) will be fragile – using a smaller α to accept more null hypotheses might counterintuitively result in less partitions being selected. As we will demonstrate in §2.5 through simulations, many existing algorithms to find the largest quasi-clique do not satisfy the monotone property empirically. Therefore, we develop the following new algorithm to remedy this.

We describe the algorithm below. It starts by constructing a list containing all maximal cliques in the graph based on (2.7). A maximal clique is a vertex set that forms a clique but is not subset of a larger clique. The algorithm then proceeds by determining if the union

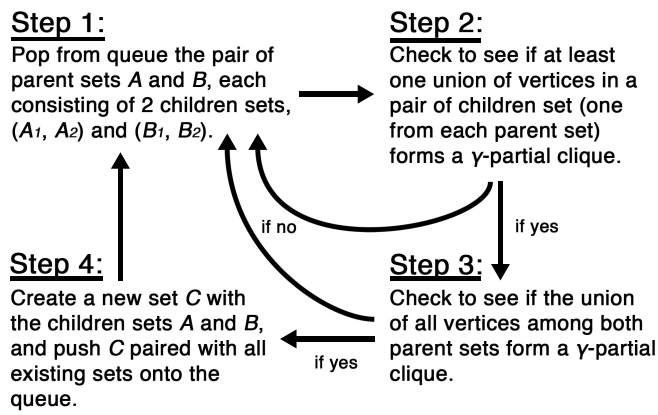


Figure 2.4: Schematic of Algorithm 4's implementation. Step 2 is able to leverage hash tables which stores previous calculations to see if the union of vertices in a pair of children sets forms a γ-quasi-clique. This has a near-constant computational complexity. This can save tremendous computational time since Step 3, which checks if the union of vertices in both parent sets form a γ-quasi-clique, has a computational complexity of $O(r^2)$.

of any two vertex sets forms a γ-quasi-clique. If so, this union of vertices is added to the list of vertex sets. The algorithm returns the largest vertex set in the list when all pairs of vertex sets are tried and no new γ-quasi-clique is found. We demonstrate in §2.5 that this algorithm exhibits the monotone property (2.8) empirically.

Algorithm 4: Clique-based selection

1. Form graph G based on (2.7).
2. Form $\mathcal{Q}$, the set of all vertex sets that form a maximal clique in G. Each vertex set is initialized with a child set equal to itself.
3. While there are vertex sets $A, B \in \mathcal{Q}$ the algorithm has not tried yet:
   a) Determine if $C = A \cup B$ forms a γ-quasi-clique in G. If so, add C as a new vertex set into $\mathcal{Q}$, with A and B as its two children sets.
4. Return the largest vertex set in $\mathcal{Q}$.

A naive implementation of the above algorithm would require checking if an exponential number of vertex set unions $C = A \cup B$ forms a γ-quasi-clique, and each check requires $O(r^2)$ operations. However, we are able to dramatically reduce the number of checks required by using the following heuristic: we only check whether the union of A and B forms a

γ-quasi-clique if the union of two children sets, one from each A and B, forms a γ-quasi-clique. This heuristic allows us to exploit previous calculations and reduce computational costs. We implement this idea by using one hash table to record which vertex sets are children of other vertex sets, and another hash table table to record if the union of two vertex sets forms a γ-quasi-clique. This idea is illustrated in Figure 2.4. Additional details on how to initialize and optionally post-process Algorithm 4 are given in Appendix 2.D.

2.5 Simulation study

We perform empirical studies to show that COBS has more power and yields a better estimation of the desired covariance matrix Σ over conventional methods as the samples among different partitions are drawn from increasingly dissimilar distributions.

Setup: We generate synthetic data in $r = 25$ partitions, where the data in each partition has $n = 15$ samples and $d = 1000$ dimensions drawn from a non-Gaussian distribution. Among these r partitions, the first group of $r_1 = 15$ partitions, second group of $r_2 = 5$ partitions and third group of $r_3 = 5$ partitions are drawn from three different nonparanormal distributions respectively (Liu et al., 2009). The goal in this simulation suite is to detect these r_1 partitions with the same covariance structure. The nonparanormal distribution is a natural candidate to model genomic data with heavier tails and multiple modes (Liu et al. (2012) and Xue and Zou (2012)), and serves to demonstrate that our methods in §2.4 does not rely on the Gaussian assumption. Formally, a random vector $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$ is drawn from a nonparanormal distribution if there exists d monotonic and differentiable functions $f_1, \ldots, f_d$ such that when applied marginally, $\boldsymbol{Z} = (f_1(X_1), \ldots, f_d(X_d)) \sim N(\mu, \Sigma)$, a Gaussian distribution with proxy mean vector μ and proxy covariance matrix[1] Σ. We provide the details of how we generate the three nonparanormal distributions in Appendix 2.E, but we highlight the key features regarding Σ below.

We construct three different proxy covariance matrices $\Sigma^{(1)}, \Sigma^{(2)}$, and $\Sigma^{(3)}$ in such a way that for a given parameter $\beta \in [0, 1]$, we construct $\Sigma^{(2)}$ and $\Sigma^{(3)}$ to be more dissimilar from $\Sigma^{(1)}$ as β increases. We highlight the key features of our constructed proxy covariance matrices here. All three proxy covariance matrices are all based on a stochastic block model (SBM), a common model used to model gene networks (Liu et al., 2018a; Funke and Becker, 2019). The first r_1 partitions are generated using proxy covariance matrix $\Sigma^{(1)}$, which is an SBM with two equally-sized clusters where the within-cluster covariance is $a = 0.9$ and the between-cluster covariance is $b = 0.1$. The second r_2 partitions are generated using proxy covariance matrix $\Sigma^{(2)}$, which is similar to $\Sigma^{(1)}$ except a and b are shrunk towards 0.5 depending on the magnitude of β. The last r_2 partitions are generated using proxy covariance matrix $\Sigma^{(3)}$, which is similar to $\Sigma^{(1)}$ except an equal fraction of variables from both clusters

[1] We emphasize "proxy" covariance matrix, for example, since the covariance of X, the random variable we sample, is not Σ.

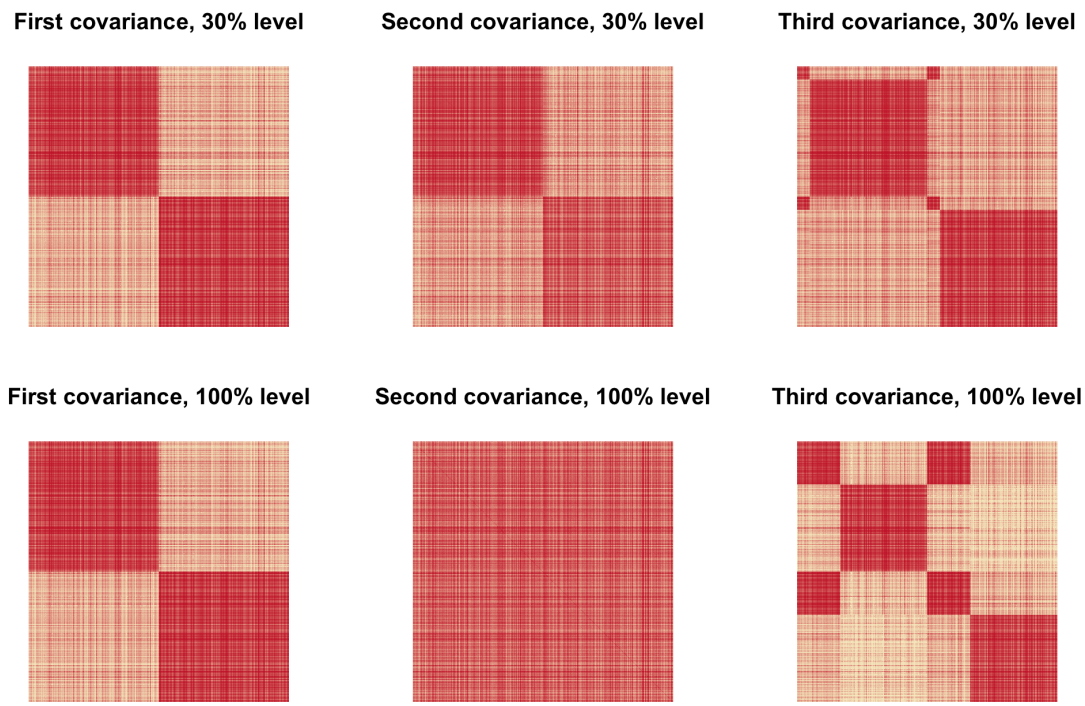


Figure 2.5: (Top row) Heatmap visualizations of the empirical covariance matrix of the three partitions, each drawn from a different nonparanormal distribution when $\beta = 0.3$. The distribution using $\Sigma^{(1)}$, $\Sigma^{(2)}$ and $\Sigma^{(3)}$ are shown as the left, middle and right plots respectively. The darker shades of red denote a higher covariance. (Bottom row) Visualizations similar to the top row except $\beta = 1$, so the dissimilarity comparing $\Sigma^{(2)}$ or $\Sigma^{(3)}$ to $\Sigma^{(1)}$ is increased.

break off to form a third cluster, depending on the magnitude of β. By generating $\Sigma^{(1)}, \Sigma^{(2)}$, and $\Sigma^{(3)}$ in this fashion, the parameter β can control the difficulty of the simulation setting – a larger β means COBS would ideally have more power in distinguishing among the first r_1 partitions from the other partitions. Figure 2.5 visualizes the resulting covariance matrices for the three nonparanormal distribution we generate in this fashion for $\beta = 0.3$ and $\beta = 1$.

Multiple testing: We use the Stepdown method described in Subsection 2.4.1 on our simulated data where $\beta = \{0, 0.3, 0.6, 1\}$ to see how the true positive rates and false positive rates vary with β. Let $\mathcal{L} = \{(i_1, j_1), (i_2, j_2), \ldots\}$ denote the returned set of partition pairs that correspond to the accepted null hypothesis. Since our goal is to find the first r_1 partitions, we define the true positive rate and false positive rate for individual hypotheses

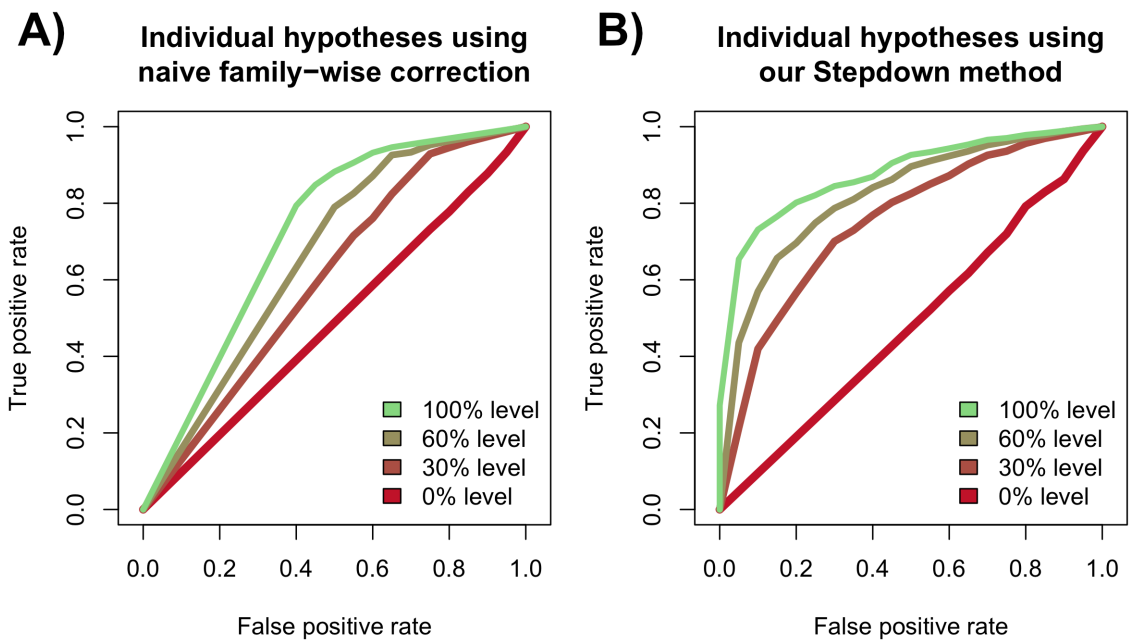


Figure 2.6: RoC curves for the accepted null hypotheses, for settings where $\beta = (0, 0.3, 0.6, 1)$, where each curve traces out the results as α varies from 0 to 1. (A) The curves resulting from using a Bonferroni correction to the $\binom{r}{2}$ individual hypothesis tests. (B) The curves resulting from using our Stepdown method.

to be

$$\text{True positive rate (TPR) for hypotheses} = \frac{\left|\left\{(i,j) \in \mathcal{L} : i \leq r_1 \text{ and } j \leq r_1\right\}\right|}{\binom{r_1}{2}},$$

$$\text{False positive rate (FDR) for hypotheses} = \frac{\left|\left\{(i,j) \in \mathcal{L} : i > r_1 \text{ or } j > r_1\right\}\right|}{\binom{r}{2} - \binom{r_1}{2}}.$$

We plot the RoC curves visualizing the TPR and FPR in Figure 2.6. Each curve traces out the mean true and false positive rate over 25 simulations as α ranges from 0 (top-right of each plot) to 1 (bottom-left of each plot), where we use 200 bootstrap trials per simulation. Figure 2.6A shows the naive analysis where we compute all $\binom{r}{2}$ p-values, one for each hypothesis test comparing two partitions, and accept hypotheses for varying levels of α after using a Bonferroni correction. Figure 2.6B shows the Stepdown method. In both plots, we see that as β increases, each method has more power. However, as we mentioned in Subsection 2.4.1, there is a considerable loss of power when comparing the

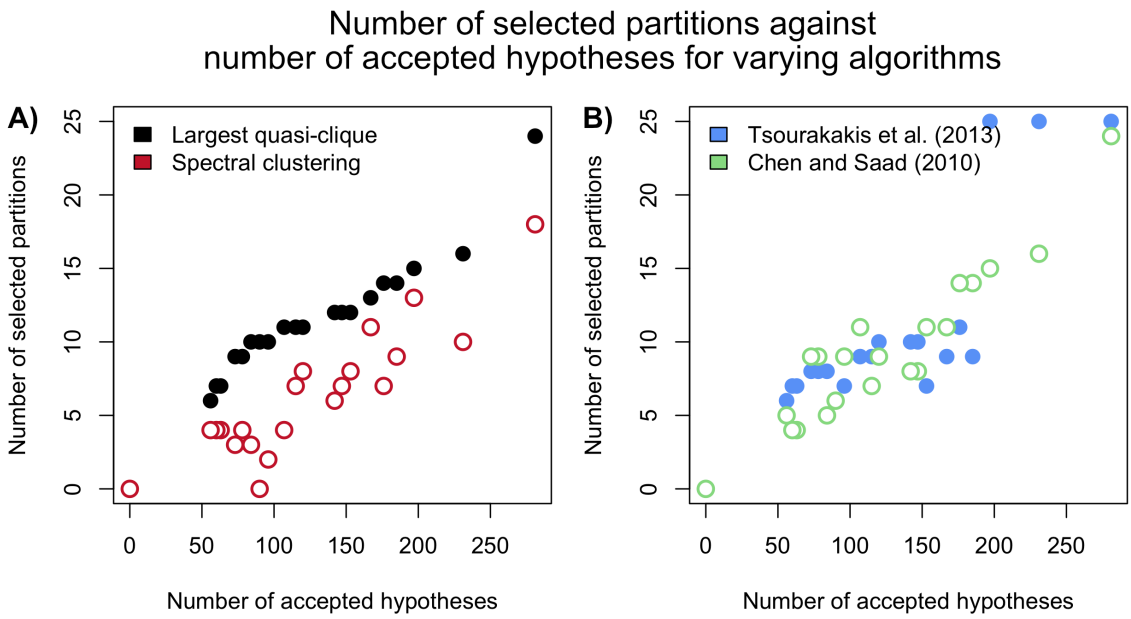


Figure 2.7: Number of selected partitions for a particular simulated dataset as the number of accepted null hypotheses varies with the FWER level α. (A) Results using our clique-based selection method developed in Subsection 2.4.2 and spectral clustering. (B) Results using the methods developed in Tsourakakis et al. (2013) and Chen and Saad (2010). See Appendix 2.D for more details of these methods.

Bonferroni correction to the Stepdown method. This is because the Bonferroni correction is too conservative when accounting for dependencies.

Partition selection: After using Stepdown, we proceed to select the partitions as in Subsection 2.4.2 to understand the monotone property and see how the true and false positive rates for partitions vary with β.

Figure 2.7 shows that three methods currently in the literature that can be used to find the largest quasi-clique in (2.7) fail the monotone property (2.8), whereas COBS succeeds. In Figure 2.7A, we compare our clique-based selection method, described in Subsection 2.4.2, against spectral clustering, a method used in network analyses designed to find highly connected vertices (Lei and Rinaldo, 2015), whereas in Figure 2.7B, two methods recently developed in the computer science community are compared (Chen and Saad (2010) and Tsourakakis et al. (2013)). These three methods are detailed in Appendix 2.D, and all the methods receive the same set of accepted null hypotheses as the FWER level α varies. Recall that since the Stepdown method accepts more hypotheses as α decreases, the graph formed

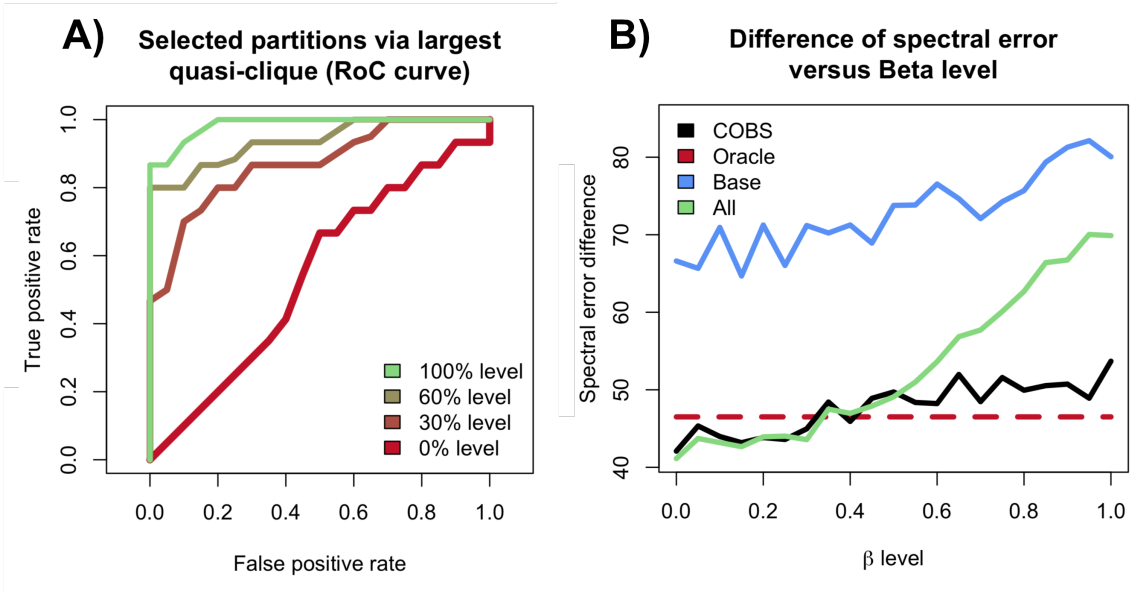


Figure 2.8: A) Similar RoC curves to Figure 2.6, but for selected partitions selected by COBS. B) The mean spectral error of each method's downstream estimated covariance matrix for varying β over 25 trials. The four methods to select partitions shown are COBS for $\alpha = 0.1$ (black), the method that selects all partitions (green), the method that selects a fixed set of 5 partitions (blue), and the method that selects exactly the partitions that contain samples drawn from a nonparanormal distribution with proxy covariance $\Sigma^{(1)}$ (red).

by (2.7) becomes denser as α increases. However, as we see in Figure 2.7, the number of partitions selected by all but our method sometimes decreases as number of accepted null hypotheses increases, hence violating the desired monotone property.

Figure 2.8A shows the RoC curves for varying β as the FWER level α varies. This figure is closely related to Figure 2.6B. We use our clique-based selection method to find the largest γ-quasi-clique for $\gamma = 0.95$. Let $\widehat{\mathcal{P}}$ denote the selected set of partitions. Similar to before, we define the TPR and FPR in this setting as

$$\text{TPR for partitions} = \frac{\left|\left\{p \in \widehat{\mathcal{P}} : p \leq r_1\right\}\right|}{r_1},$$
$$\text{FDR for partitions} = \frac{\left|\left\{p \in \widehat{\mathcal{P}} : p > r_1\right\}\right|}{r_2 + r_3}.$$

We see that the power of the COBS increases as β increases, as expected.

Covariance estimation: Finally, we show that COBS is able to improve the downstream covariance estimation compared to other approaches. To do this, we use four different methods

to select partitions and compute the empirical covariance matrix among the samples in those partitions. The first three methods resemble analyses that could be performed on the BrainSpan dataset in practice. The first method uses the COBS. The second method always selects all the partitions, which resembles using all the partitions in the BrainSpan dataset. The third method always selects the same 5 partitions – 3 partitions contain samples drawn from the nonparanormal distribution with proxy covariance $\Sigma^{(1)}$, while the other 2 partitions contain samples from each of the remaining two distributions. This resembles previous work (Liu et al., 2015) that consider only partitions in Window 1B. For comparison, the last method resembles an oracle that selects exactly the r_1 partitions containing samples drawn the nonparanormal distribution with proxy covariance $\Sigma^{(1)}$.

Figure 2.8B shows that our partition selection procedure performs almost as well as the oracle method over varying β level. Notice that for low β, COBS and the method using all partitions yield a smaller spectral error than the oracle method. This is because for low β, the covariance matrices $\Sigma^{(1)}$, $\Sigma^{(2)}$, and $\Sigma^{(3)}$ are almost indistinguishable. However, as β increases, the dissimilarities among $\Sigma^{(1)}$, $\Sigma^{(2)}$, and $\Sigma^{(3)}$ grow. This means methods that do not adaptively choose which partitions to select become increasingly worse. However, our procedure remains competitive, performing almost as if it knew which partitions contain samples drawn the nonparanormal distribution with proxy covariance $\Sigma^{(1)}$. Additional simulations that go beyond the results in this section are deferred to Appendix 2.F.

2.6 Application on BrainSpan study

We demonstrate the utility of COBS by applying it within the DAWN framework established in Liu et al. (2015). Specifically, in this section, we ask two questions. First, does COBS select reasonable partitions within the BrainSpan data, given our current scientific understanding outlined in §2.2? Second, does using COBS within the DAWN framework lead to a more meaningful gene co-expression network that can implicate genes using a "guilt-by-association" strategy?

Here, we discuss the different datasets relevant to the analysis in this section. DAWN relies on two types of data to identify risk genes: gene expression data to estimate a gene co-expression network and genetic risk scores to implicate genes associated with ASD. For the former, we use the BrainSpan microarray dataset (Kang et al., 2011), which has been the primary focus of this article so far. For the latter, we use the TADA scores published in De Rubeis et al. (2014) which are p-values, one for each gene, resulting from a test for marginal associations with ASD based on rare genetic variations and mutations.[2] For enrichment analysis, we use a third dataset consisting of TADA scores from Satterstrom et al. (2020). We use this third dataset only to assess the quality of our findings, and these TADA

[2]TADA stands for Transmission and De novo association (He et al., 2013).

scores are derived as in De Rubeis et al. (2014), but include additional data assimilated since 2014. Relying on a later "data freeze," this 2020 study has greater power to detect risk genes compared to the 2014 study: the two studies report 102 and 33 risk genes, respectively, with FDR cutoff of 10%. Additional details of our analysis in this section can be found in Appendix 2.G.

2.6.1 Gene screening

We first preprocess the BrainSpan data by determining which genes to include in our analysis. This is necessary since there are over 13,939 genes in the BrainSpan dataset, most of which are probably not correlated with any likely risk genes. Including such genes increases the computationally cost and is not informative for our purposes. Hence, we adopt a similar screening procedure as in Liu et al. (2015), which involves first selecting genes with high TADA scores based on De Rubeis et al. (2014), and then selecting all genes with a high Pearson correlation in magnitude with any of the aforementioned genes within the BrainSpan dataset. We select a total of 3,500 genes to be used throughout the remainder of this analysis.

2.6.2 Partition selection

Motivated by the findings in Willsey et al. (2013), we analyze the BrainSpan dataset using COBS to find many partitions that are homogeneous with most partitions in Window 1B (Figure 2.1). We use the Stepdown method with 200 bootstrap trials and FWER level $\alpha = 0.1$. This simultaneously finds which null hypotheses are accepted among the $\binom{125}{2}$ hypotheses tested. Based on these results, we select the partitions that form the largest γ-quasi-clique for $\gamma = 0.95$.

We visualize the results of the Stepdown method in Figure 2.9, illustrating that COBS finds 24 partitions which have statistically indistinguishable covariance matrices, 7 of which are in Window 1B. We form the graph G based on the accepted null hypotheses, as described in (2.7). Figure 2.9A shows the full graph with all 125 nodes, while Figure 2.9B shows the connected component of G as an adjacency matrix. We can see that the 24 partitions we select, which contain 272 microarray samples, correspond to 24 nodes in G that form a dense quasi-clique.

We visualize the proportion of selected partitions per window in the BrainSpan dataset in Figure 2.10A to demonstrate that our findings are consistent with the findings in Willsey et al. (2013). As mentioned in §2.2, Willsey et al. (2013) find that partitions in Window 1B are mostly homogeneous and are enriched for tightly clustered risk genes. The authors also found that, on average, gene expression varies smoothly across developmental periods, meaning there is greater correlation between the gene expressions belonging to adjacent developmental windows. The authors also estimate a hierarchical clustering among the four brain regions. Indeed, our results match these finding. We select a large proportion of partitions in Window 1B, and the proportion of selected partitions smoothly decreases as

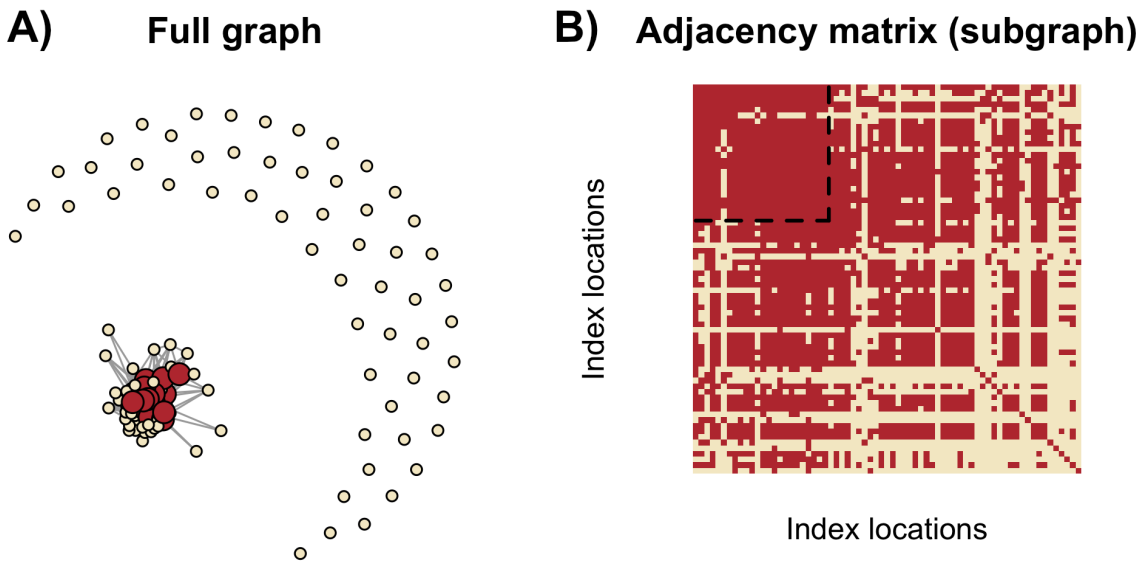


Figure 2.9: (A) The graph G containing all 125 nodes. Red nodes correspond to the 24 selected partitions, while pale nodes correspond to partitions not selected. (B) The adjacency matrix of a connect component of G, where each row and corresponding column represents a different node, similar to Figure 2.3. A red pixel corresponds to an edge between two nodes, while a pale pixel represents no edge.

the window representing older developmental periods as well as brain regions become more dissimilar to Window 1B.

Lastly, we apply the same diagnostic as in §2.3 to show in Figure 2.10B that the 272 samples within our 24 selected partitions are much more homogeneous than the 107 samples among the 10 partitions in Window 1B. The p-values we obtain after 250 divisions are much closer to uniform that those shown in Figure 2.2.

2.6.3 Overview of DAWN framework

As alluded to in §2.1, DAWN estimates a gene co-expression network using the microarray partitions to boost the power of the TADA scores using a "guilt-by-association" strategy. Figure 2.11 illustrates this procedure as a flowchart. The first step uses COBS to select 24 partitions from the BrainSpan dataset, as stated in the previous subsection. In the second step, DAWN estimates a Gaussian graphical model via neighborhood selection (Meinshausen and Bühlmann, 2006) from the 272 samples in these partitions to represent the gene co-expression network. In the third step, DAWN implicates risk genes via a Hidden Markov random field (HMRF) model by combining the Gaussian graphical model with the TADA

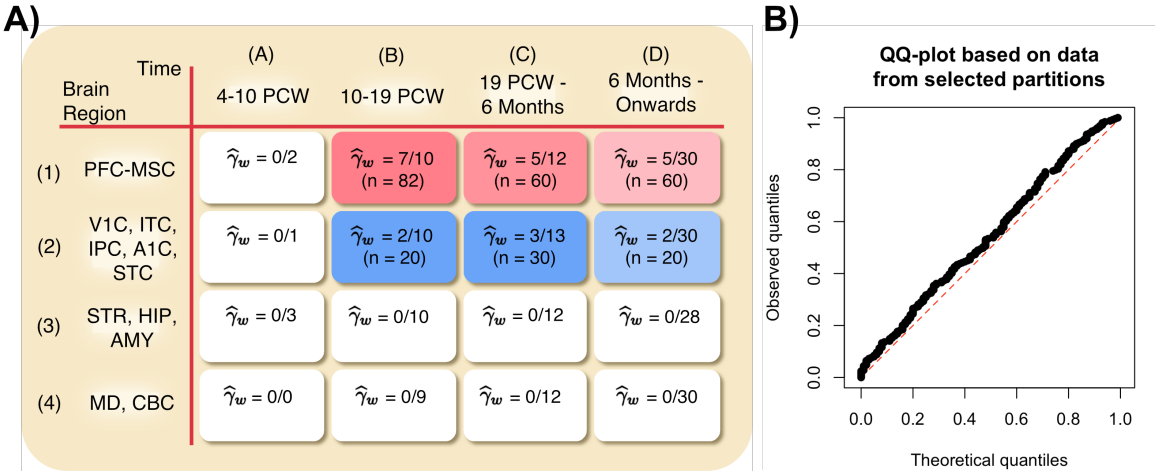


Figure 2.10: (A) The number of partitions and samples (n) selected within each window. Partitions from 6 different windows are chosen, and the estimated γ_w is empirical fraction of selected partitions within each window. The more vibrant colors display a higher value of $\widehat{\gamma}_w$. (B) A QQ-plot of the 250 p-values generated when applying our diagnostic to the 24 selected partitions, similar to Figure 2.2. While these p-values are slightly left-skewed, they suggest that the selected partitions are more homogeneous when compared to their counterparts shown in Figure 2.2.

scores. The details are in Liu et al. (2015), but in short, this procedure assumes a mixture model of the TADA scores between risk genes and non-risk genes, and the probability that a gene is a risk gene depends on the graph structure. An EM algorithm is used to estimate the parameters of this HMRF model, after which a Bayesian FDR procedure (Muller et al., 2006) is used on the estimated posterior probabilities of being a risk gene to output the final set of estimated risk genes. The methodology in the second and third step are the same as those in Liu et al. (2015), as we wish to compare only different ways to perform the first step.

2.6.4 Investigation on gene network and risk genes

In this subsection, we compare the DAWN analysis using the 24 partitions selected by COBS (i.e., the "COBS analysis") to using the 10 partitions in Window 1B originally used in Liu et al. (2015) (i.e., the "Window 1B analysis") to show how COBS improves the estimated gene network.

Closeness of genes within co-expression network. We demonstrate that the 102 genes detected by the newer TADA scores (Satterstrom et al., 2020) are roughly 10%-30% closer to the 33 genes detected by the older TADA scores (De Rubeis et al., 2014) in the gene network estimated in the COBS analysis than in the Window 1B analysis. This suggests

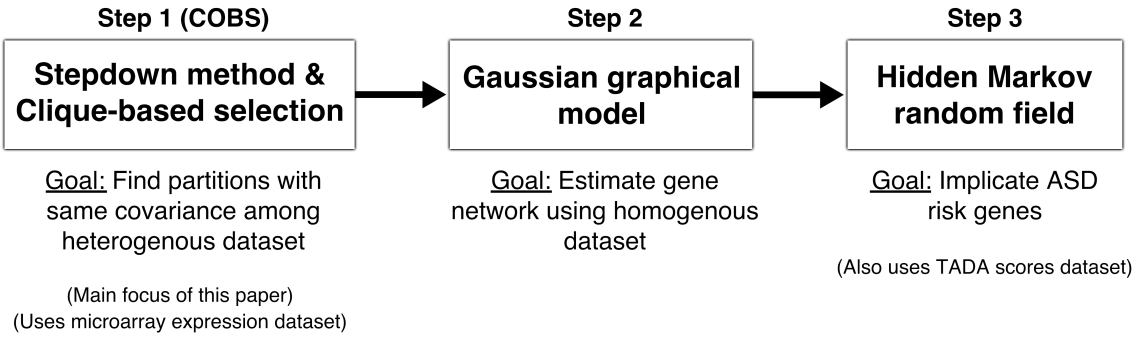


Figure 2.11: Flowchart of how COBS (Stepdown method and clique-based selection method) is used downstream to find risk genes within the DAWN framework. Step 2 and 3 are taken directly from Liu et al. (2015).

that the COBS analysis estimates a more useful gene network, because when future TADA scores are published after Satterstrom et al. (2020), the next wave of detected risk genes are more likely to also be closer to the original risk genes detected in De Rubeis et al. (2014). We defer the details to Appendix 2.G, but highlight the procedure to derive this result here. Effectively comparing the distances between genes in a network is a difficult problem since the estimated gene networks in the COBS and Window 1B analyses have different number of edges. In addition, current research has suggested that natural candidates such as the shortest-path distance or the commute distance do not accurately capture the graph topology (Alamgir and Von Luxburg (2012) and Von Luxburg et al. (2014)). Hence, we use two different distance metrics to measure the closeness of two sets of genes that potentially overcome this problem. The first is using the path distance via the minimum spanning tree, and the second is using the Euclidean distance via the graph root embedding (Lei, 2018). Using either of these metrics lead to the same conclusion.

Enrichment analysis. We demonstrate that COBS improves DAWN's ability to predict risk genes based on the newer TADA scores (Satterstrom et al., 2020) when utilizing the older TADA scores (De Rubeis et al., 2014) as input. Specifically, the COBS analysis and the Window 1B analysis implicate 209 and 249 risk genes respectively an FDR cutoff of 10%, respectively. The risk genes implicated in the COBS analysis have a better enrichment for the 102 genes detected using the newer TADA scores (Satterstrom et al., 2020): 18.8% (COBS analysis) versus 14.6% (Window 1B analysis). We note that genes implicated by DAWN but not by the TADA scores are not considered false positives. In fact, He et al. (2013) suggests that there are upwards of 500 to 1000 genes that increase risk for ASD. Hence, we are unlikely to detect all of the true risk genes based on tests that rely on rare genetic variation alone.

Robustness to γ. We additionally verify the robustness of the above enrichment results to the parameter γ. Recall that γ controls the density of the edges in the quasi-clique, as introduced in Subsection 2.4.2, and we typically set $\gamma = 0.95$ by default. When we re-run the entire analysis with different values of γ varying from 0.85 to 0.97 at intervals of 0.01, we obtain 13 different sets of estimated risk genes. We stop at $\gamma = 0.97$ since larger values result in no partitions selected outside of Window 1B. When we intersect all 13 sets of risk genes together, we find that 144 risk genes are implicated regardless of the value of γ, of which 22.9% are in the list of 102 risk genes found using only the newer TADA scores (Satterstrom et al., 2020). This is a promising result, as it demonstrates that the implicated risk genes in the COBS analysis are more enriched than those in the Window 1B analysis for a wide range of γ.

2.7 Conclusion and discussions

In this article, we develop COBS to select many partitions with statistically indistinguishable covariance matrices in order to better estimate graphical models for ASD risk gene detection. Our procedure first applies a Stepdown method to simultaneously test all $\binom{r}{2}$ hypotheses, each testing whether or not a pair of partitions share the same population covariance matrix. The Stepdown method is critical since it can account for the dependencies among all $\binom{r}{2}$ hypotheses via bootstrapping the joint null distribution. Then, our procedure uses a clique-based selection method to select the partitions based on the accepted null hypotheses. The novelty in this latter method is its ability to preserve monotonicity, a property stating that less partitions should be selected as the number of accepted null hypotheses is smaller. We demonstrate empirically that the COBS achieves this property while common methods such as spectral clustering do not. When we apply COBS to the BrainSpan dataset, we find scientifically meaningful partitions based on the results in Willsey et al. (2013). We also find that COBS aids in clustering the risk genes detected in Satterstrom et al. (2020) closer to the risk genes detected in (De Rubeis et al., 2014) within the estimated gene co-expression network and in getting a better enrichment in implicated risk genes via the DAWN analysis.

The theoretical role of the FWER level α is not well understood mathematically. Specifically, while (2.6) provides a theoretical guarantee about the set of null hypothesis accepted, we would like to prove a theoretical guarantee about the set of selected partitions $\widehat{\mathcal{P}}$. Towards this end, we suspect that with some modification to COBS, closed testing offers a promising theoretical framework (see Dobriban (2018) and references within). This will be investigated in future work.

COBS is applied directly to help implicate risk genes for ASD, but this line of work has broader implications in genetics. Due to the improvement of high throughput technologies, it has become increasingly accessible to gather large amounts of gene expression data. This includes both microarray and RNA sequencing data. However, as we have seen in this article,

gene expression patterns can vary wildly among different tissues. Hence, it is challenging to select samples that are relevant for specific scientific tasks. Beyond analyzing brain tissues, Greene et al. (2015) develop procedures to select relevant samples amongst a corpus of microarray expression data to estimate gene co-expression networks for different tissue types. While Greene et al. (2015) does not motivate their method from a statistical model, our work provides a possible statistical direction for this research field to move towards.

Acknowledgments: We thank Bernie Devlin and Lambertus Klei for the insightful discussions about our analysis and results. We thank Li Liu and Ercument Cicek for providing the code used in Liu et al. (2015) to build off of. We also thank the anonymous reviewers for helpful suggestions on how to restructure the simulations and analyses.

2.A Code and dataset

The `R` code for replicating all analyses and figures in this article are hosted on GitHub in the repository `https://github.com/linnylin92/covarianceSelection`. The three major datasets used in this article are also included in the repository. The first dataset is the BrainSpan microarray samples collected by (Kang et al., 2011). While the original dataset is publicly available on GEO (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE25219`), we provide a locally preprocessed dataset, which was created to be amendable for our analysis in `R`. The second dataset is the older TADA scores (De Rubeis et al., 2014). The third dataset is the list of 102 risk genes detected using the newer TADA scores (Satterstrom et al., 2020).

2.B Brain region details

There are four primary brain regions, each containing smaller subregions.

- **PFC-MSC**: The prefrontal cortex (PFC) and primary motor-somatosensory cortex (MSC) consist of six smaller regions: primary motor cortex, primary somatosensory cortex, ventral prefrontal cortex, medial prefrontal cortex, dorsal prefrontal cortex and orbital prefrontal cortex.
- **V1C, ITC, IPC, A1C, STC**: A region consisting of the primary visual cortex (V1C), inferior temporal cortex (ITC), primary auditory cortex (A1C), and superior temporal cortex (STC).
- **STR, HIP, AMY**: A region consisting of the stratum (STR), hippocampal anlage or hippocampus (HIP) and amygdala (AMY).
- **MD, CBC**: A region consisting of the mediodorsal nucleus of the thalamus (MD) and the cerebellar cortex (CD).

2.C Extension to the Stepdown method

One of the largest drawbacks of the Stepdown method lies in its intensive computational cost. For r partitions, at most $\binom{r}{2}$ bootstrap statistics need to be computed in each bootstrap trial, each requiring a computational cost of $O(d^2 \cdot n_p)$. In this section, we develop a computational extension to the Stepdown method that yields a more computationally efficient algorithm as long as the test statistic $\widehat{T}$ satisfies the *triangle inequality* and the number of variables d is large. That is, for any bootstrap trial b and for any partitions i, j and k, we require that the bootstrap statistics satisfy

$$\widehat{T}^{(b)}_{(i,k)} \leq \widehat{T}^{(b)}_{(i,j)} + \widehat{T}^{(b)}_{(j,k)}. \tag{2.9}$$

This property can potentially save expensive calculations when calculating (2.5) in Algorithm 2 by reducing the number of bootstrap statistics we need to explicitly calculate. Since we only care about the maximum bootstrap statistic $\widehat{T}^{(b)}$ in each trial, the triangle inequality gives an upper bound on the bootstrap statistic $\widehat{T}^{(b)}_{(i,k)}$ between partitions i and k, leveraging bootstrap statistics already calculated within a specific bootstrap trial. As we sequentially iterate through all pairs of partitions (i, k), if the upper bound for $\widehat{T}^{(b)}_{(i,k)}$ is smaller than the current maximum bootstrap statistic within a specific bootstrap trial b, we do not need to explicitly compute $\widehat{T}^{(b)}_{(i,k)}$.

Unfortunately, the test statistic (2.3) described in Subsection 2.3.1 originally from Chang et al. (2017) does not satisfy the triangle inequality (2.9). Hence, we consider a new test statistic defined as

$$\widehat{T} = \max_{ij} \left(\widehat{t}_{ij}\right) \quad \text{where } \widehat{t}_{ij} = \left|\widehat{\sigma}_{X,ij} - \widehat{\sigma}_{Y,ij}\right|, \quad i, j \in 1, \ldots, d, \tag{2.10}$$

and we make a similar modification for its bootstrap counterpart, $\widehat{T}^{(b)}$. It can easily be shown that the above bootstrap statistics satisfies the desired triangle inequality. Additionally, using the techniques in Chernozhukov et al. (2013), it can be proven that this test statistic will still yield a hypothesis test with asymptotic $1 - \alpha$ coverage under the null, analogous to (2.6). We will call the Stepdown procedure that uses (2.10) the "Accelerated Stepdown" procedure.

To formalize how to take advantage of this triangle inequality property, we describe a subroutine that leverages this property to compute $\widehat{T}^{(b)}$ in (2.5) by representing the individual bootstrap statistics $\widehat{T}^{(b)}_{(i,j)}$ as weighted edges in a graph. The algorithm uses Dijsktra's algorithm to find the shortest path between vertices. This implicitly computes the upper bound in the bootstrap statistic between two partitions using the triangle inequality. This algorithm can provide substantial improvement in computational speed by leveraging the fact that determining the shortest path on a fully-dense graph has a computational complexity of $O(r^2)$, whereas computing $T^{(b)}_{(i,j)}$ has a computational complexity of $O(d^2 \cdot n_p)$.

Algorithm 3: Distance metric-based procedure to compute $\widehat{T}^{(b)}$

1. Form graph $G = (V, E)$ with r nodes and all $\binom{r}{2}$ edges, and initialize each edge to have a weight of infinity.

2. Arbitrarily construct a spanning tree $\mathcal{T}$ and compute all $\widehat{T}^{(b)}_{(i,j)}$'s corresponding to edges $(i, j) \in \mathcal{T}$. Record $z = \max_{(i,j)\in\mathcal{T}} \widehat{T}^{(b)}_{(i,j)}$.

3. Construct a set of edges $\mathcal{S} = \mathcal{L}Ackslash\mathcal{T}$ which represents the bootstrap statistics between specific pairs of partitions that have yet to be computed.

4. While $\mathcal{S}$ is not empty:

   a) Arbitrarily select an edge $(i, j) \in \mathcal{S}$ and remove it from $\mathcal{S}$. Compute the shortest-path distance from vertex i to j in G.

   b) If the shortest-path distance is larger than z, update the edge (i, j) to have weight $\widehat{T}^{(b)}_{(i,j)}$, and update z to be $\max(z, \widehat{T}^{(b)}_{(i,j)})$.

5. Return z.

As we will see in §2.F, while (2.10) can take advantage of this computational speedup, it yields a much less powerful test when compared to test using (2.3). This is intuitive, as (2.10) does not normalize by the sum of the empirical variances, unlike (2.3). Hence, we do not use the Accelerated Stepdown procedure within COBS when analyzing the BrainSpan dataset in this paper. However, we believe there are potentially other settings outside of covariance testing where this computational speedup idea can be utilized more effectively. We leave this as direction for future work.

2.D Details of algorithms to find quasi-cliques

The first subsection remarks on possible extensions to the clique-based selection method described in Algorithm 4. The second subsection describes the three other algorithm used in §2.5 for us to compare against. Throughout this section, for a generic graph G, we use V to denote the set of vertices in G, G_S to denote a subgraph formed by a vertex set $S \subseteq V$, and $E(G)$ to denote the number of edges in G.

2.D.1 Extensions to clique-based selection method

We mention two extensions to clique-based selection method (Algorithm 4) that can be useful in practice.

- **Initializing algorithm around a desired set of vertices**: In certain cases, the user would want the γ-quasi-clique to be initialized around a desired subset of vertices in $G = (V, E)$. For instance, in our setting, since Liu et al. (2015) applies DAWN to the 10 partitions in Window 1B, it is natural for us to encourage COBS to select as many partitions in Window 1B as possible to enable a meaningful comparison.

  To resolve this, first, we run Algorithm 4 at the desired level γ on G_S. This would output a subset of vertices $S_{\text{core}} \subseteq S$ that form the largest γ-quasi-clique in G_S. Then, we run Algorithm 4 at the same level γ on the full graph G but perform an additional operation after (2.): after $\mathcal{Q}$ is initialized with all maximal cliques in G, we check each vertex set $A \in \mathcal{Q}$ if $A \cup S_{\text{core}}$ forms a γ-quasi-clique. If yes, we replace A with $A \cup S_{\text{core}}$ in $\mathcal{Q}$. If not, we remove A from $\mathcal{Q}$. The algorithm then proceeds to (3.) as usual. By applying this simple change, we are ensured the returned vertex set by Algorithm 4 contains S_{core}.

- **Post-processing the returned vertex set**: In certain cases, the returned vertex set of Algorithm 4 has a few vertices with a very low degree when compared to the other vertices. To resolve this, we post-process this vertex set by removing vertices that are connected to less than half the other vertices in the returned set.

In this paper, we use the initialization extension only when analyzing the BrainSpan dataset in §2.6, where we initialize the largest quasi-clique around the 10 partitions in Window 1B.

2.D.2 Overview of other algorithms

We overview the three algorithms introduced in §2.5 that are designed to find large quasi-cliques.

- Chen and Saad (2010): This algorithm recursively splits a graph G into two in a hierarchical-clustering type approach with respect to a carefully constructed weight matrix. This forms a tree-type data structure, and then the algorithm scans the tree in a breath-first-search type fashion for the largest subgraph with an edge density larger than γ.

- Tsourakakis et al. (2013): This algorithm performs a local search by adding vertices mypoically and then removing vertices occasionally until no more myopic improvements can be made. Specifically, it first initializes the set S of vertices to contain a vertex that maximizes the ratio between the number of triangles and the degree, and includes all of the neighbors of said vertex. Then algorithm iteratively tries to incrementally improve the $f_\gamma(S) = E(G_S) - \gamma\binom{|S|}{2}$ as much as possible by adding neighbors of S. When it is no longer able to improve $f_\gamma(S)$, the algorithm tries removing a vertex from

S to improve $f_\gamma(S)$. The algorithm then iterates between such adding and removing vertices from S for a fixed number of iterations.

- Spectral clustering: While many different community detection methods for random graphs now exist (for example, see Abbe (2017) and Athreya et al. (2017) and the references within), we choose spectral clustering as described in Lei and Rinaldo (2015) as a prototypical example of how many of such methods fail to demonstrate the monotone property as described in Subsection 2.4.2. Specifically, this method applies K-means clustering to the top K eigenvectors of the adjacency matrix, where K is a tuning parameter to specify. To find large quasi-cliques, we iteratively try spectral clustering for a range of K's (i.e., $K = 2, \ldots, 5$), and for each detected cluster in any of the estimated clusterings, we compute if the corresponding vertices of said cluster forms a γ-quasi-clique. If any γ-quasi-clique is found, we return the largest γ-quasi-clique discovered in this fashion.

2.E Formal description of simulation setup

We say a multivariate vector $X \in \mathbb{R}^d$ is distributed based a nonparanormal distribution with proxy mean vector μ, proxy covariance matrix Σ, and monotonic and differentiable functions $f_1, \ldots, f_d$ if the density of X is

$$p(X) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\Big\{ -\frac{1}{2}\big(f(X)-\mu\big)^\top \Sigma \big(f(X)-\mu\big)\Big\} \prod_{j=1}^{d} \big|f_j'(x_j)\big|, \tag{2.11}$$

where $f(X) = (f_1(x_1), \ldots, f_d(x_d))$. This is defined in Liu et al. (2009). In our simulation suite, we set $\mu = 0$. Let this distribution be denoted as $\mathrm{NPN}(0, \Sigma, f)$. In the next two subsections, we formalize the details of Σ and $f_1, \ldots, f_d$.

2.E.1 Details on proxy covariance matrices Σ

The following three bullet points detail the construction of $\Sigma^{(1)}$, $\Sigma^{(2)}$ and $\Sigma^{(3)}$ respectively. As mentioned in §2.5, $\beta \in [0, 1]$ is a user-defined parameter that controls the dissimilarity among these three matrices.

- **Construction of $\Sigma^{(1)}$**: As mentioned in §2.5, $\Sigma^{(1)} \in \mathbb{R}_+^{d\times d}$ follows an SBM with two equally-sized clusters. Specifically, the first cluster contains indices $1, \ldots, \lfloor d/2 \rfloor$ and the second cluster contains indices $\lfloor d/2 \rfloor + 1, \ldots, d$. Then, we construct $\Sigma^{(1)}$ where

$$\Sigma_{ij}^{(1)} = \begin{cases} 1 & \text{if } i = j, \\ a & \text{if } i \neq j,\ i \text{ is in the same cluster as } j, \\ b & \text{if } i \neq j,\ i \text{ is not in the same cluster as } j, \end{cases} \tag{2.12}$$

for all $i, j \in 1, \ldots, d$ and $a = 0.9$ and $b = 0.1$.

- **Construction of** $\Sigma^{(2)}$: $\Sigma^{(2)}$ is constructed the same as $\Sigma^{(1)}$, except

$$a = 0.9 - \beta \cdot 0.4, \quad \text{and} \quad b = 0.1 + \beta \cdot 0.4.$$

  When $\beta = 1$, this means that $\Sigma^{(2)}$ is a matrix with 0.5 everywhere along the off-diagonal.

- **Construction of** $\Sigma^{(3)}$: $\Sigma^{(3)}$ is constructed in a similar way to $\Sigma^{(1)}$, except there are three clusters. The first cluster contains indices $1, \ldots, \lfloor \beta \cdot d/6 \rfloor, \lfloor d/2 \rfloor + 1, \ldots, \lfloor d/2 + \beta \cdot d/6 \rfloor$. The second cluster contains indices $\lfloor \beta \cdot d/6 \rfloor + 1, \ldots, \lfloor d/2 \rfloor$. The third cluster contains indices $\lfloor d/2 + \beta \cdot d/6 \rfloor + 1, \ldots, d$. Observe that this partitions $1, \ldots, d$, and when $\beta = 1$, this results in three clusters of the roughly the same size. We then construct $\Sigma^{(3)}$ using (2.12) but using these three clusters.

2.E.2 Details on functions $f_1, \ldots, f_d$

At a high-level, these functions $f_1, \ldots, f_d$ ensure that these marginal distributions of our sampled nonparanormal random variables are similar to the marginal distributions of the BrainSpan data. These marginal distributions are constructed in the following way. We first randomly sample d variables (i.e., genes) uniformly from the BrainSpan dataset, $\{g_1, \ldots, g_d\} \subseteq \{1, \ldots, n\}$. Next, for each j, let $\widehat{p}_{g_j}$ denote the kernel density estimate of variable g_j in the BrainSpan dataset, using the default bandwidth selection used by the `stats::density` function in `R`.

We now formalize how to construct $f_1, \ldots, f_d$. As described in Liu et al. (2009), we actually construct the inverse of these functions $f_1^{-1}, \ldots, f_d^{-1}$ as they are more amendable for sampling, which must exist since $f_1, \ldots, f_d$ are monotonic and differentiable. Recall that $\mu = 0$. We first sample a vector $\boldsymbol{z} = (z_1, \ldots, z_d)$ from a Gaussian distribution $N(0, \Sigma)$. Let $\Phi(t; P)$ denote the cumulative distribution function evaluated at t for a univariate density P. For any $j \in 1, \ldots, d$, we construct f_j^{-1} such that

$$\Phi\big(t; N(0, \Sigma_{jj})\big) = \Phi\big(f_j^{-1}(t); \widehat{p}_{g_j}\big), \quad \forall t \in \mathbb{R}.$$

That is, we construct f_j^{-1} so that z_j is at the same quantile with respect to $N(0, \Sigma_{jj})$ as $f_j^{-1}(z_j)$ is with respect to the kernel density estimate $\widehat{p}_{g_j}$. Notice that by constructing $f_1^{-1}, \ldots, f_d^{-1}$ in this fashion, each function is monotone and differentiable. We then set

$$X = \big(x_1, \ldots, x_d\big) = \big(f_1^{-1}(z_1), \ldots, f_d^{-1}(z_d)\big)$$

as one sample from the nonparanormal distribution $\mathrm{NPN}(0, \Sigma, f)$.

Notice that by introducing non-Gaussianity into our simulation suite in this fashion, we ensure that the marginal distribution of all r partitions resemble the BrainSpan dataset,

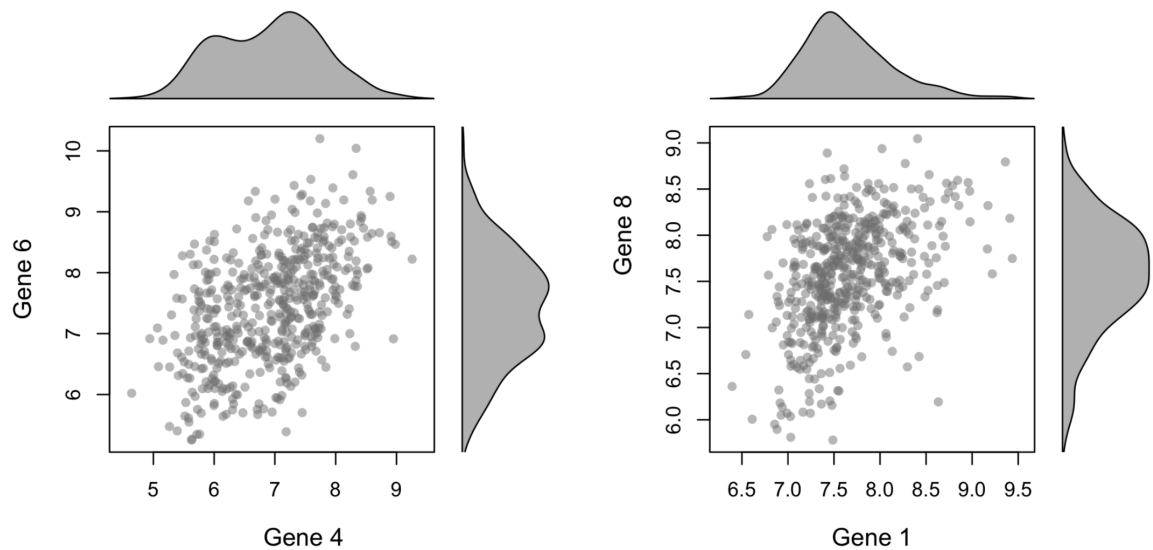


Figure 2.12: Two scatter plots of bivariate distributions sampled from the nonparanormal for $\beta = 0$. The densities shown on the top and the right of each plot represents the targeted kernel density estimates from the BrainSpan data that the nonparanormal is sampling from, captured by $f_1, \ldots, f_d$.

and also ensure that the first r_1 partitions still are drawn from the same population covariance matrix. Also, by generating data in this fashion, we are able to obtain complicated dependencies between the mean and variance, as well as observe multi-modal distributions and heavier-tailed distributions compared to the Gaussian. See Liu et al. (2009) for a more detailed discussion.

2.E.3 Example of sampled nonparanormal distribution

We provide a visual illustration of what the sampled nonparanormal distribution could look like. We sample 375 samples from $\mathrm{NPN}(0, \Sigma^{(1)}, f)$ when $\beta = 0$, and plot two of the resulting pairwise scatterplots in Figure 2.12. We can think of the 375 samples as equivalent to aggregating all $r = 25$ partitions together, each having $n = 15$ samples. These two scatterplots show that the nonparanormal can display multiple modes marginally or heavier tails.

2.F Additional simulation results

2.F.1 Covariance homogeneity diagnostic in simulation

In this section, we apply the diagnostic developed in §2.3 to the simulation suite described in §2.5. Our goal is to determine how the QQ-plots behave as the selected partitions become

less homogeneous. As done in §2.5, we consider four partition selection strategies: COBS (using $\alpha = 0.1$ and $\gamma = 0.95$), Base (which selects 3 partitions contain samples drawn from the nonparanormal distribution with proxy covariance $\Sigma^{(1)}$, while the other 2 partitions contain samples from each of the remaining two distributions), All (which selects all r partitions) and Oracle (which selects exactly the r_1 partitions containing samples drawn the nonparanormal distribution with proxy covariance $\Sigma^{(1)}$).

We see in Figure 2.13 and Figure 2.14 that the QQ-plot is a reasonable diagnostic in this simulation suite. Between these two figures, we vary β among 0, 0.3, 0.6 and 1. We notice that as β increases, the QQ-plot derived from COBS remains relative uniform, similar to that of the Oracle. When $\beta = 1$, COBS selects one erroneous partition in this particular trial shown, which results in the QQ-plot showing a deviation away from uniform. The QQ-plots derived from the Base procedure looks relative uniform when $\beta = 0$ (which is to be expected, as all r partitions share the same covariance matrix when $\beta = 0$), but quickly has QQ-plots that deviate from uniform as β increases. Note that the since the Base procedure selects only 5 partitions, there are a limited number of ways to split the partitions into two groups, which yields a limited number of points in the QQ-plot. The QQ-plots derived from the All procedure follow a similar trend as the Base procedure, but not as severe. These plots match the findings shown in Figure 2.8.

2.F.2 Simulation under Gaussian setting

While the simulations in §2.5 use nonparanormal distributions, we demonstrate that similar results hold for Gaussian distributions. This demonstrates that there is nothing particularly special about the nonparanormal or the Gaussian distribution that enable COBS to work well, and suggests COBS can work in much more general settings. Specifically, in this simuation suite, everything is the same as in §2.5, except all the functions $f_1, \ldots, f_d$ are set to be the identity function. Hence, this means that the first r_1 partitions are drawn from Gaussian distributions with covariance $\Sigma^{(1)}$, the next r_2 partitions are drawn from Gaussian distributions with covariance $\Sigma^{(2)}$, and so on.

When we use Bonferroni or the Stepdown method in this Gaussian setting, we observe ROC curves for the individual hypotheses that strongly resemble Figure 2.6. This is shown in Figure 2.15.

Similarly, when we use COBS to select partitions, the ROC curves as well as the spectral error curves strongly resemble Figure 2.8A and B. This is shown in Figure 2.16.

2.F.3 Simulation using Accelerated Stepdown

In this subsection, we apply the Accelerated Stepdown procedure described in §2.C within the COBS procedure in the simulation setting described in §2.5. Specifically, we use the test statistic (2.10) and analogous bootstrap statistics, but keep all other parts of the simulation suite the same.

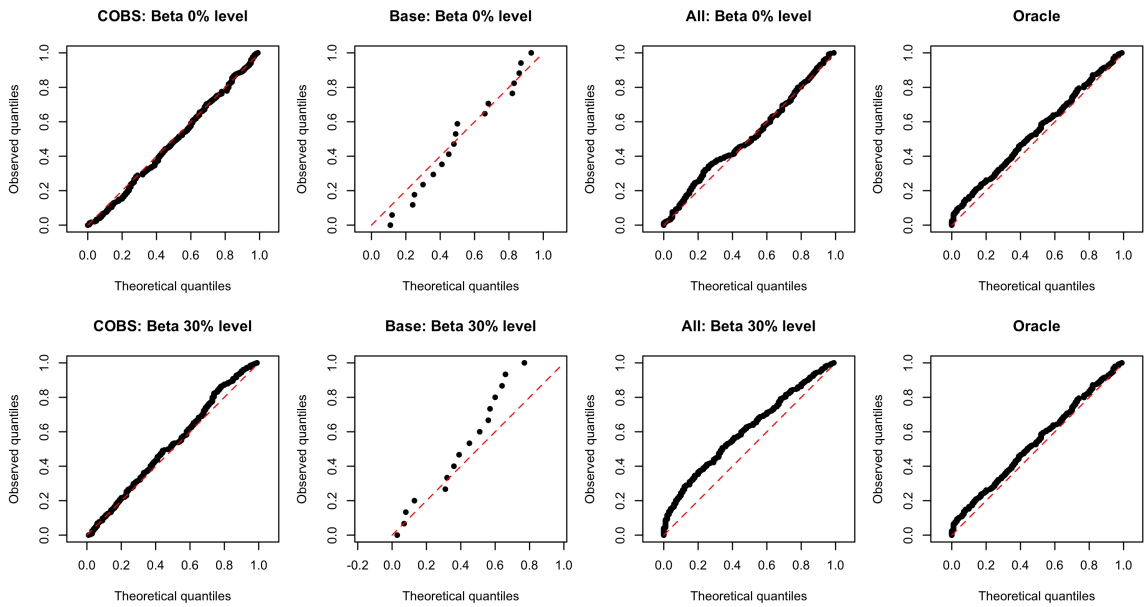


Figure 2.13: QQ-plots from the covariance homogeneity diagnostic using four different selection procedures: COBS (left-most), Base (center left), All (center right) and Oracle (right-most). The top row represents the simulation setting where $\beta = 0$, while the second row represents the simulation setting where $\beta = 0.3$. The plots are created from one instance of COBS, Base, All and Oracle procedures, and 250 trials are used within the covariance homogeneity diagnostic.

When we plot the ROC curve for the individual hypotheses in Figure 2.17, we already notice a dramatic loss of power when compared to its original counterpart using the test statistic (2.3) shown in Figure 2.6. In fact, it seems like the Bonferroni procedure has almost no power at all, even when $\beta = 1$.

Due to the loss of power for the individual hypotheses, we observe a loss of power for the selected partitions as well (Figure 2.18A) and spectral errors that strongly resemble selecting all the partitions (Figure 2.18B).

2.G Additional details on BrainSpan analysis

The first subsection describes the analysis pipeline we used throughout §2.6 in more detail. The second subsection describes the two distance metrics used in Subsection 2.6.4. The third subsection describes additional results alluded to in Subsection 2.6.4.

2.G.1 Description of analysis pipeline

We now summarize the pipeline used in §2.6 for clarity.

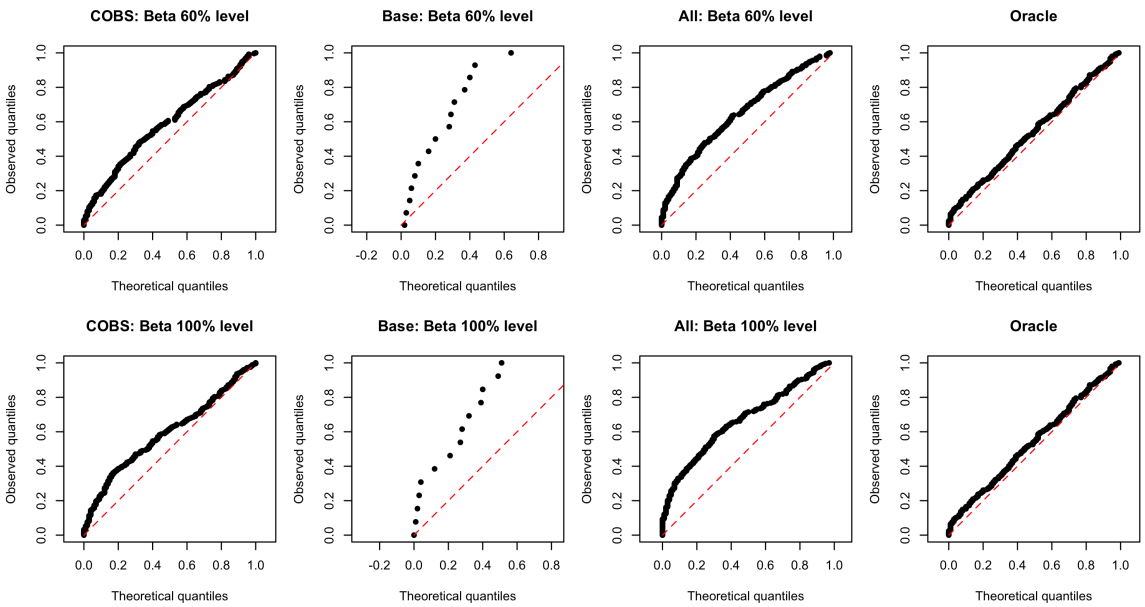


Figure 2.14: QQ-plots derived in a similar way as in Figure 2.14. However, in this plot, the top row represents the simulation setting where $\beta = 0.6$, while the second row represents the simulation setting where $\beta = 1$.

1. **Screening of genes**: This is the step described in Subsection 2.6.1, derived from Liu et al. (2015). We first select all genes whose p-value in the older TADA dataset (De Rubeis et al., 2014) is less than 0.01. Then, we rank all remaining genes by their maximum Pearson correlation in magnitude with any of the formerly selected genes in decreasing order based on the BrainSpan partitions within Window 1B aggregated. We select genes based on this ranking in order until we have selected a combined total of $d = 3500$ genes. We analyze all 125 partitions using only these d genes for the remainder of the analysis.

2. **Applying COBS**: This is the two-staged procedure we developed in this paper, detailed in §2.4. In the first stage, we apply the Stepdown procedure using $\alpha = 0.1$. In the second stage, we select the clique-based selection method where $\gamma = 0.95$, as well as using both extensions discussed in Subsection 2.D.1. This results in 24 selected partitions within the BrainSpan dataset, as detailed in Subsection 2.6.2. We then combine all the 24 selected partitions to form a dataset $\mathbb{X} \in \mathbb{R}^{n \times d}$ with $n = 272$ microarray samples and d genes to be used for the remainder of the analysis.

3. **Estimating the Gaussian graphical model**: This step is described in Subsec-

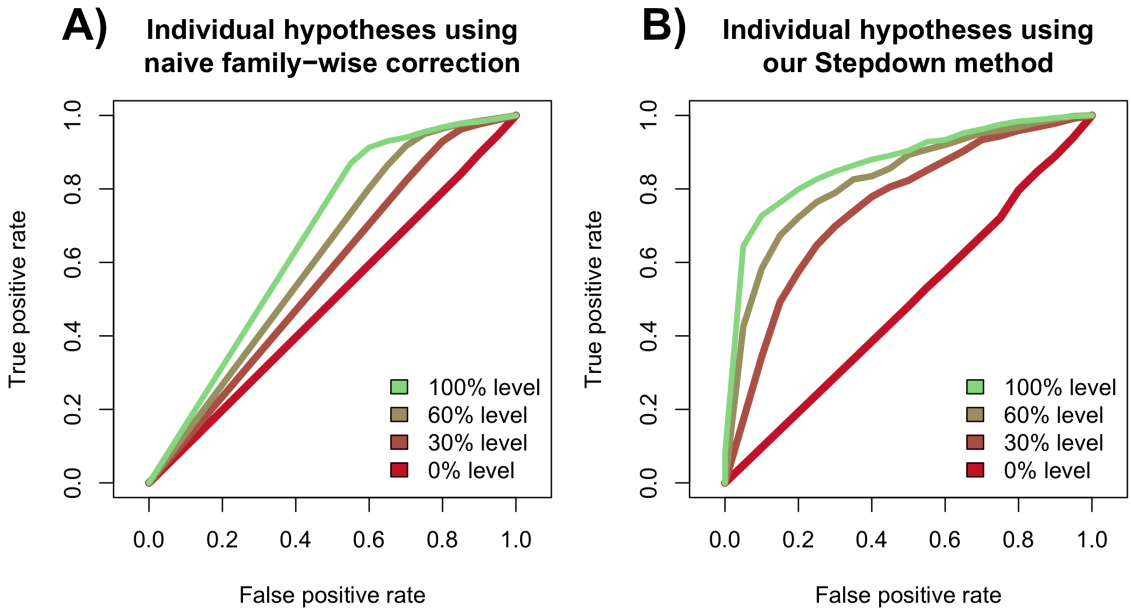


Figure 2.15: ROC curves for the hypothesis under the Gaussian setting. These plots are similar to those in Figure 2.6.

tion 2.6.3 and is the same as in Liu et al. (2015). We fit a Gaussian graphical model using neighborhood selection (Meinshausen and Bühlmann, 2006) based on $\mathbb{X}$, where the tuning parameter λ (which controls the sparsity of the graphical model) is chosen such that the resulting graph has high scale-free index as well as a comparable number of edges to the estimated graph when COBS is not used. This choice of λ is detailed at the end of this subsection. We defer the remaining estimation and computation details to Liu et al. (2015). We denote the adjacency matrix of the estimated graphical model as $\widehat{A} \in \{0,1\}^{d \times d}$.

4. **Estimating the HMRF**: This step is also described in Subsection 2.6.3 and is the same as in Liu et al. (2015). We briefly summarize this step here, as it is less common in the statistical literature. Let $\boldsymbol{Z} \in \mathbb{R}^d$ denote the Z-scores for the selected genes, derived from the TADA scores in De Rubeis et al. (2014). We model $\boldsymbol{Z}$ using a HMRF, where for each gene in $j = 1, \ldots, d$, Z_j is an i.i.d. random variable drawn from a mixture of two Gaussians,

$$Z_j \sim \mathbb{P}(I_j = 0)N(0, \sigma^2) + \mathbb{P}(I_j = 1)N(\mu, \sigma^2),$$

where $I_j \in \{0,1\}$ is an unobserved Bernoulli random variable and $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+$

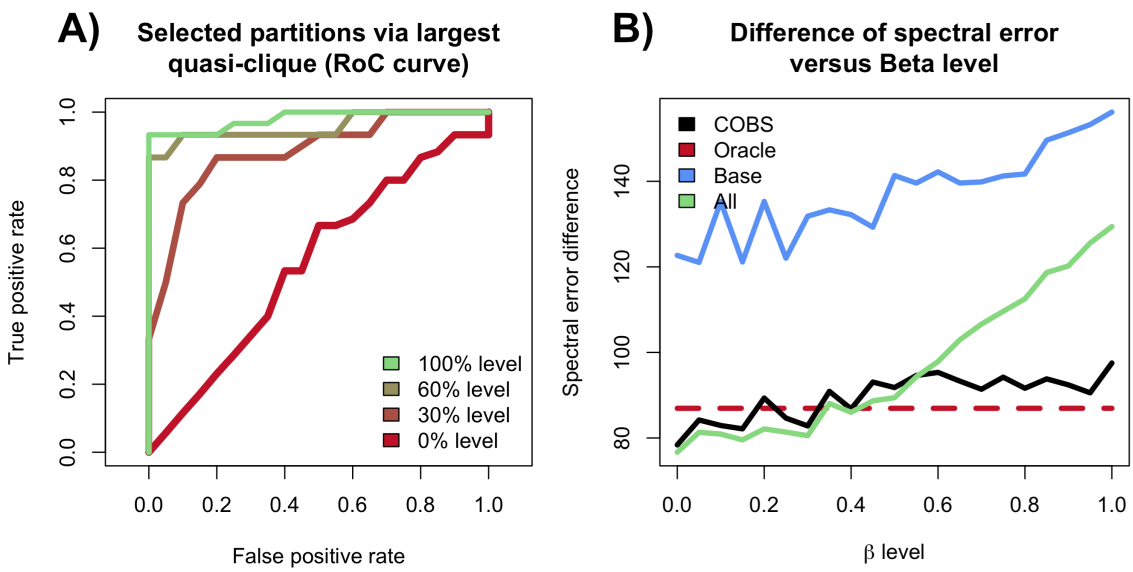


Figure 2.16: A) ROC curves for the partitions selected by COBS under the Gaussian setting. This plot is similar to Figure 2.8A. B) Mean spectral error of the estimated covariance matrix for varying β level under the Gaussian setting. This plot is similar to Figure 2.8B.

are two unknown scalars to be estimated. The first Gaussian distribution represents the Z-scores for genes that are not associated with ASD, and the second Gaussian distribution represents the Z-scores for risk genes. The distribution of entire vector $I \in \{0,1\}^d$ follows an Ising model with probability mass function,

$$\mathbb{P}(I = \eta) \propto \exp\Big(b \cdot \sum_{j=1}^{d} \eta_j + c \cdot \eta^T \widehat{A}\eta\Big),$$

for any $\eta \in \{0,1\}^d$ and two unknown scalars $b, c \in \mathbb{R}$ to be estimated. An EM algorithm is used to fit this HMRF model, and we obtain the estimated posterior probability $\widehat{p}_j = \mathbb{P}(I_j = 0|\boldsymbol{Z})$, representing the probability gene j is not a risk gene given the risk scores. We defer the estimation and computation details to Liu et al. (2015).

5. **Applying Bayesian FDR**: This step is also described in Subsection 2.6.3 and is the same as in Liu et al. (2015). We apply a procedure (Muller et al., 2006) to $\widehat{\boldsymbol{p}}$ to select a set of genes where the Bayesian FDR is controlled at level 10%. We defer the computation details to Liu et al. (2015). This results in the set of 209 detected risk genes detailed in Subsection 2.6.4.

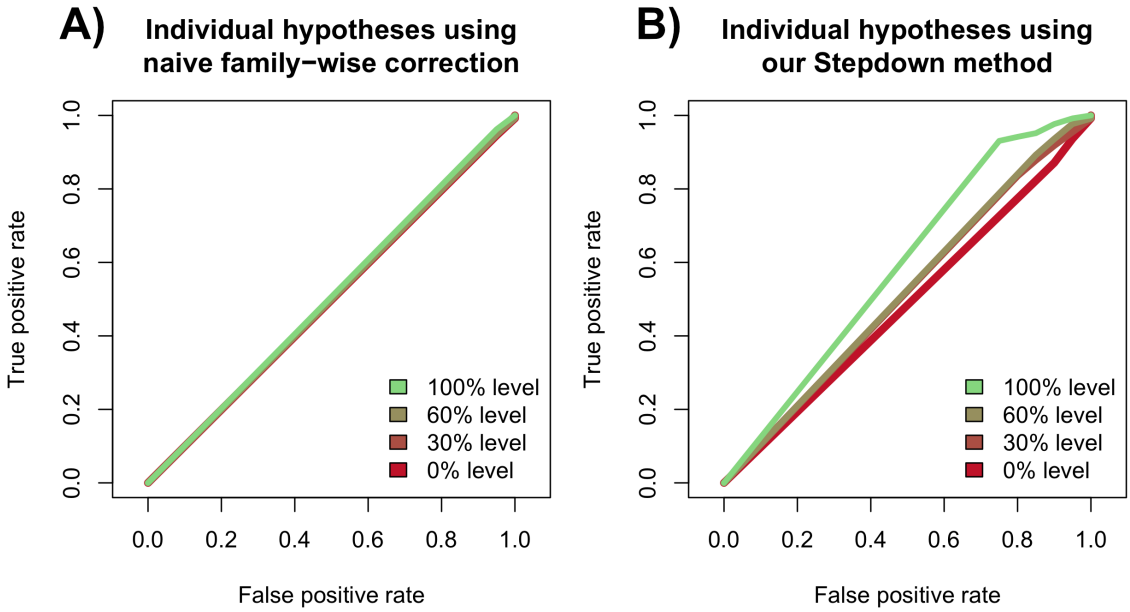


Figure 2.17: ROC curves for the hypothesis using the Accelerated Stepdown procedure described in §2.C in the nonparanormal setting. These plots are set up in the same as in Figure 2.6.

Usage of De Rubeis et al. (2014). We note that the older risk scores dataset (De Rubeis et al., 2014) is used twice, once in the screening stage (Step 1 above) and again to estimate the parameters of the HMRF (Step 4 above). As argued by Liu et al. (2015), it is important for this dataset to be the same in both steps, as the goal of DAWN is to boost the power of the risk scores by a "guilt-by-association" strategy. Hence, it is important to ensure the genes with low TADA scores remain in the analysis after screening, so they can implicate genes with TADA scores that are not as low.

Choice of λ. We use the following procedure to tune λ when estimating the Gaussian graphical model using only the 10 partitions from Window 1B as well as when using the 24 partitions selected by COBS. We tune λ on a grid between 0.05 and 0.1, equally spaced into 15 values, for both graphical models. Our criteria for selecting λ within this grid is inspired by Liu et al. (2015), who use a *scale-free index*, a number between 0 and 1 that measures how well the graph follows a power law. Specifically, we ensure the scale-free indices from both graphical models are approximately comparable as well as that both estimated graphical models have about 10,000 edges. Our focus on this number of edges comes from Liu et al. (2015), which estimated a graphical model with 10,065 edges. By ensuring both of our estimated graphical models have around 10,000 edges, we are able to

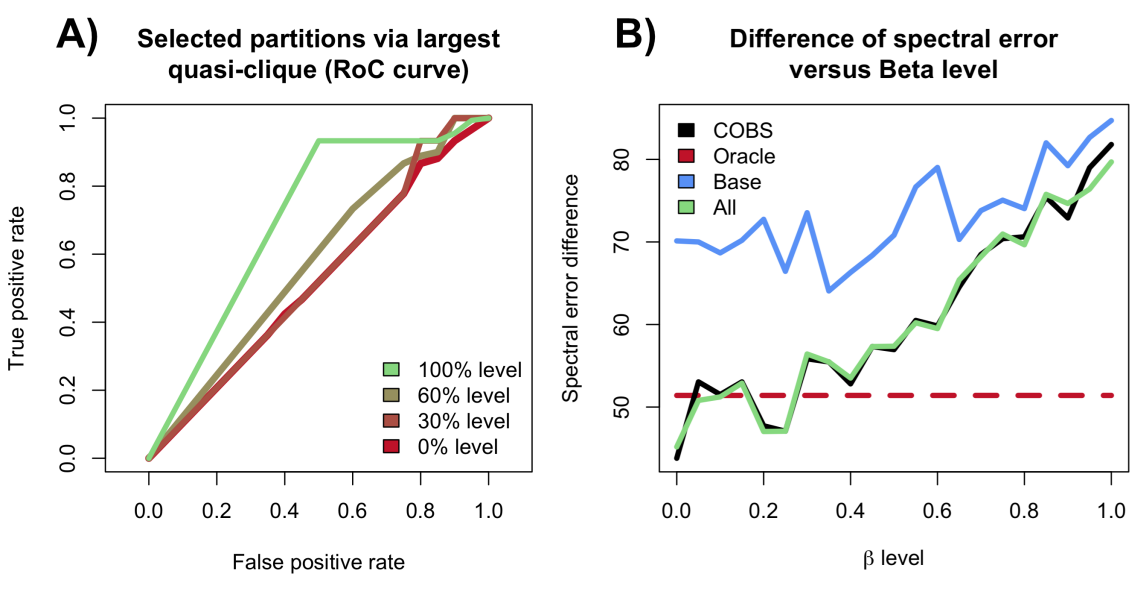


Figure 2.18: A) ROC curves for the partitions selected by COBS using the Accelerated Stepdown procedure described in §2.C in the nonparanormal setting. This plot is set up in the same way as in Figure 2.8A. B) Mean spectral error of the estimated covariance matrix for varying β level using the Accelerated Stepdown procedure in the nonparanormal setting. This plot is set up in the same way as in Figure 2.8B.

ensure that both graphical models pass roughly the same amount of information into the HMRF stage of DAWN.

Using this procedure, we set $\lambda = 0.05$ when estimating the graphical model using only the 10 partitions from Window 1B (for 9990 edges and a scale-free index of 0.77) and $\lambda = 0.064$ when estimating the graphical model using the 24 partitions selected by COBS (for 9142 edges and scale-free index of 0.83).

2.G.2 Methods to measure distance of two nodes in a graph

As alluded to in Subsection 2.6.4, the shortest path distance and the commute distance do not seem like appropriate candidates to measure the distance between two genes (i.e., vertices) in a gene co-expression network (i.e., graph) due to the fact that the network estimated in the Window 1B analysis has more edges than in the COBS analysis (9990 and 9142 edges respectively). Hence, both of these distance metrics would naturally favor the denser graph.

To overcome this problem, we use two distance metrics that we believe enable a more fair comparison.

- **Minimal spanning tree (MST) distance**: This is a natural alternative to measure the distance between two vertices. Given a graph $G = (V, E)$, we first find the MST $G_{(\text{MST})} \subseteq G$, and then compute the path distance between the two vertices in $G_{(\text{MST})}$.

- **Graph root embedding distance**: A more statistically motivated way to measure the distance between two vertices is to first embed all vertices V into a latent space. As shown in Lei (2018), the graph root embedding is a natural candidate to do this, as it can theoretically represent a wide range of random graphs. This is essentially a more sophisticated spectral embedding. We first represent the graph G as an adjacency matrix A, and compute the top-k eigenvectors (corresponding both the largest k positive eigenvalues and largest k negative eigenvalues in magnitude). Each vertex is then represented as a latent vector of length $2k$. The distance between two vertices is then defined as the Euclidean distance between their corresponding latent vectors. We defer the remaining details to Lei (2018).

  It is important to use both positive and negative eigenvalues since a scree plot reveals there are almost the same number of positive and negative eigenvalues for the adjacency matrices estimated in both the COBS and Window 1B analyses.

2.G.3 Additional results about closeness of genes

We provide more details that the 102 genes detected by the newer TADA scores (Satterstrom et al., 2020) are roughly 10%-30% closer to the 33 genes detected in the older TADA scores (De Rubeis et al., 2014) in the gene network estimated in the COBS analysis than in the Window 1B analysis. We call the 33 genes detected in De Rubeis et al. (2014) as the De Rubeis genes, and the 102 genes detected in Satterstrom et al. (2020) that are not part of the former 33 genes as the Satterstrom genes.

We use the MST distance defined above to ask: how far away are the closest k De Rubeis genes from any Satterstrom gene on average (mean). Figure 2.19A plots this average distance against k. We use the graph root embedding distance to ask: how close is the nearest De Rubeis gene from any Satterstrom gene on average (mean) when using an embedding of latent dimension $2k$. Figure 2.19B plots this average distance against k. In both instances, regardless of how the parameter k is chosen, the plot shows that the Satterstrom genes are closer to the De Rubeis genes on average. Both metrics show that the red curve is roughly 10%-30% lower than the pale curve across all values of k, hence giving our stated result.

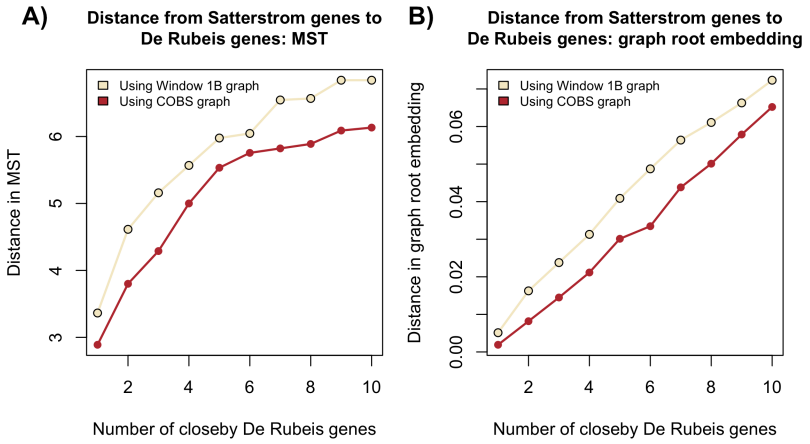


Figure 2.19: A) Average MST distance from a Satterstrom gene to the closest k De Rubeis genes against k. B) Average graph root embedding distance from a Satterstrom gene to the closest De Rubeis genes against the half of the embedding dimension k.

Three

Detecting heterogeneity – Fused lasso analysis

Paper summary: In the 1-dimensional multiple changepoint detection problem, we derive a new fast error rate for the fused lasso estimator, under the assumption that the mean vector has a sparse number of changepoints. This rate is seen to be suboptimal (compared to the minimax rate) by only a factor of $\log \log n$. Our proof technique is centered around a novel construction that we call a *lower interpolant*. We extend our results to misspecified models and exponential family distributions. We also describe the implications of our error analysis for the approximate screening of changepoints.

The work in this chapter was done jointly with James Sharpnack, Alessandro Rinaldo, and Ryan J. Tibshirani, and has been accepted at Advances in Neural Information Processing Systems under the title "A sharp error analysis for the fused lasso, with application to approximate changepoint screening."

3.1 Introduction

Consider the 1-dimensional multiple changepoint model

$$y_i = \theta_{0,i} + \epsilon_i, \quad i = 1, \ldots, n, \tag{3.1}$$

where ϵ_i, $i = 1, \ldots, n$ are i.i.d. errors, and $\theta_{0,i}$, $i = 1, \ldots, n$ is a piecewise constant mean sequence, having a set of changepoints

$$S_0 = \big\{ i \in \{1, \ldots, n-1\} : \theta_{0,i} \neq \theta_{0,i+1} \big\}. \tag{3.2}$$

This is a well-studied setting, and there is a large body of literature on estimation of the piecewise constant mean vector $\theta_0 \in \mathbb{R}^n$ and its changepoints S_0 using various estimators; refer, e.g., to the surveys Brodsky and Darkhovski (1993); Chen and Gupta (2000); Eckley et al. (2011).

In this work, we consider the *1-dimensional fused lasso* (also called 1d fused lasso, or simply fused lasso) estimator, which, given a data vector $y \in \mathbb{R}^n$ from a model as in (3.1), is defined by

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}|, \tag{3.3}$$

where $\lambda \geq 0$ serves as a tuning parameter. This was proposed and named by Tibshirani et al. (2005), but the same idea was proposed earlier in signal processing, under the name *total variation denoising*, by Rudin et al. (1992). Variants of the fused lasso have been used in biology to detect regions where two genomic samples differ due to genetic variations (Tibshirani and Wang, 2008), in finance to detect shifts in the stock market (Chan et al., 2014), and in neuroscience to detect changes in stationary behaviors of the brain (Aston and Kirch, 2012). Popularity of the fused lasso can be attributed in part to its computational scalability, the optimization problem in (3.3) being convex and highly structured. There has also been plenty of supporting statistical theory developed for the fused lasso, which we review in Section 3.2.

Notation. We will make use of the following quantities that are defined in terms of the mean θ_0 in (3.1) and its changepoint set S_0 in (3.2). We denote the size of the changepoint set by $s_0 = |S_0|$. We enumerate $S_0 = \{t_1, \ldots, t_{s_0}\}$, where $1 \leq t_1 < \ldots < t_{s_0} < n$, and for convenience we set $t_0 = 0$, $t_{s_0+1} = n$. The smallest distance between changepoints in θ_0 is denoted by

$$W_n = \min_{i=0,1\ldots,s_0} (t_{i+1} - t_i), \tag{3.4}$$

and the smallest distance between consecutive levels of θ_0 by

$$H_n = \min_{i \in S_0} |\theta_{0,i+1} - \theta_{0,i}|. \tag{3.5}$$

We use $D \in \mathbb{R}^{(n-1)\times n}$ to denote the difference operator

$$D = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ 0 & -1 & 1 & \ldots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \ldots & -1 & 1 \end{bmatrix}. \tag{3.6}$$

Note that $s_0 = \|D\theta_0\|_0$. We write D_S to extract rows of D indexed by a subset $S \subseteq \{1, \ldots, n-1\}$, and D_{-S} to extract the rows in $S^c = \{1, \ldots, n-1\} \setminus S$.

For a vector $x \in \mathbb{R}^n$, we use $\|x\|_n^2 = \|x\|_2^2/n$ to denote its length-scaled ℓ_2 norm. For sequences a_n, b_n, we use standard asymptotic notation: $a_n = O(b_n)$ to denote that a_n/b_n is bounded for large enough n, $a_n = \Omega(b_n)$ to denote that b_n/a_n is bounded for large

enough n, $a_n = \Theta(b_n)$ to denote that both $a_n = O(b_n)$ and $a_n = \Omega(b_n)$, $a_n = o(b_n)$ to denote that $a_n/b_n \to 0$, and $a_n = \omega(b_n)$ to denote that $b_n/a_n \to 0$. For random sequences A_n, B_n, we write $A_n = O_{\mathbb{P}}(B_n)$ to denote that A_n/B_n is bounded in probability. A random variable Z is said to have a sub-Gaussian distribution provided that $\mathbb{E}(Z) = 0$ and $\mathbb{P}(|Z| > t) \leq 2\exp(-t^2/(2\sigma^2))$ for all $t \geq 0$, and a constant $\sigma > 0$.

Summary of results. Our main focus is on deriving a sharp estimation error bound for the fused lasso, parametrized by the number of changepoints s_0 in θ_0. We also study several consequences of our error bound and its analysis. A summary of our contributions is as follows.

- **New error analysis for the fused lasso.** In Section 3.3, we develop a new error analysis for the fused lasso, in the model (3.1) with sub-Gaussian errors. Our analysis leverages a novel quantity that we call a *lower interpolant* to approximate the fused lasso estimate (once it has been orthogonalized with respect to the changepoint structure of the mean θ_0) with $2s_0 + 2$ monotonic segments, which allows for finer control of the empirical process term.

  When $s_0 = O(1)$, and the changepoint locations in S_0 are (asymptotically) evenly spaced, our main result implies $\mathbb{E}\|\widehat{\theta} - \theta_0\|_n^2 = O(\log n(\log\log n)/n)$ for the fused lasso estimator $\widehat{\theta}$ in (3.3). This is slower than the minimax rate by a $\log\log n$ factor. Our result improves on previously established results from Dalalyan et al. (2017), and after the completion of this paper, was itself improved upon by Guntuboyina et al. (2020) (who are able to remove the extraneous $\log\log n$ factor).

- **Extension to misspecified and exponential family models.** In Section 3.4, we extend our error analysis to cover a mean vector θ_0 that is not necessarily piecewise constant (or in other words, has potentially many changepoints). In Section 3.5, we extend our analysis to exponential family models. The latter extension, especially, is of practical importance, as many applications, e.g., CNV data analysis, call for changepoint detection on count data.

- **Application to approximate screening and recovery.** In Section 3.6, we establish that the maximum distance between any true changepoint and its nearest estimated changepoint is $O_{\mathbb{P}}(\log n(\log\log n)/H_n^2)$ using the fused lasso, when $s_0 = O(1)$ and all changepoints are (asymptotically) evenly spaced. After applying simple post-processing step, we show that the maximum distance between any estimated changepoint and its nearest true changepoint is of the same order. Our proof technique relies only on the estimation error rate of the fused lasso, and therefore immediately generalizes to *any* estimator of θ_0, where the distance (for approximate changepoint screening and recovery) is a function of the inherent error rate.

The supplementary document gives numerical simulations that support the theory in this paper.

3.2 Preliminary review of existing theory

We begin by describing known results on the quantity $\|\widehat{\theta} - \theta_0\|_n^2$, the estimation error between the fused lasso estimate $\widehat{\theta}$ in (3.3) and the mean θ_0 in (3.1).

Early results on the fused lasso are found in Mammen and van de Geer (1997) (see also Tibshirani (2014) for a translation to a setting more consistent with that of the current paper). These authors study what may be called the *weak sparsity* case, in which it is that assumed $\|D\theta_0\|_1 \leq C_n$, with D being the difference operator in (3.6). Assuming additionally that the errors in (3.1) are sub-Gaussian, Mammen and van de Geer (1997) show that for a choice of tuning parameter $\lambda = \Theta(n^{1/3} C_n^{-1/3})$, the fused lasso estimate $\widehat{\theta}$ in (3.3) satisfies

$$\|\widehat{\theta} - \theta_0\|_n^2 = O_{\mathbb{P}}(n^{-2/3} C_n^{2/3}). \tag{3.7}$$

The weak sparsity setting is not the focus of our paper, but we still recall the above result to give a sense of the difference between the weak and *strong sparsity* settings, the latter being the setting in which we assume control over $s_0 = \|D\theta_0\|_0$, as we do in the current paper. Prior to this paper, the strongest result in the strong sparsity setting was given by Dalalyan et al. (2017), who assume $N(0, \sigma^2)$ errors in (3.1), and show that for $\lambda = \sigma\sqrt{2n\log(n/\delta)}$, the fused lasso estimate satisfies

$$\|\widehat{\theta} - \theta_0\|_n^2 \leq C\sigma^2 \frac{s_0 \log(n/\delta)}{n} \bigg(\log n + \frac{n}{W_n} \bigg), \tag{3.8}$$

with probability at least $1 - 2\delta$, for large enough n, and a constant $C > 0$, where recall W_n is the minimum distance between changepoints in θ_0, as in (3.4). Our main result in Theorem 1 improves upon (3.8) in two ways: by reducing the first $\log n$ term inside the brackets to $\log s_0 + \log\log n$, and reducing the second n/W_n term to $\sqrt{n/W_n}$.

After our paper was completed, Guntuboyina et al. (2020) gave an even sharper error rate for the fused lasso (and more broadly, for trend the family of higher-order filtering estimates as defined in Steidl et al. (2006); Kim et al. (2009); Tibshirani (2014)). Again assuming $N(0, \sigma^2)$ errors in (3.1), as well as $W_n \geq cn/(s_0 + 1)$ for some constant $c \geq 1$, these authors show that the family of fused lasso estimates $\{\widehat{\theta}_\lambda,\, \lambda \geq 0\}$ (using subscripts here to explicitly denote the dependence on the tuning parameter λ) satisfies

$$\inf_{\lambda \geq 0} \|\widehat{\theta}_\lambda - \theta_0\|_n^2 \leq C\sigma^2 \frac{s_0 + 1}{n} \log\bigg(\frac{en}{s_0 + 1}\bigg) + \frac{4\sigma^2\delta}{n}, \tag{3.9}$$

with probability at least $1 - \exp(-\delta)$, for large enough n, and a constant $C > 0$. The above bound is sharper than ours in Theorem 1 in that $(\log s_0 + \log\log n)\log n + \sqrt{n/W_n}$

is replaced essentially by $\log W_n$. (Also, the result in (3.9) does not actually require $W_n \geq cn/(s_0+1)$, but only requires the distance between changepoints where jumps alternate in sign to be larger than $cn/(s_0+1)$, which is another improvement.) Further comparisons will be made in Remark 3 following Theorem 1.

There are numerous other estimators, e.g., based on segmentation techniques or wavelets, that admit estimation results comparable to those above. These are described in Remark 4 following Theorem 1. Lastly, it can be seen the minimax estimation error over the class of signals θ_0 with s_0 changepoints, assuming $N(0,\sigma^2)$ errors in (3.1), satisfies

$$\inf_{\widehat{\theta}} \sup_{\|D\theta_0\|_0 \leq s_0} \mathbb{E}\|\widehat{\theta}-\theta_0\|_n^2 \geq C\sigma^2 \frac{s_0}{n} \log\left(\frac{n}{s_0}\right), \tag{3.10}$$

for large enough n, and a constant $C>0$. This says that one cannot hope to improve the rate in (3.9). The minimax result in (3.10) follows from standard minimax theory for sparse normal means problems, as in, e.g., Johnstone (2015); for a proof, see Padilla et al. (2016).

3.3 Sharp error analysis for the fused lasso estimator

Here we derive a sharper error bound for the fused lasso, improving upon the previously established result of Dalalyan et al. (2017) as stated in (3.8). Our proof is based on a concept that we call a *lower interpolant*, which as far as we can tell, is a new idea that may be of interest in its own right.

Theorem 1. *Assume the data model in* (3.1)*, with errors* ϵ_i*,* $i=1,\ldots,n$ *i.i.d. from a sub-Gaussian distribution. Then under a choice of tuning parameter* $\lambda=(nW_n)^{1/4}$*, the fused lasso estimate* $\widehat{\theta}$ *in* (3.3) *satisfies*

$$\|\widehat{\theta}-\theta_0\|_n^2 \leq \gamma^2 c \frac{s_0}{n}\left((\log s_0 + \log\log n)\log n + \sqrt{\frac{n}{W_n}}\right),$$

with probability at least $1-\exp(-C\gamma)$*, for all* $\gamma>1$ *and* $n \geq N$*, where* $c, C, N > 0$ *are constants that depend on only* σ *(the parameter appearing in the sub-Gaussian distribution of the errors).*

An immediate corollary is as follows.

Corollary 2. *Under the same assumptions as in Theorem 1, we have*

$$\mathbb{E}\|\widehat{\theta}-\theta_0\|_n^2 \leq c \frac{s_0}{n}\left((\log s_0 + \log\log n)\log n + \sqrt{\frac{n}{W_n}}\right),$$

for some constant $c>0$.

We give some remarks comparing Theorem 1 to related results in the literature.

Remark 3 (**Comparison to Dalalyan et al. (2017); Guntuboyina et al. (2020)**). *We can see that the result in Theorem 1 is sharper than that in* (3.8) *from Dalalyan et al. (2017) for any* s_0, W_n*, as* $\log s_0 \le \log n$ *and* $\sqrt{n/W_n} \le n/W_n$*. Moreover, when* $s_0 = O(1)$ *and* $W_n = \Theta(n)$*, the rates are* $\log^2 n/n$ *and* $\log n(\log\log n)/n$ *from Theorem 1 and* (3.8)*, respectively.*

Comparing the result in Theorem 1 to that in (3.9) *from Guntuboyina et al. (2020), the latter is sharper in that it reduces the factor of* $(\log s_0 + \log\log n)\log n + \sqrt{n/W_n}$ *to a single term of* $\log W_n$*. In the case* $s_0 = O(1)$ *and* $W_n = \Theta(n)$*, the rates are* $\log n(\log\log n)/n$ *and* $\log n/n$ *from Theorem 1 and* (3.8)*, respectively, and the latter rate cannot be improved, owing to the minimax lower bound in* (3.10)*. Similar to our expectation bound in Corollary 2, Guntuboyina et al. (2020) establish*

$$\inf_{\lambda \ge 0} \mathbb{E}\|\widehat{\theta}_\lambda - \theta_0\|_n^2 \le C\sigma^2 \frac{s_0+1}{n} \log\left(\frac{en}{s_0+1}\right), \tag{3.11}$$

for the family of fused lasso estimates $\{\widehat{\theta}_\lambda,\ \lambda \ge 0\}$*, for large enough* n*, and a constant* $C > 0$*. Like their high probability result in* (3.9)*, their expectation result in* (3.11) *is stated in terms of an infimum over* $\lambda \ge 0$*, and does not provide an explicit value of* λ *that attains the bound. (Inspection of their proofs suggests that it is not at all easy to make such a value of* λ *explicit.) Meanwhile, Theorem 1 and Corollary 1 have the advantage this choice is made explicit, as in* $\lambda = (nW_n)^{1/4}$*.*

Remark 4 (**Comparison to other estimators**). *Various other estimators obtain comparable estimation error rates. In what follows, all results are stated in the case* $s_0 = O(1)$*. The Potts estimator, defined by replacing the* ℓ_1 *penalty* $\sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}|$ *in* (3.3) *with the* ℓ_0 *penalty* $\sum_{i=1}^{n-1} 1\{\theta_i \ne \theta_{i+1}\}$*, and denoted say by* $\widehat{\theta}^{\text{Potts}}$*, satisfies a bound* $\|\widehat{\theta}^{\text{Potts}} - \theta_0\|_n^2 = O(\log n/n)$ *a.s. as shown by Boysen et al. (2009). Wavelet denoising (placing weak conditions on the wavelet basis), denoted by* $\widehat{\theta}^{\text{wav}}$*, satisfies* $\mathbb{E}\|\widehat{\theta}^{\text{wav}} - \theta_0\|_n^2 = O(\log^2 n/n)$ *as shown by Donoho and Johnstone (1994). Pairing unbalanced Haar (UH) wavelets with a basis selection method, Fryzlewicz (2007) developed an estimator* $\widehat{\theta}^{\text{UH}}$ *with* $\mathbb{E}\|\widehat{\theta}^{\text{UH}} - \theta_0\|_n^2 = O(\log^2 n/n)$*. Though they are not written in this form, the results in Fryzlewicz et al. (2018) imply that his "tail-greedy" unbalanced Haar (TGUH) estimator,* $\widehat{\theta}^{\text{TGUH}}$*, satisfies* $\|\widehat{\theta}^{\text{TGUH}} - \theta_0\|_n^2 = O(\log^2 n/n)$ *with probability tending to 1.*

Here is an overview of the proof of Theorem 1. The full proof is deferred until the supplement, as with all proofs in this paper. We begin by deriving a basic inequality (stemming from the optimality of the fused lasso estimate $\widehat{\theta}$ in (3.3)):

$$\|\widehat{\theta} - \theta_0\|_2^2 \le 2\epsilon^\top(\widehat{\theta} - \theta_0) + 2\lambda\big(\|D\theta_0\|_1 - \|D\widehat{\theta}\|_1\big). \tag{3.12}$$

To precisely control the empirical process term $\epsilon^\top(\widehat{\theta} - \theta_0)$, we consider a decomposition

$$\epsilon^\top(\widehat{\theta} - \theta_0) = \epsilon^\top \widehat{\delta} + \epsilon^\top \widehat{x},$$

where we define $\widehat{\delta} = P_0(\widehat{\theta} - \theta_0)$ and $\widehat{x} = P_1\widehat{\theta}$. Here P_0 is the projection matrix onto the piecewise constant structure inherent in θ_0, and $P_1 = I - P_0$. More precisely, writing $S_0 = \{t_1, \ldots, t_{s_0}\}$ for the set of ordered changepoints in θ_0, we define $B_j = \{t_j + 1, \ldots, t_{j+1}\}$, and denote by $\mathbf{1}_{B_j} \in \mathbb{R}^n$ the indicator of block B_j, for $j = 0, \ldots, s_0$. In this notation, P_0 is the projection onto the $(s_0 + 1)$-dimensional linear subspace $\mathcal{R} = \text{span}\{\mathbf{1}_{B_0}, \ldots, \mathbf{1}_{B_{s_0}}\}$. The parameter $\widehat{\delta}$ lies in an low-dimensional subspace, which makes bounding the term $\epsilon^\top \widehat{\delta}$ relatively easy. Bounding the term $\epsilon^\top \widehat{x}$ requires a much more intricate argument, which is spelled out in the following lemmas.

Lemma 5 is a deterministic result ensuring the existence of what we call a *lower interpolant* $\widehat{z}$ to $\widehat{x}$. This interpolant approximates $\widehat{x}$ using $2s_0 + 2$ monotonic segments, and its empirical process term $\epsilon^\top \widehat{z}$ can be finely controlled, as shown in Lemma 6. The residual from the interpolant approximation, denoted $\widehat{w} = \widehat{x} - \widehat{z}$, has an empirical process term $\epsilon^\top \widehat{w}$ that is more crudely controlled, in Lemma 7. Put together, as in $\epsilon^\top \widehat{x} = \epsilon^\top \widehat{z} + \epsilon^\top \widehat{w}$, gives the final control on $\epsilon^\top \widehat{x}$.

Before stating Lemma 5, we define the class of vectors containing the lower interpolant. Given any collection of changepoints $t_1 < \ldots < t_{s_0}$ (and $t_0 = 0$, $t_{s_0+1} = n$), let $\mathcal{M}$ be the set of "piecewise monotonic" vectors $z \in \mathbb{R}^n$, with the following properties, for each $i = 0, \ldots, s_0$:

(i) there exists a point t_i' such that $t_i + 1 \le t_i' \le t_{i+1}$, and such that the absolute value $|z_j|$ is nonincreasing over the segment $j \in \{t_i + 1, \ldots, t_i'\}$, and nondecreasing over the segment $j \in \{t_i', \ldots, t_{i+1}\}$;

(ii) the signs remain constant on the monotone pieces,

$$\begin{aligned} &\text{sign}(z_{t_i}) \cdot \text{sign}(z_j) \ge 0, \quad j = t_i + 1, \ldots, t_i', \\ &\text{sign}(z_{t_{i+1}}) \cdot \text{sign}(z_j) \ge 0, \quad j = t_i' + 1, \ldots, t_{i+1}. \end{aligned}$$

Now we state our lemma that characterizes the lower interpolant.

Lemma 5. *Given changepoints $t_0 < \ldots < t_{s_0+1}$, and any $x \in \mathbb{R}^n$, there exists a vector $z \in \mathcal{M}$ (not necessarily unique), such that the following statements hold:*

$$\|D_{-S_0} x\|_1 = \|D_{-S_0} z\|_1 + \|D_{-S_0}(x - z)\|_1, \tag{3.13}$$

$$\|D_{S_0} x\|_1 = \|D_{S_0} z\|_1 \le \|D_{-S_0} z\|_1 + \frac{4\sqrt{s_0}}{\sqrt{W_n}} \|z\|_2, \tag{3.14}$$

$$\|z\|_2 \le \|x\|_2 \quad \textit{and} \quad \|x - z\|_2 \le \|x\|_2, \tag{3.15}$$

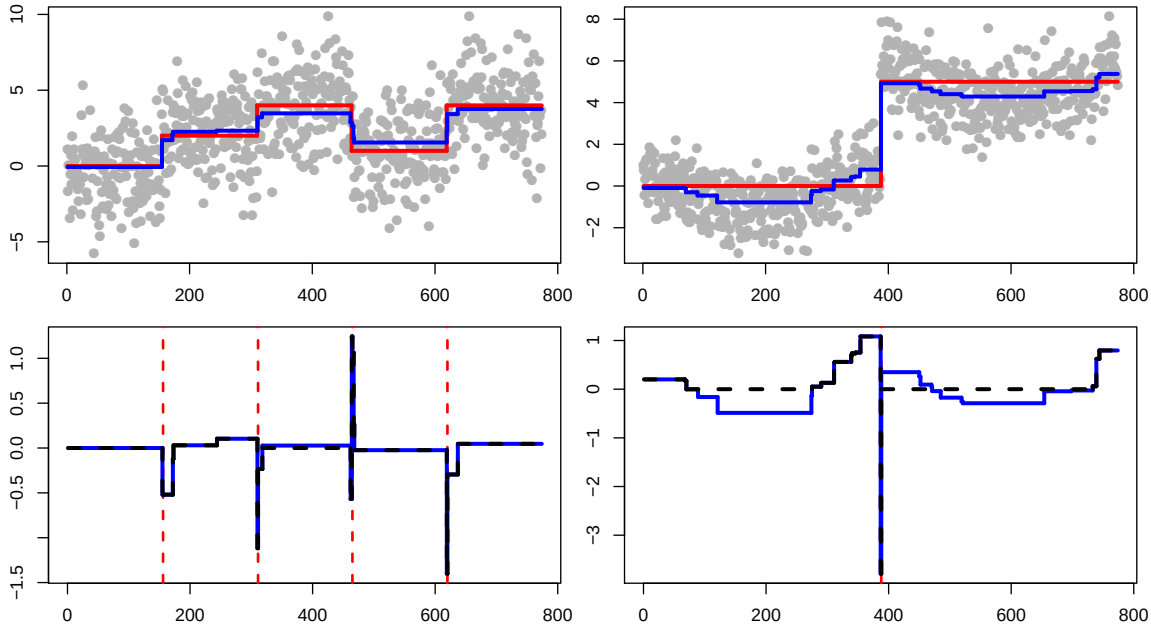


Figure 3.1: The lower interpolants for two examples (in the left and right columns), each with $n = 800$ points. In the top row, the data y (in gray) and underlying signal θ_0 (red) are plotted across the locations $1, \ldots, n$. Also shown is the fused lasso estimate $\widehat{\theta}$ (blue). In the bottom row, the error vector $\widehat{x} = P_1\widehat{\theta}$ is plotted (blue) as well as the interpolant (black), and the dotted vertical lines (red) denote the changepoints $t_1, \ldots t_{s_0}$ of θ_0.

where $D \in \mathbb{R}^{(n-1)\times n}$ is the difference matrix in (3.6)*. We call a vector z with these properties a* lower interpolant *to x.*

Loosely speaking, the lower interpolant $\widehat{z}$ can be visualized by taking a string that lies initially on top of $\widehat{x}$, is nailed down at the changepoints $t_0, \ldots t_{s_0+1}$, and then pulled taut while maintaining that it is not greater (elementwise) than $\widehat{x}$, in magnitude. Here "pulling taut" means that $\|D\widehat{z}\|_1$ is made small. Figure 3.1 provides illustrations of the interpolant $\widehat{z}$ to $\widehat{x}$ for a few examples.

Note that $\widehat{z}$ consists of $2s_0 + 2$ monotonic pieces. This special structure leads to a sharp concentration inequality. The next lemma is the primary contributor to the fast rate given in Theorem 1.

Lemma 6. *Given changepoints $t_1 < \ldots < t_{s_0}$, there exists constants $c_I, C_I, N_I > 0$ such*

that when $\epsilon \in \mathbb{R}^n$ has i.i.d. sub-Gaussian components,

$$\mathbb{P}\bigg(\sup_{z\in\mathcal{M}} \frac{|\epsilon^\top z|}{\|z\|_2} > \gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n}\bigg) \leq 2\exp\big(-C_I\gamma^2 c_I^2(\log s_0 + \log\log n)\big),$$

for any $\gamma > 1$, and $n \geq N_I$.

Finally, the following lemma controls the residuals, $\widehat{w} = \widehat{x} - \widehat{z}$.

Lemma 7. *Given changepoints $t_1 < \ldots < t_{s_0}$, there exists constants $c_R, C_R > 0$ such that when $\epsilon \in \mathbb{R}^n$ has i.i.d. sub-Gaussian components,*

$$\mathbb{P}\bigg(\sup_{w\in\mathcal{R}^\perp} \frac{|\epsilon^\top w|}{\sqrt{\|D_{-S_0}w\|_1\|w\|_2}} > \gamma c_R (ns_0)^{1/4}\bigg) \leq 2\exp(-C_R\gamma^2 c_R^2\sqrt{s_0}),$$

for any $\gamma > 1$, where $\mathcal{R}^\perp$ is the orthogonal complement of $\mathcal{R} = \mathrm{span}\{\mathbf{1}_{B_0}, \ldots, \mathbf{1}_{B_{s_0}}\}$.

3.4 Extension to misspecified models

We consider data from the model in (3.1) but where the mean θ_0 is not necessarily piecewise constant (i.e., where s_0 is potentially large). Let us define

$$\theta_0(s) = \underset{\theta\in\mathbb{R}^n}{\mathrm{argmin}}\ \|\theta_0 - \theta\|_2^2 \quad \text{subject to} \quad \|D\theta\|_0 \leq s, \tag{3.16}$$

which we call the *best s-approximation* to θ_0. We now present an extension of Theorem 1.

Theorem 8. *Assume the data model in (3.1), with errors ϵ_i, $i = 1, \ldots, n$ i.i.d. from a sub-Gaussian distribution. For any s, consider the best s-approximation $\theta_0(s)$ to θ_0, as in (3.16), and let $W_n(s)$ be the minimum distance between the s changepoints in $\theta_0(s)$. Then under a choice of tuning parameter $\lambda = (nW_n(s))^{1/4}$, the fused lasso estimate $\widehat{\theta}$ in (3.3) satisfies*

$$\|\widehat{\theta} - \theta_0\|_n^2 \leq \|\theta_0(s) - \theta_0\|_n^2 + \gamma^2 c\frac{s}{n}\bigg((\log s + \log\log n)\log n + \sqrt{\frac{n}{W_n(s)}}\bigg), \tag{3.17}$$

with probability at least $1 - \exp(-C\gamma)$, for all $\gamma > 1$ and $n \geq N$, where $c, C, N > 0$ are constants that depend on only σ. Further, if λ is chosen large enough so that $\|D\widehat{\theta}\|_0 \leq s$ on an event E, then

$$\|\widehat{\theta} - \theta_0(s)\|_n^2 \leq \gamma^2 c\frac{s}{n}\bigg((\log s + \log\log n)\log n + \frac{\lambda^2}{W_n(s)} + \frac{n}{\lambda^2}\bigg), \tag{3.18}$$

on E intersected with an event of probability at least $1 - \exp(-C\gamma)$, for all $\gamma > 1$, $n \geq N$, where $c, C, N > 0$ are the same constants as above.

The first result in (3.17) in Theorem 8 is a standard oracle inequality. It provides a bound on the error of the fused lasso estimator that decomposes into two parts, the first term being the approximation error, determined by the proximity of $\theta_0(s)$ to θ_0, and second term being the usual bound we would encounter if the mean truly had s changepoints.

The second result in (3.18) in the theorem is a direct bound on the estimation error $\|\widehat{\theta} - \theta_0(s)\|_n^2$. We see that the estimation error can be small, apparently regardless of the size of $\|\theta_0(s) - \theta_0\|_n^2$, if we take λ to be large enough for $\widehat{\theta}$ to itself have s changepoints. But the rate worsens as λ grows larger, so implicitly, the proximity of $\theta_0(s)$ to θ_0 does play an role (if θ_0 were actually far away from a signal with s changepoints, then we may have to take λ very large to ensure that $\widehat{\theta}$ has s changepoints).

Remark 9 (**Comparison to other results**). *Dalalyan et al. (2017); Guntuboyina et al. (2020) also provide oracle inequalities and their results could be adapted to take forms as in Theorem 8. It is not clear to us that previous results on other estimators, such as those from Remark 4, adapt as easily.*

3.5 Extension to exponential family models

We consider data $y = (y_1, \dots, y_n) \in \mathbb{R}^n$ with independent components distributed according to

$$p(y_i; \theta_{0,i}) = h(y_i) \exp\big(y_i \theta_{0,i} - \Lambda(\theta_{0,i})\big), \quad i = 1, \dots, n. \tag{3.19}$$

Here, for each $i = 1, \dots, n$, the parameter $\theta_{0,i}$ is the natural parameter in the exponential family and Λ is the cumulant generating function. As before, in the location model, we are mainly interested in the case in which the natural parameter vector θ_0 is piecewise constant (with s_0 denoting its number of changepoints, as before). Estimation is now based on penalization of the negative log-likelihood:

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^n \big(- y_i \theta_i + \Lambda(\theta_i)\big) + \lambda \sum_{i=1}^n |\theta_i - \theta_{i+1}|, \tag{3.20}$$

Since the cumulant generating function Λ is always convex in exponential families, the above is a convex optimization problem. We present an estimation error bound the present setting.

Theorem 10. *Assume the data model in* (3.19)*, with a strictly convex, twice continuously differentiable cumulant generating function* Λ*. Assume that* $\theta_{0,i} \in [l, u]$*,* $i = 1, \dots, n$ *for constants* $l, u \in \mathbb{R}$*, and add the constraints* $\theta_i \in [l, u]$*,* $i = 1, \dots, n$ *in the optimization problem in* (3.20)*. Finally, assume that the random variables* $y_i - \mathbb{E}(y_i)$*,* $i = 1, \dots, n$ *obey a sub-Gaussian distribution, with parameter* σ*. Then under a choice of tuning parameter* $\lambda = (nW_n)^{1/4}$*, the exponential family fused lasso estimate* $\widehat{\theta}$ *in* (3.20) *(subject to the additional boundedness*

constraints) satisfies

$$\|\widehat{\theta} - \theta_0\|_n^2 \leq \gamma^2 c \frac{s_0}{n} \left((\log s_0 + \log\log n) \log n + \sqrt{\frac{n}{W_n}} \right),$$

with probability at least $1 - \exp(-C\gamma)$*, for all* $\gamma > 1$ *and* $n \geq N$*, where* $c, C, N > 0$ *are constants that depend on only* l, u, σ*.*

Remark 11 (**Roles of** l, u)**.** *The restriction of* $\theta_{0,i}$ *and the optimization parameters in* (3.20) *to* $[l, u]$*, for* $i = 1, \ldots, n$*, is used to ensure that the second derivative of* Λ *is bounded away from zero. (The same property could be accomplished by instead adding a small squared* ℓ_2 *penalty on* θ *in* (3.20)*.) A more refined analysis could alleviate the need for this bounded domain (or extra squared* ℓ_2 *penalty) but we do not pursue this for simplicity.*

Remark 12 (**Sub-Gaussianity in exponential families**)**.** *When are the random variables* $y_i - \mathbb{E}(y_i)$*,* $i = 1, \ldots, n$ *sub-Gaussian, in an exponential family model* (3.19)*? A simple sufficient condition (not specific to exponential families, in fact) is that these centered variates are bounded. This covers the binomial model* $y_i \sim \mathrm{Bin}(k, \mu(\theta_{0,i}))$*, where* $\mu(\theta_{0,i}) = 1/(1 + e^{-\theta_{0,i}})$*,* $i = 1, \ldots, n$*, and* k *is a fixed constant. Hence Theorem 10 applies to binomial data.*

For Poisson data $y_i \sim \mathrm{Pois}(\mu(\theta_{0,i}))$*, where* $\mu(\theta_{0,i}) = e^{\theta_{0,i}}$*,* $i = 1, \ldots, n$*, we now give two options for the analysis. The first is to assume a maximum achieveable count (which may be reasonable in CNV data) and then apply Theorem 10 owing again to boundedness. The second is to invoke the fact that Poisson random variables have sub-exponential (rather than sub-Gaussian) tails, and then use a truncation argument, to show that for the Poisson fused lasso estimate* $\widehat{\theta}$ *in* (3.20) *(under the additional boundedness constraints), with* $\lambda = \log n (n W_n)^{1/4}$*,*

$$\|\widehat{\theta} - \theta_0\|_n^2 \leq \gamma^2 c \frac{s_0 \log n}{n} \left((\log s_0 + \log\log n) \log n + \sqrt{\frac{n}{W_n}} \right), \tag{3.21}$$

with probability at least $1 - \exp(-C\gamma) - 1/n$*, for all* $\gamma > 1$ *and* $n \geq N$*, where* $c, C, N > 0$ *are constants depending on* l, u*. This is slower than the rate in Theorem 10 by a factor of* $\log n$*.*

Remark 13 (**Comparison to other results**)**.** *The results in Dalalyan et al. (2017); Guntuboyina et al. (2020) assume normal errors. It seems believable to us that the results of Dalalyan et al. (2017) could be extended to sub-Gaussian errors and hence exponential family data, in a manner similar to what we have done above in Theorem 10. To us, this is less clear for the results of Guntuboyina et al. (2020), which rely on some technical calculations involving Gaussian widths. It is even less clear to us how results from other estimators, as in Remark 4, extend to exponential family data.*

3.6 Approximate changepoint screening and recovery

In many applications of changepoint detection, one may be interested in estimation of the changepoint locations in θ_0, rather than the mean vector θ_0 as a whole. In this section, we show that estimation of the changepoint locations and of θ_0 itself are two very closely linked problems, in the following sense: any procedure with guarantees on its error in estimating θ_0 automatically has certain approximate changepoint detection guarantees, and not surprisingly, a faster error rate (in estimating θ_0) translates into a stronger statement about approximate changepoint detection. We use this general link to prove new approximate changepoint screening results for the fused lasso. We also show that in general a simple post-processing step may be used to discard spurious detected changepoints, and again apply this to the fused lasso to yield new approximate changepoint recovery results.

It helps to introduce some additional notation. For a vector $\theta \in \mathbb{R}^n$, we write $S(\theta)$ for the set of its changepoint indices, i.e.,

$$S(\theta) = \big\{ i \in \{1, \dots, n-1\} : \theta_i \neq \theta_{i+1} \big\}.$$

Recall, we abbreviate $S_0 = S(\theta_0)$ for the changepoints of the underlying mean θ_0. For two discrete sets A, B, we define the metrics

$$d(A|B) = \max_{b \in B} \min_{a \in A} |a - b| \quad \text{and} \quad d_H(A, B) = \max\big\{ d(A|B), d(B|A) \big\}.$$

The first metric above can be seen as a one-sided screening distance from B to A, measuring the furthest distance of an element in B to its closest element in A. The second metric above is known as the *Hausdorff distance* between A and B.

Approximate changepoint screening. We present our general theorem on changepoint screening. The basic idea behind the result is quite simple: if an estimator misses a (large) changepoint in θ_0, then its estimation error must suffer, and we can use this fact to bound the screening distance.

Theorem 14. *Let $\widetilde{\theta} \in \mathbb{R}^n$ be an estimator such that $\|\widetilde{\theta} - \theta_0\|_n^2 = O_{\mathbb{P}}(R_n)$. Assume that $nR_n/H_n^2 = o(W_n)$, where, recall, H_n is the minimum gap between adjacent levels of θ_0, defined in* (3.5)*, and W_n is the minimum distance between adjacent changepoints of θ_0, defined in* (3.4)*. Then*

$$d\big(S(\widetilde{\theta}) \,|\, S_0\big) = O_{\mathbb{P}}\bigg(\frac{nR_n}{H_n^2}\bigg).$$

Remark 15 (**Generic setting: no specific data model, and no assumptions on estimator**)**.** *Importantly, Theorem 14 assumes no data model whatsoever, and treats $\widetilde{\theta}$ as a generic estimator of θ_0. (Of course, through the statement $\|\widetilde{\theta} - \theta_0\|_n^2 = O_{\mathbb{P}}(R_n)$, one can see*

that $\widetilde{\theta}$ is random, constructed from data that depends on θ_0, but no specific data model is required, nor are any specific properties of $\widetilde{\theta}$, other than its error rate.) This flexibility allows for the result to be applied in any problem setting in which one has control of the error in estimating a piecewise constant parameter θ_0 (in some cases this may be easier to obtain, compared to direct analysis of detection properties). A similar idea was used (concurrently and independently) by Fryzlewicz et al. (2018) in the analysis of the TGUH estimator.

Combining the above theorem with known error rates for the fused lasso estimator—(3.7) in the weak sparsity case, and Theorem 1 in the strong sparsity case—gives the following result.

Corollary 16. *Assume the data model in* (3.1)*, with errors ϵ_i, $i = 1, \ldots, n$ i.i.d. from a sub-Gaussian distribution. Let $C_n = \|D\theta_0\|_1$, and assume that $H_n = \omega(n^{1/6} C_n^{1/3} / \sqrt{W_n})$. Then the fused lasso estimator $\widehat{\theta}$ in* (3.3) *with $\lambda = \Theta(n^{1/3} C_n^{-1/3})$ satisfies*

$$d\big(S(\widehat{\theta}) \,|\, S_0\big) = O_{\mathbb{P}}\bigg(\frac{n^{1/3} C_n^{2/3}}{H_n^2}\bigg). \tag{3.22}$$

Alternatively, assume $s_0 = O(1)$, $W_n = \Theta(n)$, and $H_n = \omega(\sqrt{\log n (\log\log n)/n})$. Then the fused lasso with $\lambda = \Theta(\sqrt{n})$ satisfies

$$d\big(S(\widehat{\theta}) \,|\, S_0\big) = O_{\mathbb{P}}\bigg(\frac{\log n (\log\log n)}{H_n^2}\bigg). \tag{3.23}$$

Remark 17 (**Changepoint detection limit**)**.** *The restriction $H_n = \omega(\sqrt{\log n (\log\log n)/n})$ for* (3.23) *in Corollary 16 is very close to the optimal detection limit of $H_n = \omega(1/\sqrt{n})$: Duembgen and Walther (2008) showed that in Gaussian changepoint model with a single elevated region, and $W_n = \Theta(n)$, there is no test for detecting a changepoint that has asymptotic power 1 unless $H_n = \omega(1/\sqrt{n})$.*

Combining Theorem 14 with (3.21) gives the following (a similar result holds for the binomial model).

Corollary 18. *Assume $y_i \sim \mathrm{Pois}(e^{\theta_{0,i}})$, independently, for $i = 1, \ldots, n$, and assume $\|\theta_0\|_\infty = O(1)$, $s_0 = O(1)$, $W_n = \Theta(n)$, $H_n = \omega(\log n \sqrt{\log\log n / n})$. Then for the Poisson fused lasso estimator $\widehat{\theta}$ in* (3.20) *(subject to appropriate boundedness constraints) with $\lambda = \Theta(\log n \sqrt{n})$, we have*

$$d\big(S(\widehat{\theta}) \,|\, S_0\big) = O_{\mathbb{P}}\bigg(\frac{\log^2 n (\log\log n)}{H_n^2}\bigg).$$

Approximate changepoint recovery. We present a post-processing procedure for the estimated changepoints in $\widetilde{\theta}$, to eliminate changepoints of $\widetilde{\theta}$ that lie far away from changepoints of θ_0. Our procedure is based on convolving $\widetilde{\theta}$ with a filter that resembles the mother Haar wavelet. Consider

$$F_i(\widetilde{\theta}) = \frac{1}{b_n} \sum_{j=i+1}^{i+b_n} \widetilde{\theta}_j - \frac{1}{b_n} \sum_{j=i-b_n+1}^{i} \widetilde{\theta}_j, \quad \text{for } i = b_n, \dots, n - b_n, \tag{3.24}$$

for an integral bandwidth $b_n > 0$. By evaluating the filter $F_i(\widetilde{\theta})$ at all locations $i = b_n, \dots, n - b_n$, and retaining only locations at which the filter value is large (in magnitude), we can approximately recovery the changepoints of θ_0, in the Hausdorff metric. This idea is very similar to the one proposed in Hao et al. (2013), which study the approximate false discovery rate of a similar procedure. We wish to investigate the theoretical relations in our future work.

Theorem 19. *Let $\widetilde{\theta} \in \mathbb{R}^n$ be such that $\|\widetilde{\theta} - \theta_0\|_n^2 = O_{\mathbb{P}}(R_n)$. Consider the following procedure: we evaluate the filter in* (3.24) *with bandwidth b_n at locations in*

$$I_F(\widetilde{\theta}) = \Big\{ i \in \{b_n, \dots, n - b_n\} : i \in S(\widetilde{\theta}), \text{ or } i + b_n \in S(\widetilde{\theta}), \text{ or } i - b_n \in S(\widetilde{\theta}) \Big\} \cup \{b_n, n - b_n\},$$

and define a set of filtered points $S_F(\widetilde{\theta}) = \{i \in I_F(\widetilde{\theta}) : |F_i(\widetilde{\theta})| \geq \tau_n\}$, for a threshold level τ_n. If b_n, τ_n satisfy $b_n = \omega(nR_n/H_n^2)$, $2b_n \leq W_n$, and $\tau_n/H_n \to \rho \in (0,1)$ as $n \to \infty$, then

$$\mathbb{P}\Big(d_H\big(S_F(\widetilde{\theta}), S_0\big) \leq 2b_n \Big) \to 1 \quad \text{as } n \to \infty.$$

Note that the set of filtered points $|S_F(\widetilde{\theta})|$ in Theorem 19 is not necessarily of a subset of the original set of estimated changepoints $S(\widetilde{\theta})$, but it has the property $|S_F(\widetilde{\theta})| \leq 3|S(\widetilde{\theta})| + 2$.

We finish with corollaries for the fused lasso. For space reasons, remarks comparing them to related approximate recovery results in the literature are deferred to the supplement.

Corollary 20. *Assume the data model in* (3.1)*, with errors ϵ_i, $i = 1, \dots, n$ i.i.d. from a sub-Gaussian distribution. Let $C_n = \|D\theta_0\|_1$. If we apply the post-processing procedure in Theorem 19 to the fused lasso estimator $\widehat{\theta}$ in* (3.3) *with $\lambda = \Theta(n^{1/3} C_n^{-1/3})$, $b_n = \lfloor n^{1/3} C_n^{2/3} \nu_n^2 / H_n^2 \rfloor \leq W_n/2$ for a sequence $\nu_n \to \infty$, and $\tau_n/H_n \to \rho \in (0,1)$, then*

$$\mathbb{P}\left(d_H\big(S_F(\widehat{\theta}), S_0\big) \leq \frac{2n^{1/3} C_n^{2/3} \nu_n^2}{H_n^2} \right) \to 1 \quad \text{as } n \to \infty. \tag{3.25}$$

Alternatively, assuming $s_0 = O(1)$, $W_n = \Theta(n)$, if we apply the same post-processing procedure to the fused lasso with $\lambda = \Theta(\sqrt{n})$, $b_n = \lfloor \log n (\log\log n) \nu_n^2 / H_n^2 \rfloor \leq W_n/2$ for a sequence $\nu_n \to \infty$, and $\tau_n/H_n \to \rho \in (0,1)$, then

$$\mathbb{P}\left(d_H\big(S_F(\widehat{\theta}), S_0\big) \leq \frac{2 \log n (\log\log n) \nu_n^2}{H_n^2} \right) \to 1 \quad \text{as } n \to \infty. \tag{3.26}$$

Corollary 21. *Assume* $y_i \sim \mathrm{Pois}(e^{\theta_{0,i}})$*, independently, for* $i = 1, \ldots, n$*, and assume* $\|\theta_0\|_\infty = O(1)$*,* $s_0 = O(1)$*,* $W_n = \Theta(n)$*. If we apply the post-processing method in Theorem 19 to the Poisson fused lasso estimator* $\widehat{\theta}$ *in* (3.20) *(subject to appropriate boundedness constraints) with* $\lambda = \Theta(\log n\sqrt{n})$*,* $b_n = \lfloor \log^2 n(\log\log n)\nu_n^2/H_n^2 \rfloor \le W_n/2$ *for a sequence* $\nu_n \to \infty$*, and* $\tau_n/H_n \to \rho \in (0,1)$*, then*

$$\mathbb{P}\bigg(d_H\big(S_F(\widehat{\theta}), S_0\big) \le \frac{2\log^2 n(\log\log n)\nu_n^2}{H_n^2}\bigg) \to 1 \quad \text{as } n \to \infty.$$

3.7 Summary

We gave a new error analysis for the fused lasso, with extensions to misspecified models and data from exponential families. We showed that error bounds for general changepoint estimators lead to approximate changepoint screening results, and after post-processing, approximate recovery results.

3.A Proofs

3.A.1 Proofs of Theorem 1 and Corollary 2

We denote by $N(r, S, \|\cdot\|)$ the covering number of a set S in a norm $\|\cdot\|$, i.e., the smallest number of $\|\cdot\|$-balls of radius r needed to cover S. We call $\log N(r, S, \|\cdot\|)$ the log covering or entropy number. Recall that we write $\|\cdot\|_n = \|\cdot\|_2/\sqrt{n}$ for the scaled ℓ_2 norm, and that we say a random variable Z has a sub-Gaussian distribution provided that

$$\mathbb{E}[Z] = 0 \quad \text{and} \quad \mathbb{P}(|Z| > t) \le 2\exp\big(-t^2/(2\sigma^2)\big) \quad \text{for } t \ge 0, \tag{3.27}$$

for some constant $\sigma > 0$.

In the proof of Theorem 1, we will rely on the following result from van de Geer (1990) (which is derived closely from Dudley's chaining for sub-Gaussian processes).

Theorem 22 (Theorem 3.3 of van de Geer 1990). *Assume that* $\epsilon = (\epsilon_1, \ldots, \epsilon_n) \in \mathbb{R}^n$ *has i.i.d. components drawn from a sub-Gaussian distribution, as in* (3.27)*. Consider a set* $\mathcal{X} \subseteq \mathbb{R}^n$ *such that* $\|x\|_n \le 1$ *for all* $x \in \mathcal{X}$*, and let* $\mathcal{K}(\cdot)$ *be a continuous function upper bounding the* $\|\cdot\|_n$*-entropy of* $\mathcal{X}$*, i.e.,* $\mathcal{K}(r) \ge \log N(r, \mathcal{X}, \|\cdot\|_n)$*. Then there are constants* $C_1, C_2, C_3, C_4 > 0$ *depending only on* σ *(the parameter in the underlying sub-Gaussian distribution) such that for all* $t > C_1$*, with*

$$t > C_2 \int_0^{t_0} \sqrt{\mathcal{K}(r)}\, dr,$$

where $t_0 = \inf\{r : \mathcal{K}(r) \le C_3 t^2\}$*, we have*

$$\mathbb{P}\bigg(\sup_{x \in \mathcal{X}} \frac{|\epsilon^\top x|}{\sqrt{n}} > t\bigg) \le 2\exp(-C_4 t^2).$$

Now we give the proof of Theorem 1.

Proof of Theorem 1. We define three events that will be critical to our proof, and we will show later on that each event occurs with high probability:

$$\Omega_0 = \left\{ \sup_{z \in \mathcal{M}} \frac{|\epsilon^\top z|}{\|z\|_2} \le \gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n} \right\}, \tag{3.28}$$

$$\Omega_1 = \left\{ \sup_{w \in \mathcal{R}^\perp} \frac{|\epsilon^\top w|}{\|D_{-S_0} w\|_1^{1/2} \|w\|_2^{1/2}} \le \gamma c_R (n s_0)^{1/4} \right\}, \tag{3.29}$$

$$\Omega_2 = \left\{ \sup_{\delta \in \mathcal{R}} \frac{|\epsilon^\top \delta|}{\|\delta\|_2} \le \gamma c_S \sqrt{s_0} \right\}, \tag{3.30}$$

where $\gamma > 1$ is parameter free to vary in our analysis, $c_I, c_R > 0$ are the constants in Lemmas 6, 7, and $c_S > 0$ is a constant to be determined below. Focusing on the third event, we will lower bound its probability by applying Theorem 22 to $\mathcal{X} = \mathcal{R} \cap \{\delta : \|\delta\|_n \le 1\}$. Note that

$$\log N(r, \mathcal{R} \cap \{\delta : \|\delta\|_n \le 1\}, \|\cdot\|_n) \le (s_0 + 1) \log(3/r),$$

as $\mathcal{R}$ is (s_0+1)-dimensional, and it is well-known that in $\mathbb{R}^d$, the number of balls of radius r that are needed to cover the unit ball is at most $(3/r)^d$. The quantity t_0 in Theorem 22 may be taken to be $t_0 = \inf\{r : (s_0+1)\log(3/r) \le C_3 C_1^2\} = 3\exp(-C_3 C_1^2/(s_0+1))$. The restrictions on t are hence $t > C_1$, as well as

$$t > C_2 \int_0^{t_0} \sqrt{(s_0+1)\log(3/r)}\, dr.$$

But, writing $\mathrm{erf}(\cdot)$ for the error function,

$$C_2 \int_0^{t_0} \sqrt{(s_0+1)\log(3/r)}\, dr = (\sqrt{s_0+1}) \cdot 3C_2 \left[r\sqrt{\log\frac{1}{r}} - \mathrm{erf}\left(\sqrt{\log\frac{1}{r}}\right) \right]\Bigg|_0^{t_0/3} \le C_2\sqrt{s_0},$$

where the constant $C_2 > 0$ is adjusted to be larger, as needed. Let us define $c_S = \max\{C_1, C_2\}$ and $C_S = C_4$. Then we have by Theorem 22, for $t = \gamma c_S \sqrt{s_0}$ and any $\gamma > 1$,

$$1 - 2\exp(-C_S \gamma^2 c_S^2 s_0) \le \mathbb{P}\left(\sup_{\delta \in \mathcal{R}} \frac{|\epsilon^\top \delta|}{\sqrt{n}\|\delta\|_n} \le \gamma c_S \sqrt{s_0} \right) = \mathbb{P}\left(\sup_{\delta \in \mathcal{R}} \frac{|\epsilon^\top \delta|}{\|\delta\|_2} \le \gamma c_S \sqrt{s_0} \right) = \mathbb{P}(\Omega_2). \tag{3.31}$$

The rest of this proof is divided into subparts for readability.

Basic inequality. The basic inequality in (3.12) is established by comparing objective values in (3.3) at $\widehat{\theta}$ and θ_0, writing $y = \theta_0 + \epsilon$, and rearranging. Using $\widehat{\theta} - \theta_0 = P_0(\widehat{\theta} - \theta_0) + P_1(\widehat{\theta} - \theta_0) = \widehat{\delta} + \widehat{x}$,

and using the fact that $\widehat{\delta}$ and $\widehat{x}$ are orthogonal, we have

$$\begin{aligned}
\|\widehat{\delta}\|_2^2 + \|\widehat{x}\|_2^2 &\le 2\epsilon^\top \widehat{\delta} + 2\epsilon^\top \widehat{x} + 2\lambda\big(\|D\theta_0\|_1 - \|D\widehat{\theta}\|_1\big) \\
&= 2\epsilon^\top \widehat{\delta} + 2\epsilon^\top \widehat{x} + 2\lambda\big(\|D_{S_0}\theta_0\|_1 - \|D_{S_0}\widehat{\theta}\|_1 - \|D_{-S_0}\widehat{\theta}\|_1\big) \\
&\le 2\epsilon^\top \widehat{\delta} + 2\epsilon^\top \widehat{x} + 2\lambda\big(\|D_{S_0}(\theta_0 - \widehat{\theta})\|_1 - \|D_{-S_0}\widehat{\theta}\|_1\big) \\
&\le 2\epsilon^\top \widehat{\delta} + 2\epsilon^\top \widehat{x} + 2\lambda\big(\|D_{S_0}\widehat{\delta}\|_1 + \|D_{S_0}\widehat{x}\|_1 - \|D_{-S_0}\widehat{x}\|_1\big) \\
&= \underbrace{2\epsilon^\top \widehat{\delta} + 2\lambda\|D_{S_0}\widehat{\delta}\|_1}_{A_0} + \underbrace{2\epsilon^\top \widehat{x} + 2\lambda\big(\|D_{S_0}\widehat{x}\|_1 - \|D_{-S_0}\widehat{x}\|_1\big)}_{B_0},
\end{aligned}$$

where in the third line, we used the triangle inequality, and in the fourth, we again used the triangle inequality and the fact that $D_{-S_0}\widehat{\delta} = 0$.

Bounding A_0. Note that

$$A_0 = 2\left(\frac{|\epsilon^\top \widehat{\delta}|}{\|\widehat{\delta}\|_2} + \lambda \frac{\|D_{S_0}\widehat{\delta}\|_1}{\|\widehat{\delta}\|_2}\right)\|\delta\|_2,$$

and observe

$$\begin{aligned}
\|D_{S_0}\widehat{\delta}\|_1 = \sum_{i=1}^{s_0} |\widehat{\delta}_{t_{i+1}} - \widehat{\delta}_{t_i}| \le 2\sum_{i=1}^{s_0+1} |\widehat{\delta}_{t_i}| &\le 2\sqrt{(s_0+1)\sum_{i=1}^{s_0+1} \widehat{\delta}_{t_i}^2} \\
&\le 4\sqrt{s_0 \sum_{i=1}^{s_0+1} \frac{t_i - t_{i-1}}{W_n}\widehat{\delta}_{t_i}^2} = 4\sqrt{\frac{s_0}{W_n}}\|\widehat{\delta}\|_2.
\end{aligned}$$

The second inequality used Cauchy-Schwartz, and the last equality used that $\widehat{\delta}$ is piecewise constant on the blocks $B_0, \ldots, B_{s_0}$, as $\widehat{\delta} \in \mathcal{R} = \mathrm{span}\{\mathbf{1}_{B_0}, \ldots, \mathbf{1}_{B_{s_0}}\}$. Hence, on the event Ω_2 in (3.30), we have

$$A_0 \le 2\left(\gamma c_S \sqrt{s_0} + 4\lambda\sqrt{\frac{s_0}{W_n}}\right)\|\widehat{\delta}\|_2 \tag{3.32}$$

Bounding B_0. In the definition of B_0, let us expand $\widehat{x} = \widehat{z} + \widehat{w}$, where $\widehat{z} \in \mathcal{M}$ is the lower interpolant to $\widehat{x}$, as defined in Lemma 5, and $\widehat{w} = \widehat{x} - \widehat{z}$ is the remainder. Using properties (3.13) and (3.14) from Lemma 5, we arrive at

$$\begin{aligned}
B_0 &= 2\epsilon^\top \widehat{z} + 2\epsilon^\top \widehat{w} + 2\lambda\big(\|D_{S_0}\widehat{z}\|_1 - \|D_{-S_0}\widehat{z}\|_1 - \|D_{-S_0}\widehat{w}\|_1\big) \\
&\le 2\epsilon^\top \widehat{z} + 8\lambda\sqrt{\frac{s_0}{W_n}}\|\widehat{z}\|_2 + 2\epsilon^\top \widehat{w} - 2\lambda\|D_{-S_0}\widehat{w}\|_1.
\end{aligned} \tag{3.33}$$

On the event Ω_0 in (3.28),

$$\epsilon^\top \widehat{z} \le \gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n} \|\widehat{z}\|_2.$$

And, on the event Ω_1 in (3.29), as $P_1\widehat{w} \in \mathcal{R}^\perp$, $\|D_{-S_0}P_1\widehat{w}\|_1 = \|D_{-S_0}\widehat{w}\|_1$, and $\|P_1\widehat{w}\|_2 \le \|\widehat{w}\|_2$,

$$\epsilon^\top P_1\widehat{w} \le \gamma c_R (ns_0)^{1/4} \|D_{-S_0}\widehat{w}\|_1^{1/2} \|\widehat{w}\|_2^{1/2},$$

Also, on the event Ω_2 in (3.30), since $P_0\widehat{w} \in \mathcal{R}$,

$$\epsilon^\top P_0\widehat{w} \le \gamma c_S \sqrt{s_0} \|\widehat{w}\|_2.$$

Hence, on $\Omega_0 \cap \Omega_1 \cap \Omega_2$, combining the last three displays with (3.33),

$$B_0 \le 2\left(\gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n} + 4\lambda\sqrt{\frac{s_0}{W_n}}\right)\|\widehat{z}\|_2 + 2\gamma c_S\sqrt{s_0}\|\widehat{w}\|_2 + \\ 2\gamma c_R (ns_0)^{1/4}\|D_{-S_0}\widehat{w}\|_1^{1/2}\|\widehat{w}\|_2^{1/2} - 2\lambda\|D_{-S_0}\widehat{w}\|_1. \quad (3.34)$$

Consider the first case in which $\gamma c_R (ns_0)^{1/4}\|D_{-S_0}\widehat{w}\|_1^{1/2}\|\widehat{w}\|_2^{1/2} \ge \lambda\|D_{-S_0}\widehat{w}\|_1$. Then

$$\|D_{-S_0}\widehat{w}\|_1 \le \left(\frac{\gamma c_R}{\lambda}\right)^2 \sqrt{ns_0}\|\widehat{w}\|_2,$$

and from (3.34), on the event $\Omega_0 \cap \Omega_1 \cap \Omega_2$,

$$B_0 \le 2\left(\gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n} + 4\lambda\sqrt{\frac{s_0}{W_n}} + \gamma c_S\sqrt{s_0} + \frac{\gamma^2 c_R^2 \sqrt{ns_0}}{\lambda}\right)\|\widehat{x}\|_2. \quad (3.35)$$

where we have used (3.15). In the case $\gamma c_R (ns_0)^{1/4}\|D_{-S_0}\widehat{w}\|_1^{1/2}\|\widehat{w}\|_2^{1/2} < \lambda\|D_{-S_0}\widehat{w}\|_1$, we have from (3.34), on the event $\Omega_0 \cap \Omega_1 \cap \Omega_2$,

$$B_0 \le 2\left(\gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n} + 4\lambda\sqrt{\frac{s_0}{W_n}} + \gamma c_S\sqrt{s_0}\right)\|\widehat{x}\|_2.$$

Therefore, the bound (3.35) always holds on the event $\Omega_0 \cap \Omega_1 \cap \Omega_2$.

Putting it all together. Combining (3.32) and (3.35), we see that on $\Omega_0 \cap \Omega_1 \cap \Omega_2$,

$$\|\widehat{\delta}\|_2^2 + \|\widehat{x}\|_2^2 \le 2\left(\gamma c_S\sqrt{s_0} + 4\lambda\sqrt{\frac{s_0}{W_n}}\right)\|\widehat{\delta}\|_2 + \\ 2\left(\gamma c_I \sqrt{(\log s_0 + \log\log n) s_0 \log n} + 4\lambda\sqrt{\frac{s_0}{W_n}} + \gamma c_S\sqrt{s_0} + \frac{\gamma^2 c_R^2 \sqrt{ns_0}}{\lambda}\right)\|\widehat{x}\|_2.$$

Denote the right-hand side by $A_1\|\widehat{\delta}\|_2 + B_1\|\widehat{x}\|_2$. Using the simple inequality $2ab \le a^2 + b^2$, twice, we have on $\Omega_0 \cap \Omega_1 \cap \Omega_2$,

$$\|\widehat{\delta}\|_2^2 + \|\widehat{x}\|_2^2 \le \frac{A_1^2}{2} + \frac{\|\widehat{\delta}\|_2^2}{2} + \frac{B_1^2}{2} + \frac{\|\widehat{x}\|_2^2}{2}.$$

Recalling that $\|\widehat{\delta}\|_2^2 + \|\widehat{x}\|_2^2 = \|\widehat{\theta} - \theta_0\|_2^2$, this implies that on the event $\Omega_0 \cap \Omega_1 \cap \Omega_2$, there exists a constant $c > 0$, such that for large enough n, and any $\gamma > 1$,

$$\|\widehat{\theta} - \theta_0\|_2^2 \le \gamma^4 c s_0 \left((\log s_0 + \log\log n)\log n + \frac{\lambda^2}{W_n} + \frac{n}{\lambda^2} \right), \tag{3.36}$$

on the event $\Omega_0 \cap \Omega_1 \cap \Omega_2$. Furthermore, using the union bound along with Lemmas 6, 7, and (3.31), we find that

$$\begin{aligned}\mathbb{P}\big((\Omega_0 \cap \Omega_1 \cap \Omega_2)^c\big) \le 2\exp\big(-C_I\gamma^2 c_I^2(\log s_0 + \log\log n)\big) + \\ 2\exp(-C_R\gamma^2 c_R^2\sqrt{s_0}) + 2\exp(-C_S\gamma^2 c_S^2 s_0) \le \exp(-C\gamma^2),\end{aligned}$$

for an appropriately defined constant $C > 0$. Optimizing the bound in (3.36) to choose the tuning parameter λ yields $\lambda = (nW_n)^{1/4}$. Plugging this in gives the final result. □

Next we give the proof of Corollary 2.

Proof of Corollary 2. Define the random variable

$$Z = \frac{\|\widehat{\theta} - \theta_0\|_2^2}{c s_0((\log s_0 + \log\log n)\log n + \sqrt{n/W_n})},$$

which we know has the tail bound $\mathbb{P}(Z > z) \le \exp(-C\sqrt{z})$ for $z > 1$, and observe that

$$\mathbb{E}(Z) = \int_0^\infty \mathbb{P}(Z > z)\,dz \le 1 + \int_1^\infty \exp(-C\sqrt{z})\,dz.$$

The right-hand side is a finite constant, and this gives the result

$$\mathbb{E}\|\widehat{\theta} - \theta_0\|_n^2 \le c\frac{s_0}{n}\left((\log s_0 + \log\log n)\log n + \sqrt{\frac{n}{W_n}} \right),$$

where the constant $c > 0$ is adjusted to be larger, as needed. □

3.A.2 Proofs of Lemma 5, Lemma 6, Lemma 7, and (3.43)

Proof of Lemma 5. We give an explicit construction of a lower interpolant $z \in \mathcal{M}$ to x, given the changepoints $0 = t_0 < \ldots < t_{s_0+1} = n$. We will use the notation $a_+ = \max\{0, a\}$ for the positive part of a. For $i = 0, \ldots, s_0$, define $z^{(i+)} \in \mathbb{R}^{t_{i+1}-t_i}$ by setting $g_i^+ = \text{sign}(x_{t_i})$ and

$$z_j^{(i+)} = g_i^+ \cdot \min\Big\{(g_i^+ x_{t_i+1})_+, \ldots, (g_i^+ x_{t_i+j})_+\Big\}, \quad j = 1, \ldots, t_{i+1} - t_i.$$

Similarly, define $z^{(i-)} \in \mathbb{R}^{t_{i+1}-t_i}$ by setting $g_i^- = \text{sign}(x_{t_{i+1}-1})$ and

$$z_j^{(i-)} = g_i^- \cdot \min\Big\{(g_i^- x_{t_i+j})_+, \ldots, (g_i^- x_{t_{i+1}})_+\Big\}, \quad j = 1, \ldots, t_{i+1} - t_i.$$

Note that $z_1^{(i+)} = x_{t_i+1}$ and $z_{t_{i+1}-t_i}^{(i-)} = x_{t_{i+1}}$; also, $\{|z_j^{(i+)}|\}_{j=1}^{t_{i+1}-t_i}$ is a nonincreasing sequence, and $\{|z_j^{(i-)}|\}_{j=1}^{t_{i+1}-t_i}$ is nondecreasing. Furthermore,

$$\text{sign}\left(z_1^{(i+)}\right) \cdot \text{sign}\left(z_j^{(i+)}\right) \geq 0 \quad \text{and} \quad \text{sign}\left(z_{t_{i+1}-t_i}^{(i-)}\right) \cdot \text{sign}\left(z_j^{(i-)}\right) \geq 0, \quad j = 1, \ldots, t_{i+1} - t_i.$$

Lastly, notice that there exists a point $j' \in 1, \ldots, t_{i+1} - t_i - 1$ (not necessarily unique) such that

$$\min_{k \in \{1,\ldots,t_{i+1}-t_i\}} \left|z_k^{(i+)}\right| = \left|z_{j'+1}^{(i+)}\right| = \left|z_j^{(i+)}\right|, \quad j = j'+1, \ldots, t_{i+1} - t_i, \tag{3.37}$$

$$\min_{k \in \{1,\ldots,t_{i+1}-t_i\}} \left|z_k^{(i-)}\right| = \left|z_{j'}^{(i-)}\right| = \left|z_j^{(i-)}\right|, \quad j = 1, \ldots, j'. \tag{3.38}$$

We define $z_{t_i+j} = z_j^{(i+)}$ for $j = 1, \ldots, j'$, and $z_{t_i+j} = z_j^{(i-)}$ for $j = j'+1, \ldots, t_{i+1} - t_i$. Letting $t_i' = t_i + j'$ and repeating this process for $i = 0, \ldots, s_0$, we have constructed $z \in \mathcal{M}$.

We now verify the claimed properties for the constructed lower interpolant z. For $i = 0, \ldots, s_0$, and any $j = 1, \ldots, t_{i+1} - t_i$, we have

$$\text{sign}(z_j^{(i+)}) \cdot \text{sign}(x_{t_i+j}) \geq 0, \tag{3.39}$$

$$|z_j^{(i+)}| \leq |x_{t_i+j}|, \tag{3.40}$$

Further, for any $j = 1, \ldots, t_{i+1} - t_i - 1$,

$$\text{sign}\left((Dz^{(i+)})_j\right) \cdot \text{sign}\left((Dx)_{t_i+j}\right) \geq 0, \tag{3.41}$$

$$\left|(Dz^{(i+)})_j\right| \leq \left|(Dx)_{t_i+j}\right|. \tag{3.42}$$

To see why (3.41) holds, note that the properties $\text{sign}(Dz^{(i+)})_j \in \{-1, 0\}$, $(Dz^{(i+)})_j < 0$ imply $(D(g_i^+ x)_+)_{t_i+j} < 0$. To see why (3.42) holds, if $(Dz^{(i+)})_j \neq 0$, then we know that

$$|z_{j+1}^{(i+)} - z_j^{(i+)}| \leq \left|\min\Big\{(g_i^+ x_{t_i+j+1})_+, (g_i^+ x_{t_i+j})_+\Big\} - (g_i^+ x_{t_i+j})_+\right| \leq |x_{t_i+j+1} - x_{t_i+j}|,$$

where we used the observation that $|\min\{a,b\} - b| \geq |\min\{a,b,c\} - \min\{b,c\}|$.

It can be shown by nearly equivalent steps that $z^{(i-)}$, z both satisfy properties analogous to (3.39)–(3.42). Using (3.39) and (3.40) on z gives (3.15). Using (3.41) and (3.42) on z gives (3.13) (note that if $\text{sign}(a) = \text{sign}(b)$ and $|a| > |b|$, then $|a| = |b| + |a - b|$). Because $z_{t_i+1} = x_{t_i+1}$ and $z_{t_{i+1}} = x_{t_{i+1}}$ for all $i = 0, \ldots, s_0$, we have the equality in (3.14) (as $D_{t_i} z = z_{t_i+1} - z_{t_i} = x_{t_i+1} - x_{t_i} = D_{t_i} x$).

Finally, for each $i = 0, \ldots, s_0$, define $t''_i = t'_i$ if $|z_{t'_i}| \geq |z_{t'_i+1}|$ and $t''_i = t'_i + 1$ otherwise. Observe that by (3.37) and (3.38), it holds that $|z_{t''_i}| = \min_{j=1,\ldots,t_{i+1}-t_i} |z_{t_i+j}|$. The inequality in (3.14) is finally established by the following chain of inequalities:

$$\begin{aligned}
\|D_{S_0} z\|_1 &= \sum_{i=1}^{s_0} |z_{t_i+1} - z_{t_i}| \leq \sum_{i=1}^{s_0} |z_{t_i+1}| + |z_{t_i}| \\
&= \sum_{i=1}^{s_0} \big(|z_{t_i+1}| - |z_{t''_i}|\big) + \big(|z_{t_i}| - |z_{t''_{i-1}}|\big) + |z_{t''_{i-1}}| + |z_{t''_i}| \\
&\leq \|D_{-S_0} z\|_1 + 2 \sum_{i=0}^{s_0} |z_{t''_i}| \leq \|D_{-S_0} z\|_1 + 4\sqrt{\frac{s_0}{W_n}} \|z\|_2,
\end{aligned}$$

where in the second inequality, we used $|a| - |c| \leq |a - c| \leq |a - b| + |b - c|$, and in the last inequality, we used the above property of $z_{t''_i}$ and

$$\sum_{i=0}^{s_0} |z_{t''_i}| \leq 2\sqrt{s_0} \sqrt{\sum_{i=0}^{s_0} |z_{t''_i}|^2} \leq 2\sqrt{s_0 \sum_{i=0}^{s_0} \frac{t_{i+1} - t_i}{W_n} z_{t''_i}^2} \leq 2\sqrt{\frac{s_0}{W_n}} \|z\|_2.$$

This completes the proof. □

Proof of Lemma 6. We consider $\epsilon \in \mathbb{R}^n$, an i.i.d. sub-Gaussian vector as referred to in the statement of the lemma, and arbitrary $z \in \mathcal{M}$. In this proof, we will also consider $E(t)$ and $Z(t)$, real-valued functions over $[0, n]$, constructed so that $E(t) = \epsilon_{\lceil t \rceil}$ for all t (i.e., $E(t)$ is a step function), $Z(t) = z_t$ for $t = 1, \ldots, n$, and $Z(t)$ is continuously differentiable and monotone over $(t_i, t'_i]$ and $(t'_i, t_{i+1}]$ for $i = 0, \ldots, s_0$. These functions will also satisfy the boundary conditions $E(0) = \epsilon_1$ and $Z(0) = z_1$.

Let $F(t) = \int_0^t E(u)\, du$. As ϵ is random, $E(t)$ and $F(t)$ are also random. It can be shown that there exists constants $c_I, C_I > 0$ such that for any $\gamma > 1$,

$$\mathbb{P}\left(\frac{|F(t) - F(t_i)|}{\sqrt{|t - t_i|}} \leq \gamma c_I \sqrt{\log s_0 + \log\log n}, \text{ for } t \in (t_i, t_{i+1}],\ i = 0, \ldots, s_0 \right) \\ \geq 1 - 2\exp\big(- C_I \gamma^2 c_I^2 (\log s_0 + \log\log n)\big). \quad (3.43)$$

So as not to distract from the main flow of ideas, we now proceed to prove Lemma 6, and we provide a proof of (3.43) later. Let Ω_3 denote the event in consideration on the left-hand side of (3.43). By integration by parts,

$$\int_{t_i}^{t_i'} E(t)Z(t)\,dt = Z(t_i')(F(t_i') - F(t_i)) - \int_{t_i}^{t_i'} Z'(t)(F(t) - F(t_i))\,dt$$

where $Z'(t) = \frac{d}{dt}Z(t)$. Thus, on the event Ω_3,

$$\left|\int_{t_i}^{t_i'} E(t)Z(t)\,dt\right| \leq \gamma c_I \sqrt{\log s_0 + \log\log n}\left(|Z(t_i')|\sqrt{t_i' - t_i} + \left|\int_{t_i}^{t_i'} Z'(t)\sqrt{t - t_i}\,dt\right|\right), \tag{3.44}$$

since Z' does not change sign within the intervals $(t_i, t_i'], (t_i', t_{i+1}]$ (as $z \in \mathcal{M}$). For n large enough, we can upper bound the last term in (3.44) as follows

$$\left|\int_{t_i}^{t_i'} Z'(t)\sqrt{t - t_i}\,dt\right| \leq \left|\int_{t_i}^{t_i+n^{-1}} Z'(t)\sqrt{t - t_i}\,dt\right| + \left|\int_{t_i+n^{-1}}^{t_i'} Z'(t)\sqrt{t - t_i}\,dt\right|. \tag{3.45}$$

Using integration by parts and the triangle inequality on the second term in (3.45),

$$\left|\int_{t_i+n^{-1}}^{t_i'} Z'(t)\sqrt{t - t_i}\,dt\right| \leq |Z(t_i')|\sqrt{t_i' - t_i} + \left|\frac{Z(t_i + n^{-1})}{\sqrt{n}}\right| + \left|\int_{t_i+n^{-1}}^{t_i'} \frac{Z(t)}{\sqrt{t - t_i}}\,dt\right|. \tag{3.46}$$

By Cauchy-Schwartz on the last term in (3.46),

$$\begin{aligned}\left|\int_{t_i+n^{-1}}^{t_i'} \frac{Z(t)}{\sqrt{t - t_i}}\,dt\right| &\leq \left(\int_{t_i+n^{-1}}^{t_i'} Z(t)^2\,dt\right)^{1/2}\left(\int_{t_i+n^{-1}}^{t_i'} \frac{1}{t - t_i}\,dt\right)^{1/2} \\ &\leq \left(\int_{t_i+n^{-1}}^{t_i'} Z(t)^2\,dt\right)^{1/2}\sqrt{2\log n}. \end{aligned} \tag{3.47}$$

Now examining the first term in (3.45),

$$\left|\int_{t_i}^{t_i+n^{-1}} Z'(t)\sqrt{t - t_i}\,dt\right| \leq n^{-1/2}\left|\int_{t_i}^{t_i+n^{-1}} Z'(t)\,dt\right| = \frac{|Z(t_i + n^{-1}) - Z(t_i)|}{\sqrt{n}}.$$

But because we only require Z to be piecewise monotonic and continuously differentiable then we are at liberty to make $Z(t_i + n^{-1}) = Z(t_i)$, forcing this term to be 0. In order to bound $Z(t_i')$, notice that because $|Z(t)|$ is non-increasing over the interval $(t_i, t_i']$ we have that

$$Z(t_i')^2|t_i' - t_i| \leq \int_{t_i}^{t_i'} Z(t)^2\,dt. \tag{3.48}$$

Combining (3.44)–(3.48), we have that on the event Ω_3,

$$\left|\int_{t_i}^{t_i'} E(t)Z(t)\,dt\right| \le \alpha_n\left(2+\sqrt{\frac{\log n}{2}}\right)\left(\int_{t_i}^{t_i'} Z(t)^2\,dt\right)^{1/2} + \alpha_n\frac{|Z(t_i)|}{\sqrt{n}}. \tag{3.49}$$

where we have abbreviated $\alpha_n = \gamma c_I\sqrt{\log s_0 + \log\log n}$. Through nearly identical steps we can show that on the event Ω_3,

$$\left|\int_{t_i'}^{t_{i+1}} E(t)Z(t)\,dt\right| \le \alpha_n\left(2+\sqrt{\frac{\log n}{2}}\right)\left(\int_{t_i'}^{t_{i+1}} Z(t)^2\,dt\right)^{1/2} + \alpha_n\frac{|Z(t_{i+1})|}{\sqrt{n}}. \tag{3.50}$$

Therefore

$$\begin{aligned}\left|\int_0^n E(t)Z(t)\,dt\right| &\le \sum_{i=0}^{s_0}\left(\left|\int_{t_i}^{t_i'} E(t)Z(t)\,dt\right| + \left|\int_{t_i'}^{t_{i+1}} E(t)Z(t)\,dt\right|\right)\\ &\le \alpha_n\sqrt{2s_0+2}\left(2+\sqrt{\frac{\log n}{2}}\right)\left(\int_0^n Z(t)^2\,dt\right)^{1/2} + 2\alpha_n\frac{\|z\|_1}{\sqrt{n}},\end{aligned} \tag{3.51}$$

where in the second line we used (3.49), (3.50), and the Cauchy-Schwartz inequality. Because we can choose $Z(t)$ to be arbitrarily close to $z_{\lceil t\rceil}$ over all t, we can make the integral $(\int_0^n Z(t)^2\,dt)^{1/2}$ arbitrarily close to $\|z\|_2$ and likewise we can make $\int_0^n E(t)Z(t)\,dt$ arbitrarily close to $\epsilon^\top z$. Furthermore, because $\|z\|_1 \le \sqrt{n}\|z\|_2$, the first term in (3.51) dominates. Hence on the event Ω_3, we have established that

$$|\epsilon^\top z| \le \gamma c_I\sqrt{(\log s_0 + \log\log n)s_0\log n}\|z\|_2,$$

where the constant c_I is adjusted to be larger, as needed. Noting that the event Ω_3 does not depend on z, the result follows. □

Proof of claim (3.43). We will construct a covering for $\mathcal{V} = \cup_{i=0}^{s_0}\mathcal{V}_i$, where for each $i = 0,\ldots,s_0$,

$$\mathcal{V}_i = \left\{\sqrt{\frac{n}{|A|}}\mathbf{1}_A : A = \{t_i,\ldots,t\},\ t = t_i+1,\ldots,n\right\} \cup \left\{\sqrt{\frac{n}{|A|}}\mathbf{1}_A : A = \{t,\ldots,t_i\},\ t = 1,\ldots,t_i-1\right\}.$$

Our scaling is such that, for any $a = \sqrt{n/|A|}\mathbf{1}_A$, where $A \subseteq \{1,\ldots,n\}$, we have $\|a\|_n = 1$. Further, for any other $b = \sqrt{n/|B|}\mathbf{1}_B$, where $B \subseteq \{1,\ldots,n\}$, we have

$$\|a-b\|_n^2 = \frac{|A\cap B|}{(\sqrt{|A|}-\sqrt{|B|})^2} + \frac{|A\setminus B|}{|A|} + \frac{|B\setminus A|}{|B|} = 2\left(1-\frac{|A\cap B|}{\sqrt{|A||B|}}\right). \tag{3.52}$$

We first construct a covering for each set $\mathcal{V}_i$, $i = 0, \ldots, s_0$, and we restrict our attention to a radius $0 < r < \sqrt{2}$. Let $\alpha = \lceil (1 - r^2/2)^{-2} \rceil$, and consider the set

$$\mathcal{C}_i = \left\{ \sqrt{\frac{n}{|A|}} \mathbf{1}_A : A = \{t_i, \ldots, \min\{t_i + \alpha^j, n\}\},\ j = 1, \ldots, \lceil \log n / \log \alpha \rceil \right\} \cup$$
$$\left\{ \sqrt{\frac{n}{|A|}} \mathbf{1}_A : A = \{\max\{t_i - \alpha^j, 1\}, \ldots, t_i\},\ j = 1, \ldots, \lceil \log n / \log \alpha \rceil \right\}.$$

Here, the set $\mathcal{C}_i$ has at most $2\lceil \log n / \log \alpha \rceil \leq 4 \log n / \log \alpha$ elements, and by (3.52), balls of radius r around elements in $\mathcal{C}_i$ cover the set $\mathcal{V}_i$. This establishes that

$$N(r, \mathcal{V}_i, \|\cdot\|_n) \leq \frac{-2 \log n}{\log(1 - r^2/2)}. \tag{3.53}$$

For a radius $0 < r < \sqrt{2}$, the covering number for $\mathcal{V} = \cup_{i=0}^{s_0} \mathcal{V}_i$ can be obtained by simply taking a union of the covers in (3.53) over $i = 0, \ldots, s_0$, giving

$$N(r, \mathcal{V}, \|\cdot\|_n) \leq \sum_{i=0}^{s_0} N(r, \mathcal{V}_i, \|\cdot\|_n) \leq 2(s_0 + 1)\left(\frac{-\log n}{\log(1 - r^2/2)} \right). \tag{3.54}$$

Using (3.52) once more, the diameter of the set $\mathcal{V}$ is $\sqrt{2}$, hence if $r \geq 1/\sqrt{2}$, then we need only 1 ball to cover $\mathcal{V}$. Combining this fact with (3.54), we obtain

$$N(r, \mathcal{V}, \|\cdot\|_n) \leq \begin{cases} 2(s_0 + 1)\left(\dfrac{-\log n}{\log(1 - r^2/2)} \right) & \text{if } 0 < r < 1/\sqrt{2} \\ 1 & \text{if } r \geq 1/\sqrt{2} \end{cases}. \tag{3.55}$$

Now let us apply Theorem 22, with $\mathcal{X} = \mathcal{V}$. First, we remark that the quantity t_0 in Theorem 22 may be taken to be $t_0 = 1/\sqrt{2}$. The bounds on t in the theorem are $t > C_1$, as well as

$$t > C_2 \int_0^{1/\sqrt{2}} \sqrt{\log\left(2(s_0 + 1) \frac{-\log n}{\log(1 - r^2/2)} \right)}\, dr.$$

Next, we know that the right-hand side above is upper bounded by

$$C_2 \int_0^{1/\sqrt{2}} \left[\sqrt{\log\left(2(s_0 + 1)\log n\right)} + \sqrt{\log\left(\frac{-1}{\log(1 - r^2/2)} \right)} \right] dr$$
$$= C_2 \sqrt{\frac{\log\left(2(s_0 + 1)\log n\right)}{2}} + C_2 \sqrt{2} \int_0^{1/2} \sqrt{\log\left(\frac{1}{\log\left(\frac{1}{1 - x^2}\right)} \right)}\, dx.$$

One can verify that the the integral in the second term above converges to a finite constant (upper bounded by 1 in fact). Thus the entire expression above is upper bounded by $C_2\sqrt{\log s_0 + \log\log n}$, where the constant $C_2 > 0$ is adjusted to be larger, as needed. Therefore, letting $c_I = \max\{C_1, C_2\}$, we may restrict our attention to $t > c_I\sqrt{\log s_0 + \log\log n}$ in Theorem 22, and letting $C_I = C_4$, the conclusion reads, for $t = \gamma c_I$ and $\gamma > 1$,

$$\mathbb{P}\bigg(\sup_{a\in\mathcal{V}} \frac{\epsilon^\top a}{\sqrt{n}} > \gamma c_I\sqrt{\log s_0 + \log\log n}\bigg) \le 2\exp\big(-C_I\gamma^2 c_I^2(\log s_0 + \log\log n)\big).$$

Recalling the form of $a = \sqrt{n/|A|}\mathbf{1}_A \in \mathcal{V}$, the above may be rephrased as

$$\mathbb{P}\bigg(\frac{\sum_{j=t_i}^{\top}\epsilon_j}{\sqrt{|t-t_i|}} > \gamma c_I\sqrt{\log s_0 + \log\log n},\ \text{for } t = 1,\dots,n,\ i = 0,\dots,s_0\bigg)$$
$$\le 2\exp\big(-C_I\gamma^2 c_I^2(\log s_0 + \log\log n)\big). \quad (3.56)$$

Finally, consider the following event

$$\Omega_4 = \bigg\{\frac{|F(t) - F(t_i)|}{\sqrt{|t-t_i|}} \le \gamma c_I\sqrt{\log s_0 + \log\log n},\ \text{for } t = 1,\dots,n,\ i = 0,\dots,s_0\bigg\}.$$

Recalling that $E(t) = \epsilon_{\lceil t\rceil}$ for all $t \in [0,1]$, we have $F(t) = \int_0^t E(u)\,du = \sum_{j=0}^t \epsilon_j$ for $t = 1,\dots,n$. In (3.56), we have thus shown $\mathbb{P}(\Omega_4) \ge 1 - 2\exp(-C_I\gamma^2 c_I^2(\log s_0 + \log\log n))$. Note that $|F(t) - F(t_i)|$ is piecewise linear with knots at $t = 1,\dots,n$ and $\sqrt{|t-t_i|}$ is concave in between these knots, so if $|F(t) - F(t_i)|/\sqrt{|t-t_i|} \le \gamma c_I\sqrt{\log s_0 + \log\log n}$ for $t = 1,\dots,n$, then the same bound must hold over all $t \in [0,n]$. This shows that $\Omega_4 \supseteq \Omega_3$, where Ω_3 is the event in question in the left-hand side of (3.43); in other words, we have verified (3.43). □

For the proof of Lemma 7, we will need the following result from van de Geer (1990).

Lemma 23 (**Lemma 3.5 of van de Geer 1990**). *Assume the setting of Theorem 22, and additionally, assume that for some $\zeta \in (0,1)$ and $K > 0$,*

$$\mathcal{K}(r) \le Kr^{-2\zeta},$$

where, recall, $\mathcal{K}(r)$ is a continuous function upper bounding the entropy number $\log N(r, \mathcal{X}, \|\cdot\|_n)$. Then there exists constants C_0, C_1 depending only on σ such that for any $t \ge C_0$,

$$\mathbb{P}\bigg(\sup_{x\in\mathcal{X}} \frac{|\epsilon^\top x|}{\sqrt{n}\|x\|_n^{1-\zeta}} > t\sqrt{K}\bigg) \le \exp(-C_1 t^2 K).$$

Proof of Lemma 7. Recall that for $i = 0, \ldots, s_0$, we let $B_i = \{t_i + 1, \ldots, t_{i+1}\}$. For $i = 0, \ldots, s_0$, also define $n_i = |B_i|$, the scaled norm $\|\cdot\|_{n_i} = \|\cdot\|_2/\sqrt{n_i}$, and

$$\mathcal{X}_i = \Big\{ w^{(i)} \in \mathbb{R}^{n_i} : (\mathbf{1}^{(i)})^\top w^{(i)} = 0,\ \|D^{(i)} w^{(i)}\|_1 \le 1,\ \|w^{(i)}\|_{n_i} \le 1 \Big\}.$$

Here, we write $\mathbf{1}^{(i)} \in \mathbb{R}^{n_i}$ for the vector of all 1s, and $D^{(i)} \in \mathbb{R}^{(n_i-1)\times n}$ for the difference operator, as in (3.6) but of smaller dimension. The set $\mathcal{X}_i$ is the discrete total variation space in $\mathbb{R}^{n_i}$, where all elements are centered and have scaled norm at most 1. From well-known results on entropy bounds for total variation spaces (e.g., from Lemma 11 and Corollary 12 of Wang et al. (2017)), we have

$$\log N(r, \mathcal{X}_i, \|\cdot\|_{n_i}) \le \frac{C}{r},$$

for a universal constant $C > 0$. Hence we may apply Lemma 23 with $\mathcal{X} = \mathcal{X}_i$ and $\zeta = 1/2$: for the random variable

$$M_i = \sup \left\{ \frac{|\epsilon_{B_i}^\top w^{(i)}|}{\sqrt{n_i}\|w^{(i)}\|_{n_i}^{1/2}} : w^{(i)} \in \mathcal{X}_i \right\},$$

we may take $t = \gamma C_0$ in the lemma, for any $\gamma > 1$, and conclude that

$$\mathbb{P}\Big(M_i > \gamma C_0 \sqrt{C} \Big) \le \exp(-C_1 \gamma^2 C_0^2 C).$$

Notice that we may rewrite M_i as

$$M_i = \sup \left\{ \frac{|\epsilon_{B_i}^\top w^{(i)}|}{n_i^{1/4} \|D^{(i)} w^{(i)}\|_1^{1/2} \|w^{(i)}\|_2^{1/2}} : w^{(i)} \in \mathbb{R}^{n_i},\ (\mathbf{1}^{(i)})^\top w^{(i)} = 0 \right\},$$

and therefore

$$\mathbb{P}\Bigg(\sup_{w^{(i)} \in \mathbb{R}^{n_i},\, (\mathbf{1}^{(i)})^\top w^{(i)} = 0} \frac{|\epsilon_{B_i}^\top w^{(i)}|}{\|D^{(i)} w^{(i)}\|_1^{1/2} \|w^{(i)}\|_2^{1/2}} > \gamma C_0 \sqrt{C} n_i^{1/4} \Bigg) \le \exp(-C_1 \gamma^2 C_0^2 C).$$

Using the union bound,

$$\mathbb{P}\left(\sup_{\substack{w^{(i)} \in \mathbb{R}^{n_i},\, (\mathbf{1}^{(i)})^\top w^{(i)} = 0 \\ i = 0, \ldots, s_0}} \frac{|\epsilon_{B_i}^\top w^{(i)}|}{\|D^{(i)} w^{(i)}\|_1^{1/2} \|w^{(i)}\|_2^{1/2}} > \gamma C_0 \sqrt{C} n_i^{1/4} \right) \le (s_0 + 1) \exp(-C_1 \gamma^2 C_0^2 C).$$

Define the constants $c_R = \max\{C_0\sqrt{C}, 1\}$ and $C_R = \max\{C_1/2, 1\}$. This ensures that we have $2C_R\gamma^2 c_R^2\sqrt{s_0} \geq \log(s_0+1)$ for any $\gamma > 1$ and any s_0, thus

$$\mathbb{P}\left(\sup_{\substack{w^{(i)}\in\mathbb{R}^{n_i},\,(\mathbf{1}^{(i)})^\top w^{(i)}=0\\ i=0,\ldots,s_0}} \frac{|\epsilon_{B_i}^\top w^{(i)}|}{\|D^{(i)}w^{(i)}\|_1^{1/2}\|w^{(i)}\|_2^{1/2}} > \gamma c_R(n_i s_0)^{1/4}\right) \leq \exp(-C_R\gamma^2 c_R^2\sqrt{s_0}).$$

The proof is completed by noting the following: if $w \in \mathcal{R}^\perp$, then $(\mathbf{1}^{(i)})^\top w_{B_i} = 0$ for $i = 0, \ldots, s_0$, and so on the event in consideration in the last display,

$$\begin{aligned}
|\epsilon^\top w| &\leq \sum_{i=0}^{s_0} |\epsilon_{B_i}^\top w_{B_i}| \leq \gamma c_R s_0^{1/4} \sum_{i=0}^{s_0} n_i^{1/4}\|D^{(i)}w_{B_i}\|_1^{1/2}\|w_{B_i}\|_2^{1/2} \\
&\leq \gamma c_R s_0^{1/4}\left(\sum_{i=0}^{s_0}\|D^{(i)}w_{B_i}\|_1\right)^{1/2}\left(\sum_{i=0}^{s_0} n_i^{1/2}\|w_{B_i}\|_2\right)^{1/2} \\
&= \gamma c_R s_0^{1/4}\|D_{-S_0}w\|_1^{1/2}\left(\sum_{i=0}^{s_0} n_i^{1/2}\|w_{B_i}\|_2\right)^{1/2} \\
&\leq \gamma c_R s_0^{1/4}\|D_{-S_0}w\|_1^{1/2}\left(\sum_{i=0}^{s_0}\|w_{B_i}\|_2^2\right)^{1/4}\left(\sum_{i=0}^{s_0} n_i\right)^{1/4} \\
&= \gamma c_R s_0^{1/4}\|D_{-S_0}w\|_1^{1/2}\|w\|_2^{1/2} n^{1/4},
\end{aligned}$$

by two successive uses of Cauchy-Schwartz. □

3.A.3 Proof of Theorem 8

Let $\widehat{\theta}$ denote the fused lasso estimate in (3.3), and $\widetilde{\theta} \in \mathbb{R}^n$ denote an arbitrary vector. By subgradient optimality, we know that $y - \widehat{\theta} = \lambda g$ for a subgradient $g \in \mathbb{R}^n$ of the function $x \mapsto \|Dx\|_1$ evaluated at $x = \widehat{\theta}$. Thus,

$$(y - \widehat{\theta})^\top\widehat{\theta} = \lambda\|D\widehat{\theta}\|_1.$$

Furthermore,

$$(y - \widehat{\theta})^\top\widetilde{\theta} \leq \lambda\|D\widetilde{\theta}\|_1.$$

Subtracting the second to last equation from the last gives

$$(y - \widehat{\theta})^\top(\widetilde{\theta} - \widehat{\theta}) \leq \lambda\big(\|D\widetilde{\theta}\|_1 - \|D\widehat{\theta}\|_1\big),$$

or

$$(\theta_0 - \widehat{\theta})^\top(\widetilde{\theta} - \widehat{\theta}) \le \epsilon^\top(\widetilde{\theta} - \widehat{\theta}) + \lambda\big(\|D\widetilde{\theta}\|_1 - \|D\widehat{\theta}\|_1\big).$$

Using the polarization identity $2a^\top b = \|a\|_2^2 + \|b\|_2^2 - \|a-b\|_2^2$ gives

$$\|\widehat{\theta} - \theta_0\|_2^2 + \|\widetilde{\theta} - \widehat{\theta}\|_2^2 - \|\theta_0 - \widetilde{\theta}\|_2^2 \le 2\epsilon^\top(\widetilde{\theta} - \widehat{\theta}) + 2\lambda\big(\|D\widetilde{\theta}\|_1 - \|D\widehat{\theta}\|_1\big).$$

As this holds for any $\widetilde{\theta}$, we can take $\widetilde{\theta} = \theta_0(s)$ in particular, and rearrange, to find that

$$\|\widehat{\theta} - \theta_0\|_2^2 + \|\widehat{\theta} - \theta_0(s)\|_2^2 \le \|\theta_0(s) - \theta_0\|_2^2 + 2\epsilon^\top(\theta_0(s) - \widehat{\theta}) + 2\lambda\big(\|D\theta_0(s)\|_1 - \|D\widehat{\theta}\|_1\big). \quad (3.57)$$

The right-hand side above can be handled just as in the proof of Theorem 1. Dropping $\|\theta_0(s) - \widehat{\theta}\|_2^2$ from the left-hand side above proves the first display (3.17) in the theorem.

To prove the second display (3.18) in the theorem, observe that on E, $\|\widehat{\theta} - \theta_0\|_2^2 \ge \|\theta_0(s) - \theta_0\|_2^2$ by construction of $\theta_0(s)$; thus from (3.57), we have

$$\|\widehat{\theta} - \theta_0(s)\|_2^2 \le 2\epsilon^\top(\theta_0(s) - \widehat{\theta}) + 2\lambda\big(\|D\theta_0(s)\|_1 - \|D\widehat{\theta}\|_1\big).$$

and the right-hand side here can be again handled as in the proof of Theorem 1. □

3.A.4 Proofs of Theorem 10 and (3.21)

Proof of Theorem 10. For each $i = 1, \ldots, n$, consider the univariate negative log-likelihood function g defined by

$$g_i(\theta_i) = -y_i\theta_i + \Lambda(\theta_i).$$

This is a strictly convex, twice continuously differentiable function, due to our assumptions on the cumulant generating function Λ. Therefore, the second derivative of g satisfies

$$g_i''(\theta_i) = \Lambda''(\theta_i) \ge m,$$

i.e., its has a (strictly positive) minimum on the compact interval $[l, u]$, which we denote as $m > 0$. Now define $f(\theta) = \sum_{i=1}^n g(\theta_i)$ as the negative log-likelihood loss over all n samples. The above display implies that

$$f(\theta) - f(\theta_0) - \nabla f(\theta_0)^\top(\theta - \theta_0) \ge \frac{m}{2}\|\theta - \theta_0\|_2^2, \quad \text{for } \theta_i, \theta_{0,i} \in [\ell, u],\ i = 1, \ldots, n. \quad (3.58)$$

Returning to our estimate $\widehat{\theta}$ in (3.20), by comparing the objectives at $\widehat{\theta}$ and at θ_0, we have

$$f(\widehat{\theta}) + \lambda\|D\widehat{\theta}\|_1 \le f(\theta_0) + \lambda\|D\theta_0\|_1.$$

Rearranging the terms in the above display and using (3.58), we have

$$\frac{m}{2}\|\widehat{\theta} - \theta_0\|_2^2 \le -\nabla f(\theta_0)^\top(\widehat{\theta} - \theta_0) + \lambda\big(\|D\theta_0\|_1 - \|D\widehat{\theta}\|_1\big). \quad (3.59)$$

By assumption, the components of the random vector $-\nabla f(\theta_0)$, namely

$$-\nabla_i f(\theta_0) = y_i - \Lambda'(\theta_{0,i}) = y_i - \mathbb{E}(y_i), \quad i = 1, \ldots, n,$$

follow a sub-Gaussian distribution. Thus the right-hand side in (3.59) can be analyzed exactly as in the proof of Theorem 1, which leads to the desired result. □

Proof of (3.21). From (3.59), observe

$$\frac{m}{2M}\|\widehat{\theta} - \theta_0\|_2^2 \leq \frac{-\nabla f(\theta_0)}{M}^{\top}(\widehat{\theta} - \theta_0) + \frac{\lambda}{M}\big(\|D\theta_0\|_1 - \|D\widehat{\theta}\|_1\big), \tag{3.60}$$

where $M > 1$ is a parameter free to vary, that we will specify below. Define an event

$$E = \{y : \|\nabla f(\theta_0)\|_\infty \leq M\} = \bigcap_{i=1}^{n}\{y_i : |y_i - \mu(\theta_{0,i})| \leq M\},$$

On E, the random vector $-\nabla f(\theta_0)/M$ has sub-Gaussian components (since it is bounded), and the right-hand side in (3.60) can be analyzed as in the proof of Theorem 1. The final error bound will be the usual error bound (i.e., that from Theorem 1) multiplied by a factor of M.

Now we bound the probability of E. For $W \sim \text{Pois}(\mu)$, by Poisson concentration results (Pollard, 2015),

$$\mathbb{P}(|W - \mu| > x) \leq 2\exp\left(-\frac{x^2}{2\mu}\psi\left(\frac{x}{\mu}\right)\right), \quad \text{for } x > 0, \text{ where } \psi(x) = \frac{(1+x)\log(1+x) - x}{x^2/2}.$$

□

Observe for any $x \geq 1$,

$$\frac{x^2}{2\mu}\psi\left(\frac{x}{\mu}\right) \geq \frac{x^2}{2\mu}\frac{1}{1 + x/(3\mu)} \geq \frac{1/2}{\mu + 1/3}x.$$

Setting $M = \delta \log n$ for a constant $\delta > 0$ to be determined, and using the last two displays, as well as the bound $\mu(\theta_{0,i}) = e^{\theta_{0,i}} \leq e^u$, $i = 1, \ldots, n$, yields

$$\mathbb{P}(E^c) \leq n\exp\left(-\frac{1/2}{e^u + 1/3}\delta\log n\right) = \exp\left(\left(1 - \frac{1/2}{e^u + 1/3}\delta\right)\log n\right).$$

Now we simply need to choose δ large enough so that the right-hand side above equals $1/n$, i.e., we choose $\delta = 4(e^u + 1/3)$, and this completes the proof.

3.A.5 Proof of Theorem 14

Fix any $\epsilon > 0$. By assumption, we know that there is a constant $C > 0$ and an integer $N_1 > 0$ such that

$$\mathbb{P}\bigg(\|\widetilde{\theta} - \theta_0\|_n^2 > \frac{C}{4} R_n \bigg) \leq \epsilon,$$

for all $n \geq N_1$. We also know that there is an integer $N_2 > 0$ such that $2CnR_n/H_n^2 \leq W_n$ for all $n \geq N_2$. Let $N = \max\{N_1, N_2\}$, take $n \geq N$, and let $r_n = \lfloor CnR_n/H_n^2 \rfloor$.

Suppose that $d(S(\widetilde{\theta}) \,|\, S_0) > r_n$. Then, by definition, there exists a changepoint $t_i \in S_0$ such that no changepoints of $\widetilde{\theta}$ are within r_n of t_i, which means that $\widetilde{\theta}_j$ is constant over $j \in \{t_i - r_n + 1, \ldots, t_i + r_n\}$. Denote

$$z = \widetilde{\theta}_{t_i - r_n + 1} = \ldots = \widetilde{\theta}_{t_i} = \widetilde{\theta}_{t_i + 1} = \ldots = \widetilde{\theta}_{t_i + r_n}.$$

We then form the lower bound

$$\frac{1}{n} \sum_{j = t_i - r_n + 1}^{t_i + r_n} \big(\widetilde{\theta}_j - \theta_{0,j}\big)^2 = \frac{r_n}{n} \big(z - \theta_{0,t_i}\big)^2 + \frac{r_n}{n} \big(z - \theta_{0,t_i+1}\big)^2 \geq \frac{r_n H_n^2}{2n} > \frac{C}{4} R_n,$$

where the first inequality holds because $(x-a)^2 + (x-b)^2 \geq (a-b)^2/2$ for all x (the quadratic in x here is minimized at $x = (a+b)/2$), and the second because $r_n = \lfloor CnR_n/H_n^2 \rfloor$. Therefore, we see that $d(S(\widetilde{\theta}) \,|\, S_0) > r_n$ implies

$$\|\widetilde{\theta} - \theta_0\|_n^2 \geq \frac{1}{n} \sum_{j = t_i - r_n + 1}^{t_i + r_n} \big(\widetilde{\theta}_j - \theta_{0,j}\big)^2 > \frac{C}{4} R_n,$$

which implies

$$\mathbb{P}\Big(d\big(S(\widetilde{\theta}) \,|\, S_0\big) > r_n \Big) \leq \mathbb{P}\bigg(\|\widetilde{\theta} - \theta_0\|_n^2 > \frac{C}{4} R_n \bigg) \leq \epsilon,$$

for all $n \geq N$, completing the proof. □

3.B Approximate changepoint recovery result, using post-processing

Here we state and prove a general result on approximate changepoint recovery using post-processing. It is a precursor to the result in Theorem 19 and will be used to prove the latter.

Theorem 24. *Let $\widetilde{\theta} \in \mathbb{R}^n$ be such that $\|\widetilde{\theta} - \theta_0\|_n^2 = O_{\mathbb{P}}(R_n)$. Consider the following procedure: we evaluate the filter in* (3.24) *with bandwidth b_n at all locations $i = b_n, \ldots, n - b_n$,*

and only keep the locations whose absolute filter value is greater than or equal to a threshold τ_n*. Denote the resulting filtered set by*

$$S_A(\widetilde{\theta}) = \big\{i \in \{b_n, \dots, n - b_n\} : |F_i(\widetilde{\theta})| \geq \tau_n\big\}.$$

For bandwidth and threshold values satisfying $b_n = \omega(nR_n/H_n^2)$*,* $2b_n \leq W_n$*, and* $\tau_n/H_n \to \rho \in (0,1)$ *as* $n \to \infty$*, we have*

$$\mathbb{P}\Big(d_H\big(S_A(\widetilde{\theta}), S_0\big) \leq b_n\Big) \to 1 \quad \text{as } n \to \infty.$$

Proof. The proof is not complicated conceptually, but requires some careful bookkeeping. Also, we make use of a few key lemmas whose details will be given later. Fix $\epsilon > 0$. Let $C > 0$ and $N_1 > 0$ be an integer such that for all $n \geq N_1$,

$$\mathbb{P}\Big(\|\widetilde{\theta} - \theta_0\|_n^2 > CR_n\Big) \leq \frac{\epsilon}{2}.$$

Set $\epsilon = \min\{\rho, 1-\rho\}/2$. As $b_n = \omega(nR_n/H_n^2)$, there is an integer $N_2 > 0$ such that for all $n \geq N_2$,

$$\frac{2CnR_n}{b_n} \leq (0.99\epsilon H_n)^2.$$

As $\tau_n/H_n \to \rho \in (0,1)$, there is an integer $N_3 > 0$ such that for all $n \geq N_3$,

$$(\rho - \epsilon)H_n \leq \tau_n \leq (\rho + \epsilon)H_n.$$

Set $N = \max\{N_1, N_2, N_3\}$, and take $n \geq N$. Note that $\epsilon \leq \rho - \epsilon$ and $\rho + \epsilon \leq 1 - \epsilon$ by construction, and thus by the last two displays,

$$\sqrt{\frac{2CnR_n}{b_n}} < \tau_n < H_n - \sqrt{\frac{2CnR_n}{b_n}}. \tag{3.61}$$

Now observe

$$\mathbb{P}\Big(d_H\big(S_A(\widetilde{\theta}), S_0\big) > b_n\Big) \leq \mathbb{P}\Big(d\big(S_A(\widetilde{\theta}) \,|\, S_0\big) > b_n\Big) + \mathbb{P}\Big(d\big(S_0 \,|\, S_A(\widetilde{\theta})\big) > b_n\Big). \tag{3.62}$$

We focus on bounding each term on the right-hand side above separately. For the first term on the right-hand side in (3.62), observe that if $F_{t_i}(\widetilde{\theta}) \geq \tau_n$ for all $t_i \in S_0$, then $d(S_A(\widetilde{\theta}) \,|\, S_0) \leq b_n$. By the contrapositive,

$$\begin{aligned}\mathbb{P}\Big(d\big(S_A(\widetilde{\theta}) \,|\, S_0\big) > b_n\Big) &\leq \mathbb{P}\Big(|F_{t_i}(\widetilde{\theta})| < \tau_n \text{ for some } t_i \in S_0\Big) \\ &\leq \mathbb{P}\Big(|F_{t_i}(\widetilde{\theta})| < H_n - \sqrt{\frac{2CnR_n}{b_n}} \text{ for some } t_i \in S_0\Big),\end{aligned} \tag{3.63}$$

where in the second line we used the upper bound on τ_n in (3.61). Suppose that $\|\widetilde{\theta} - \theta_0\|_n^2 \le CR_n$; then, for $t_i \in S_0$, Lemma 26 tells us how small $|F_{t_i}(\widetilde{\theta})|$ can be made with this error bound in place. Specifically, define

$$a = (\underbrace{-1/b_n, \ldots, -1/b_n}_{b_n \text{ times}}, \underbrace{1/b_n, \ldots, 1/b_n}_{b_n \text{ times}}) \quad \text{and} \quad c = (\theta_{0,t_i-b_n+1}, \ldots, \theta_{0,t_i+b_n}),$$

and also $r = \sqrt{CnR_n}$. Then Lemma 26 implies the following: if $\|\widetilde{\theta} - \theta_0\|_n^2 \le CR_n$, then

$$|F_{t_i}(\widetilde{\theta})| \ge |a^\top c| - r\|a\|_2 \ge |\theta_{0,t_i+1} - \theta_{0,t_i}| - \sqrt{\frac{2CnR_n}{b_n}} \ge H_n - \sqrt{\frac{2CnR_n}{b_n}}.$$

Therefore, continuing on from (3.63),

$$\begin{aligned}
\mathbb{P}\Big(d\big(S_A(\widetilde{\theta}) \,|\, S_0\big) > b_n\Big) &\le \mathbb{P}\Big(|F_{t_i}(\widetilde{\theta})| < H_n - \sqrt{\frac{2CnR_n}{b_n}} \text{ for some } t_i \in S_0\Big) \\
&\le \mathbb{P}\Big(\|\widetilde{\theta} - \theta_0\|_n^2 > CR_n\Big) \\
&\le \frac{\epsilon}{2}.
\end{aligned}$$

It suffices to consider the second term in (3.62), and show that this is also bounded by $\epsilon/2$. Note that

$$\begin{aligned}
\mathbb{P}\Big(d\big(S_0 \,|\, S_A(\widetilde{\theta})\big) > b_n\Big) &\le \mathbb{P}\Big(|F_i(\widetilde{\theta})| \ge \tau_n \text{ at some } i \text{ such that } \theta_{0,i-b_n+1} = \ldots = \theta_{0,i+b_n}\Big) \\
&\le \mathbb{P}\Big(|F_i(\widetilde{\theta})| > \sqrt{\frac{2CnR_n}{b_n}} \text{ at some } i \text{ such that } \theta_{0,i-b_n+1} = \ldots = \theta_{0,i+b_n}\Big).
\end{aligned} \tag{3.64}$$

In the second inequality we used the lower bound on τ_n in (3.61). Similar to the previous argument, suppose that $\|\widetilde{\theta} - \theta_0\|_n^2 \le CR_n$; for any location i in consideration in (3.64), Lemma 25 tells us how large $|F_i(\theta)|$ can be made with this error bound in place. Defining

$$a = (\underbrace{-1/b_n, \ldots, -1/b_n}_{b_n \text{ times}}, \underbrace{1/b_n, \ldots, 1/b_n}_{b_n \text{ times}}) \quad \text{and} \quad c = (\theta_{0,i-b_n+1}, \ldots, \theta_{0,i+b_n}),$$

and $r = \sqrt{CnR_n}$, as before, the lemma says the following: if $\|\widetilde{\theta} - \theta_0\|_n^2 \le CR_n$, then

$$|F_i(\widetilde{\theta})| \le |a^\top c| + r\|a\|_2 = \sqrt{\frac{2CnR_n}{b_n}}.$$

Hence, continuing on from (3.64),

$$\begin{aligned}\mathbb{P}\Big(d\big(S_0 \,|\, S_A(\widetilde{\theta})\big) > b_n\Big) &\leq \mathbb{P}\Bigg(|F_i(\widetilde{\theta})| > \sqrt{\frac{2CnR_n}{b_n}} \text{ at some } i \text{ such that } \theta_{0,i-b_n+1} = \ldots = \theta_{0,i+b_n}\Bigg) \\ &\leq \mathbb{P}\Big(\|\widetilde{\theta} - \theta_0\|_n^2 > CR_n\Big) \\ &\leq \frac{\epsilon}{2},\end{aligned}$$

completing the proof. □

3.B.1 Lemmas 25 and 26

The proof of Theorem 24 above relied on two lemmas, that we state below. Their proofs are based on simple arguments in convex analysis.

Lemma 25. *Given $a, c \in \mathbb{R}^m$, $r \geq 0$, the optimal value of the (nonconvex) optimization problem*

$$\max_{x \in \mathbb{R}^m} \; |a^\top x| \;\; \textit{such that} \;\; \|x - c\|_2 \leq r \tag{3.65}$$

is $|a^\top c| + r\|a\|_2$.

Proof. We first consider the convex optimization problem

$$\min_{x \in \mathbb{R}^m} \; a^\top x \;\text{ such that }\; \|x - c\|_2 \leq r, \tag{3.66}$$

whose Lagrangian may be written as, for a dual variable $\lambda \geq 0$,

$$L(x, \lambda) = a^\top x + \lambda(\|x - c\|_2^2 - r^2).$$

The stationarity condition is $a + \lambda(x - c) = 0$, thus $x = c - a/\lambda$. By primal feasibility, $\|x - c\|_2 \leq r$, we see that we can take $\lambda = \|a\|_2/r$, which gives a solution $x = c - ra/\|a\|_2$. The optimal value in (3.66) is therefore $a^\top x = a^\top c - r\|a\|_2$. By the same logic, the optimal value of the convex problem

$$\max_{x \in \mathbb{R}^m} \; a^\top x \;\text{ such that }\; \|x - c\|_2 \leq r \tag{3.67}$$

is $a^\top c + r\|a\|_2$. Now we can read off the optimal value of (3.65) from those of (3.66), (3.67): its optimal value is

$$\max\big\{ -\big(a^\top c - r\|a\|_2\big),\; a^\top c + r\|a\|_2 \big\} = |a^\top c| + r\|a\|_2,$$

completing the proof. □

Lemma 26. *Given $a, c \in \mathbb{R}^m$, $r \geq 0$ such that $|a^\top c| - r\|a\|_2 \geq 0$, the optimal value of the (convex) optimization problem*

$$\min_{x \in \mathbb{R}^n} \ |a^\top x| \ \text{ such that } \ \|x - c\|_2 \leq r \tag{3.68}$$

is $|a^\top c| - r\|a\|_2$.

Proof. The proof is nearly immediate from the proof of Lemma 25, above. Notice that the optimal value of (3.68) is lower bounded by that of (3.66), which we already know is $a^\top c - r\|a\|_2^2$. But when the latter is nonnegative, this is also the optimal value of (3.68). Repeating the argument with $-a$ in place of a gives the result as stated in the lemma. □

3.B.2 Proof of Theorem 19

We will show that

$$\Big\{ d_H\big(S_A(\widetilde{\theta}), S_0\big) \leq b_n \Big\} \subseteq \Big\{ d_H\big(S_F(\widetilde{\theta}), S_0\big) \leq 2b_n \Big\}, \tag{3.69}$$

Since the left-hand side occurs with probability tending to 1, by Theorem 24, so will the right-hand side. To show the desired containment, recall that, by the definition of Hausdorff distance,

$$\Big\{ d_H\big(S_A(\widetilde{\theta}), S_0\big) \leq b_n \Big\} = \Big\{ d\big(S_0 \,|\, S_A(\widetilde{\theta})\big) \leq b_n \Big\} \cap \Big\{ d\big(S_A(\widetilde{\theta}) \,|\, S_0\big) \leq b_n \Big\}. \tag{3.70}$$

Inspecting the first term on the right-hand side of (3.70), we observe

$$\Big\{ d\big(S_0 \,|\, S_A(\widetilde{\theta})\big) \leq b_n \Big\} \subseteq \Big\{ d\big(S_0 \,|\, S_A(\widetilde{\theta})\big) \leq 2b_n \Big\} \subseteq \Big\{ d\big(S_0 \,|\, S_F(\widetilde{\theta})\big) \leq 2b_n \Big\}, \tag{3.71}$$

where the last containment holds as $S_F(\widetilde{\theta}) \subseteq S_A(\widetilde{\theta})$. Inspecting the second term on the right-hand side of (3.70), we apply Lemma 27 which states that for each $j \in \{b_n, \ldots, n - b_n\}$, there exists $i \in I_F(\widetilde{\theta})$ such that $|i - j| \leq b_n$ and $|F_i(\widetilde{\theta})| \geq |F_j(\widetilde{\theta})|$. Using this, we see

$$\begin{aligned}
\Big\{ d\big(S_A(\widetilde{\theta}) \,|\, S_0\big) \leq b_n \Big\} &= \Big\{ \text{for all } \ell \in S_0, \text{ there exists } j \in S_A(\widetilde{\theta}) \text{ such that } |\ell - j| \leq b_n \Big\} \\
&\subseteq \Big\{ \text{for all } \ell \in S_0, \text{ there exists } i \in I_F(\widetilde{\theta}) \text{ such that } |\ell - i| \leq 2b_n \Big\} \\
&= \Big\{ d\big(S_F(\widetilde{\theta}) \,|\, S_0\big) \leq 2b_n \Big\}.
\end{aligned} \tag{3.72}$$

Above, we have used Lemma 27 for the containment in the second line. Combining (3.70), (3.71), and (3.72), we have established (3.69), as desired. □

3.B.3 Lemma 27

The proof of Theorem 19 relied on the following lemma.

Lemma 27. *Let $I_F(\widetilde{\theta})$ be the candidate set defined in Theorem 19. For each $j \in \{b_n, \ldots, n - b_n\}$ where $|F_j(\widetilde{\theta})| > 0$, there exists $i \in I_F(\widetilde{\theta})$ such that $|i - j| \leq b_n$ and $|F_i(\widetilde{\theta})| \geq |F_j(\widetilde{\theta})|$.*

Proof. To facilitate the proof, we define the concept of a *local maximum* among the absolute filter values: a location i is a local maximum if its absolute filter value $|F_i(\widetilde{\theta})|$ is be greater than or equal to the absolute values at neighboring locations, and strictly greater than at least one of these values (where the boundary points are treated as having just one neighboring location). Specifically, a local maximum i must satisfy one of the following conditions

$$|F_{i-1}(\widetilde{\theta})| < |F_i(\widetilde{\theta})|,\ |F_{i+1}(\widetilde{\theta})| \leq |F_i(\widetilde{\theta})|, \qquad \text{if } i \in \{b_n + 1, \ldots, n - b_n - 1\}, \tag{3.73}$$

$$|F_{i-1}(\widetilde{\theta})| \leq |F_i(\widetilde{\theta})|,\ |F_{i+1}(\widetilde{\theta})| < |F_i(\widetilde{\theta})|, \qquad \text{if } i \in \{b_n + 1, \ldots, n - b_n - 1\}, \tag{3.74}$$

$$|F_{i+1}(\widetilde{\theta})| < |F_i(\widetilde{\theta})| \qquad \text{if } i = b_n, \tag{3.75}$$

$$|F_{i-1}(\widetilde{\theta})| < |F_i(\widetilde{\theta})| \qquad \text{if } i = n - b_n. \tag{3.76}$$

Let $L(\widetilde{\theta})$ denote the set of local maximums derived from the filter with bandwidth b_n, i.e., the set of locations i satisfying one of the four conditions (3.73)–(3.76).

We first establish that $L(\widetilde{\theta}) \subseteq I_F(\widetilde{\theta})$. Fix $i \in L(\widetilde{\theta})$. The boundary cases, $i = b_n$ or $i = n - b_n$, are handled directly by the definition of $I_F(\widetilde{\theta})$. Hence, we may assume that $i \in \{b_n + 1, \ldots, n - b_n - 1\}$, and without a loss of generality,

$$|F_i(\widetilde{\theta})| > |F_{i-1}(\widetilde{\theta})| \quad \text{and} \quad |F_i(\widetilde{\theta})| \geq |F_{i+1}(\widetilde{\theta})|,$$

as well as $F_i(\widetilde{\theta}) > 0$. This means that

$$F_i(\widetilde{\theta}) > |F_{i-1}(\widetilde{\theta})| \quad \text{and} \quad F_i(\widetilde{\theta}) \geq |F_{i+1}(\widetilde{\theta})|,$$

which of course implies

$$F_i(\widetilde{\theta}) > F_{i-1}(\widetilde{\theta}) \quad \text{and} \quad F_i(\widetilde{\theta}) \geq F_{i+1}(\widetilde{\theta}).$$

Applying the definition of the filter in (3.24) gives

$$\left(\sum_{j=i+1}^{i+b_n} \widetilde{\theta}_j - \sum_{j=i-b_n+1}^{i} \widetilde{\theta}_j \right) - \left(\sum_{j=i}^{i+b_n-1} \widetilde{\theta}_j - \sum_{j=i-b_n}^{i-1} \widetilde{\theta}_j \right) > 0$$

$$\left(\sum_{j=i+1}^{i+b_n} \widetilde{\theta}_j - \sum_{j=i-b_n+1}^{i} \widetilde{\theta}_j \right) - \left(\sum_{j=i+2}^{i+b_n+1} \widetilde{\theta}_j - \sum_{j=i-b_n+2}^{i+1} \widetilde{\theta}_j \right) \geq 0,$$

or, after simplification,

$$\widetilde{\theta}_{i+b_n} - 2\widetilde{\theta}_i + \widetilde{\theta}_{i-b_n} > 0 \quad \text{and} \quad -\widetilde{\theta}_{i+b_n+1} + 2\widetilde{\theta}_{i+1} - \widetilde{\theta}_{i-b_n+1} \geq 0.$$

Adding the above two equations together, we get

$$-\big(\widetilde{\theta}_{i+b_n+1} - \widetilde{\theta}_{i+b_n}\big) + 2\big(\widetilde{\theta}_{i+1} - \widetilde{\theta}_i\big) - \big(\widetilde{\theta}_{i-b_n+1} - \widetilde{\theta}_{i-b_n}\big) > 0,$$

which implies at least one of the three bracketed pairs of terms must be nonzero, i.e., a changepoint must occur at one of the locations i, $i + b_n$, or $i - b_n$. The proves that $L(\widetilde{\theta}) \subseteq I_F(\widetilde{\theta})$.

Now we show the intended statement. Let $j \in \{b_n, \ldots, n - b_n\}$, and $i \in L(\widetilde{\theta})$ be in the direction of ascent from j with respect to $F(\widetilde{\theta})$, where $j \leq i$, without a loss of generality (for the case $i < j$, replace $\ell + b_n$ below by $\ell - b_n$). That is, the location i is a local maximum where

$$|F_j(\widetilde{\theta})| \leq |F_{j+1}(\widetilde{\theta})| \leq \ldots \leq |F_{i-1}(\widetilde{\theta})| \leq |F_i(\widetilde{\theta})|. \tag{3.77}$$

If $|i - j| \leq b_n$, then we have the desired result, due to (3.77). If $|i - j| > b_n$, then there must be at least one location $\ell \in S(\widetilde{\theta})$ such that $|\ell - j| \leq b_n$. (To see this, note that if $\widetilde{\theta}_{j-b_n+1} = \ldots = \widetilde{\theta}_{j+b_n}$, then $F_j(\widetilde{\theta}) = 0$.) Thus, at least one of $\ell, \ell + b_n$ lies in between j and i, and then again (3.77) implies the result, completing the proof. □

3.C Comparison of Corollaries 20 and 21 to other results in the literature

Below are some remarks on the results in Corollaries 20 and 21.

Remark 28 (**Recovery under weak sparsity, comparison to BS**). *The weak sparsity result in* (3.25) *of Corollary 20 considers a challenging setting in which the number of changepoints s_0 in θ_0 could be growing quickly with n, and we only have control on $C_n = \|D\theta_0\|_1$. We draw a comparison here to known results on binary segmentation (BS). The result in* (3.25) *on the (filtered) fused lasso and Theorem 3.1 in Fryzlewicz (2014) on the BS estimator $\widehat{\theta}^{\mathrm{BS}}$, each under the appropriate conditions on W_n, H_n, state that*

$$d_H\big(S_F(\widehat{\theta}), S_0\big) \leq \frac{2n^{1/3}C_n^{2/3}\log n}{H_n^2} \quad \textit{vs.} \quad d_H\big(S(\widehat{\theta}^{\mathrm{BS}}), S_0\big) \leq \frac{cn\log n}{H_n^2} \quad \textit{respectively,} \tag{3.78}$$

where $c > 0$ is a constant, and both bounds hold with probability approaching 1. The result on $S_F(\widehat{\theta})$ is obtained by choosing $\nu_n = \sqrt{\log n}$ and then $b_n = \lfloor n^{1/3}C_n^{2/3}\log n/H_n^2 \rfloor$ in (3.25)*. Examining* (3.78)*, we see that, when C_n scales more slowly than n, Corollary 20 provides the stronger result: the term $n^{1/3}C_n^{2/3}$ will be smaller than n, and hence the bound on $d_H(S_F(\widehat{\theta}), S_0)$ will be sharper than that on $d_H(S(\widehat{\theta}^{\mathrm{BS}}), S_0)$.*

But we must also examine the specific restrictions that each result in (3.78) *places on* s_0, W_n, H_n. *Consider the simplification* $W_n = \Theta(n/s_0)$, *corresponding to a case in which the changepoints in* θ_0 *are spaced evenly apart. Corollary 20, starting with the condition* $n^{1/3}C_n^{2/3}\log n/H_n^2 \le W_n/2$, *plugging in the relationship* $C_n \ge s_0 H_n$, *and rearranging to derive a lower bound on the minimum signal gap, requires* $H_n = \Omega(s_0^{5/4} n^{-1/2}\log^{3/4} n)$. *If* $s_0 = \Theta(n^{2/5})$, *then we see that the minimum signal gap requirement becomes* $H_n = \Omega(\log^{3/4} n)$, *which is growing with* n *and is thus too stringent to be interesting (Sharpnack et al. (2012) showed simple thresholding of pairwise differences achieves perfect recovery when* $H_n = \omega(\sqrt{\log n})$*). Hence, to accommodate signals for which* H_n *remains constant or even shrinks with* n, *we must restrict the number of jumps in* θ_0 *according to* $s_0 = O(n^{2/5-\delta})$, *for any fixed* $\delta > 0$. *Meanwhile, inspection of Assumption 3.2 in Fryzlewicz (2014) reveals that his Theorem 3.1 requires* $s_0 = O(n^{1/4-\delta})$, *for any* $\delta > 0$, *in order to handle signals such that* H_n *remains constant or shrinks with* n. *In short, the (effectively) allowable range for* s_0 *is larger for Corollary 20 than for Theorem 3.1 in Fryzlewicz (2014). Even when we look within their common range, Corollary 20 places weaker conditions on* H_n. *As an example, consider* $s_0 = \Theta(n^{1/6})$ *and* $W_n = \Theta(n^{5/6})$. *The fused lasso result in* (3.78) *requires* $H_n = \Omega(n^{-7/24}\log^{4/3} n)$, *and the BS result in* (3.78) *requires* $H_n = \Omega(n^{-1/6+\delta})$, *for any* $\delta > 0$. *Finally, to reiterate, the fused lasso result in* (3.78) *gives a better Hausdorff recovery bound when* C_n *is small compared to* n; *at the extreme end, this is better by a full factor of* $n^{2/3}$, *when* $C_n = O(1)$.

While the post-processed fused lasso looks favorable compared to BS, based on its approximate changepoint recovery properties in the weak sparsity setting, we must be clear that the analyses for other methods—wild binary segmentation (WBS), the simultaneous multiscale changepoint estimator (SMUCE), and tail-greedy unbiased Haar (TGUH) wavelets—are still much stronger in this setting. Such methods have Hausdorff recovery bounds that are only possible for the post-processed fused lasso (at least, using our current analysis technique) when we assume strong sparsity. We discuss this next.

Remark 29 (**Recovery under strong sparsity, comparison to other methods**). *When* $s_0 = O(1)$ *and* $W_n = \Theta(n)$, *the result in* (3.26) *in Corollary 20 shows that the post-processed fused lasso estimator delivers a Hausdorff bound of*

$$d_H\big(S_F(\widehat{\theta}), S_0\big) \le \frac{2\log^2 n}{H_n^2}, \tag{3.79}$$

on the set $S_F(\widehat{\theta})$ *of filtered changepoints, with probability approaching 1. This is obtained by choosing (say)* $\nu_n = \sqrt{\log n/\log\log n}$ *and* $b_n = \lfloor \log^2 n/H_n^2 \rfloor \le W_n/2$ *in the corollary. The effective restriction on the minimum signal gap is thus* $H_n = \Omega(\log n/\sqrt{n})$, *which is quite reasonable, as* $H_n = \omega(1/\sqrt{n})$ *is needed for any method to detect a changepoint with probability tending to 1. Several other methods—the Potts estimator (Boysen et al., 2009), binary segmentation (BS) and wild binary segmentation (WBS) (Fryzlewicz, 2014), the*

simultaneous multiscale changepoint estimator (SMUCE) (Frick et al., 2014), and tail-greedy unbiased Haar wavelets (TGUH) (Fryzlewicz et al., 2018)—all admit Hausdorff recovery bounds that essentially match (3.79)*, under similarly weak restrictions on H_n. But, it should be noted that the latter three methods—WBS, SMUCE, and TGUH—continue to enjoy these same sharp Hausdorff bounds* outside of *the strong sparsity setting, i.e., their analyses do not require that $s_0 = O(1)$ and $W_n = \Theta(n)$, and instead just place weak restrictions on the allowed combinations of W_n, H_n (e.g., the analysis of WBS in Fryzlewicz (2014) only requires $W_n H_n^2 \geq \log n$). These analyses (and those for all previously described estimators) are more refined than that given in Corollary 20: they are based on specific properties of the estimator in question. The corollary, on the other hand, follows from Theorem 24, which uses a completely generic analysis that only assumes knowledge of the estimation error rate.*

Remark 30 (**Recovery in the Poisson model**). *Corollary 21 gives an approximate screening result for the post-processed fused lasso in the Poisson model, similar to the result in the strong sparsity, sub-Gaussian error case discussed above. As with all of our other approximate recovery results, this is established via the estimation error guarantees for the Poisson fused lasso estimator. Analyzing changepoint detection properties directly in the Poisson model seems like it could be a challenging task, and we are not aware of many results in the literature that do so. (Likewise for the binomial model; we did not state formal recovery results for this model but they follow from the estimation error bounds exactly as in the Poisson case, and changepoint detection analysis in this model seems difficult and we are not aware of extensive literature in this setting.)*

3.D Choosing a threshold level in the post-processing procedure

We describe a data-driven procedure to determine the threshold level τ_n of the filter in (3.24), used to derive a post-processed set of changepoints $S_F(\widetilde{\theta})$ from an estimate $\widetilde{\theta}$, as described in Theorem 19.

Let $\mathcal{A}(\cdot)$ denote a fitting algorithm that, applied to data y, outputs an estimate $\widetilde{\theta}$ of θ_0 (e.g., $\mathcal{A}(y)$ could be the minimizer in (3.3), so that its output is the fused lasso estimate). In Algorithm 1 below, we present a heuristic but intuitive method for choosing the threshold level τ_n, based on (entrywise) permutations of the residual vector $y - \widetilde{\theta}$. Aside from the choice of fitting algorithm $\mathcal{A}(\cdot)$, we must specify a number of permutations B to be explored, the bandwidth b_n for the filter in (3.24), and a quantile level $q \in (0, 1)$. The intuition behind Algorithm 1 is to set τ_n large enough to suppress "false positive" changepoints $100 \cdot q\%$ of the time (according to the permutations). This is revisited later, in the discussion of the simulation results.

Some example settings: we may take $\mathcal{A}(\cdot)$ to be the fused lasso estimator, where the tuning parameter λ is selected to minimize 5-fold cross-validation (CV) error, $B = 100$, and

Algorithm 1: Permutation-based approach for choosing τ_n

Data: Input a fitting algorithm $\mathcal{A}(\cdot)$, number of permutations B, bandwidth b_n, and quantile level $q \in (0,1)$.

Compute $\widetilde{\theta} = \mathcal{A}(y)$. Let $\widetilde{S} = S(\widetilde{\theta})$ denote the changepoints, and $r = y - \widetilde{\theta}$ the residuals.

for $b = 1, \ldots, B$ **do**

  Let $r^{(b)}$ be a random permutation of r, and define auxiliary data $y^{(b)} = \widetilde{\theta} + r^{(b)}$.

  Rerun the fitting algorithm on the auxiliary data to yield $\widetilde{\theta}^{(b)} = \mathcal{A}(y^{(b)})$.

  Apply the filter in (3.24) to $\widetilde{\theta}^{(b)}$ (with the specified bandwidth b_n), and record the largest magnitude $\widehat{\tau}^{(b)}$ of the filter values at locations greater than b_n away from $\widetilde{S}$. Formally,

$$\widehat{\tau}^{(b)} = \max_{\substack{i \in \{b_n, \ldots, n-b_n\}: \\ d(\widetilde{S} | \{i\}) > b_n}} \big| F_i(\widetilde{\theta}^{(b)}) \big|.$$

end

Output $\widehat{\tau}_n$, the level q quantile of the collection $\widehat{\tau}^{(b)}$, $b = 1, \ldots, B$.

$q = 0.95$. The choice of bandwidth b_n is more subtle, and unfortunately, there is no one choice that works for all problems.[1] But, the theory in the last section provides some general guidance: e.g., for problems in which we believe there are a small number of changepoints (i.e., $s_0 = O(1)$) of reasonably large magnitude (i.e., $H_n = \Omega(1)$), Theorem 19 instructs us to choose a bandwidth that grows faster than $\log n(\log\log n)$, so, choosing b_n to scale as $\log^2 n$ would suffice. We will use this scaling, and the above suggestions for $\mathcal{A}(\cdot)$, B, and q in all coming experiments.

After running Algorithm 1 to compute $\widehat{\tau}_n$, the idea is to proceed with the filter $S_F(\widetilde{\theta})$, applied at the level $\tau_n = \widehat{\tau}_n$, to the estimate $\widetilde{\theta}$ computed on the original data y at hand.

3.E Numerical simulations to verify some of our theoretical results

The code for the the results in this section can be found at `https://github.com/linnylin92/fused_lasso`. In our experiments, we use the following simulation setup. For a given n, the mean parameter $\theta_0 \in \mathbb{R}^n$ is defined to have $s_0 = 5$ equally-sized segments, with levels 0, 2, 4, 1, 4, from left to right. Data $y \in \mathbb{R}^n$ is generated around θ_0 using i.i.d. $N(0,4)$ noise. Lastly, the sample size n is varied between 100 and $10,000$, equally-spaced on a log scale. Figure 3.2 displays example data sets with $n = 774$ and $n = 10,000$.

[1] We note that in some situations, problem-specific intuition can yield a reasonable choice of bandwidth b_n. Also, it should be possible to extend Algorithm 1 to choose both τ_n and b_n, but we do not pursue this, for simplicity.

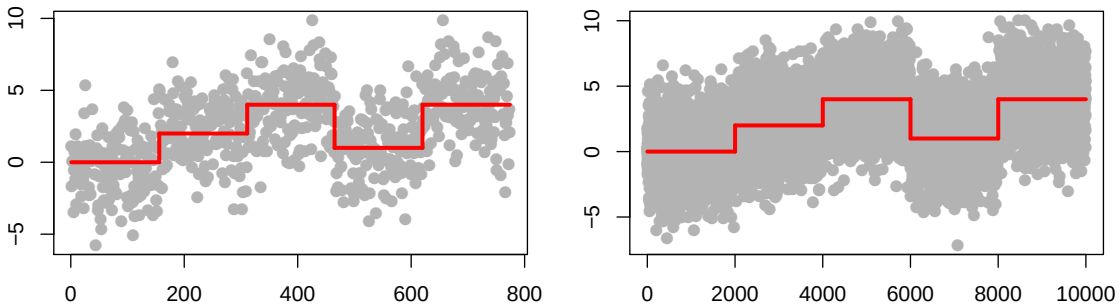


Figure 3.2: An example from our simulation setup for $n = 774$ (left) and $n = 10,000$ (right), where in each panel, the mean θ_0 is plotted in red, and the data points in gray.

For each sample size in consideration, we generated 50 example data sets from the setup described above, and on each data set, computed the full solution path of the fused lasso using the R package `genlasso`. We applied 5-fold CV to determine λ, as implemented in `genlasso`: each consecutive, non-overlapping block of 5 points were grouped into 5 different folds. When minimizing the out-of-sample test mean squared error, the average of the immediate-left and immediate-right estimates were used as a proxy for the estimate at a particular location.

Estimation error rate for fused lasso. Figure 3.3 displays the selected value of λ, as well as the estimation error $\|\widehat{\theta} - \theta_0\|_n^2$, averaged over the 50 trials, as functions of n. The results support the theoretical conclusion in Theorem 1, as the achieved estimation error rate scales as $\log n/n$ (perhaps even as $\log n(\log\log n)/n$, although it would be hard to tell the difference between the two). Also, CV appears to produce a choice of λ that scales as $\sqrt{n}$, agreeing with the scaling of λ prescribed by the theory. The screening distance $d(S(\widehat{\theta}) \,|\, S_0)$ was at most 5 across the entire simulation, regardless of n.

Evaluation of the filter. We demonstrate that the filter in (3.24), with $b_n = \lfloor 0.25 \log^2 n \rfloor$, can be effective at reducing the Hausdorff distance between estimated and true changepoint sets. We first illustrate the use of the filter in a single data example with $n = 774$, in Figure 3.4. As we can see, the fused lasso originally places a spurious jump around location 250, but this jump is eliminated when we apply the filter, provided that we set the threshold to be (say) $\tau_n = 0.5$.

Figure 3.5 now reports the results from applying the filter in problems of sizes between $n = 100$ and $n = 10,000$, using 50 trials for each n. We consider three different sets of changepoint estimates: $S(\widehat{\theta})$, the original changepoints from fused lasso estimate $\widehat{\theta}$, tuning λ via 5-fold CV; $S_F(\widehat{\theta})$, the changepoints after applying the reduced filter as described in

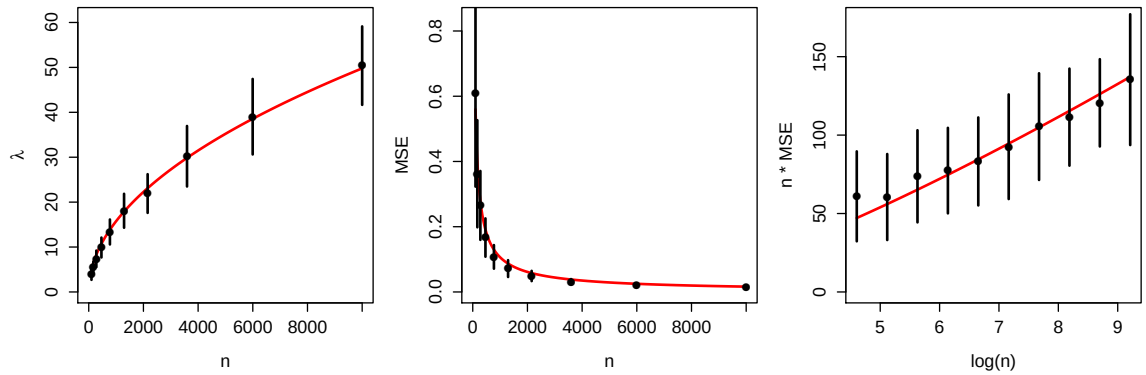


Figure 3.3: The left panel shows the median values of λ chosen to minimize 5-fold CV error, aggregated over repetitions in our simulation setup, as the sample size n varies. This scales approximately as $\sqrt{n}$, which is drawn as a red curve (with a best-fitting constant), supporting the choice prescribed by Theorem 1. The middle panel shows the corresponding estimation error $\|\widehat{\theta} - \theta_0\|_n^2$, again aggregated over repetitions, as n varies. The scaling appears to be about $\log/n$ (red curve). The right panel plots the median values of $n\|\widehat{\theta} - \theta_0\|_n^2$ against $\log n$; this looks close to linear (red line), which provides empirical support to the claim that the fused lasso error rate is $\log n/n$ (or perhaps even $\log n(\log\log n)/n$, it would be hard to distinguish these two), which is roughly in agreement with Theorem 1. In each panel, vertical bars denote ± 1 standard deviations.

Theorem 19 to $\widehat{\theta}$, with τ_n chosen by Algorithm 1; and $S_O(\widehat{\theta})$, an oracle set of changepoints given by trying a wide range of τ_n values and choosing the value that minimizes the Hausdorff distance after filtering (this assumes knowledge of S_0, and is infeasible in practice). These are labeled as "original", "data-driven", and "oracle" in the figure, respectively. As we can see from the left and middle panels, the Hausdorff distance achieved by the original changepoint set grows nearly linearly with n, but after applying the filter, the Hausdorff distance becomes very small, provided that n is larger than 1000 or so. Empirically, the Hausdorff distance associated with the filtered set appears to grow very slowly with n, nearly constant (slower than the the $\log n(\log\log n)$ rate guaranteed by Theorem 19). The right panel shows that our data-driven choices of τ_n are not substantially different from those made by the oracle.

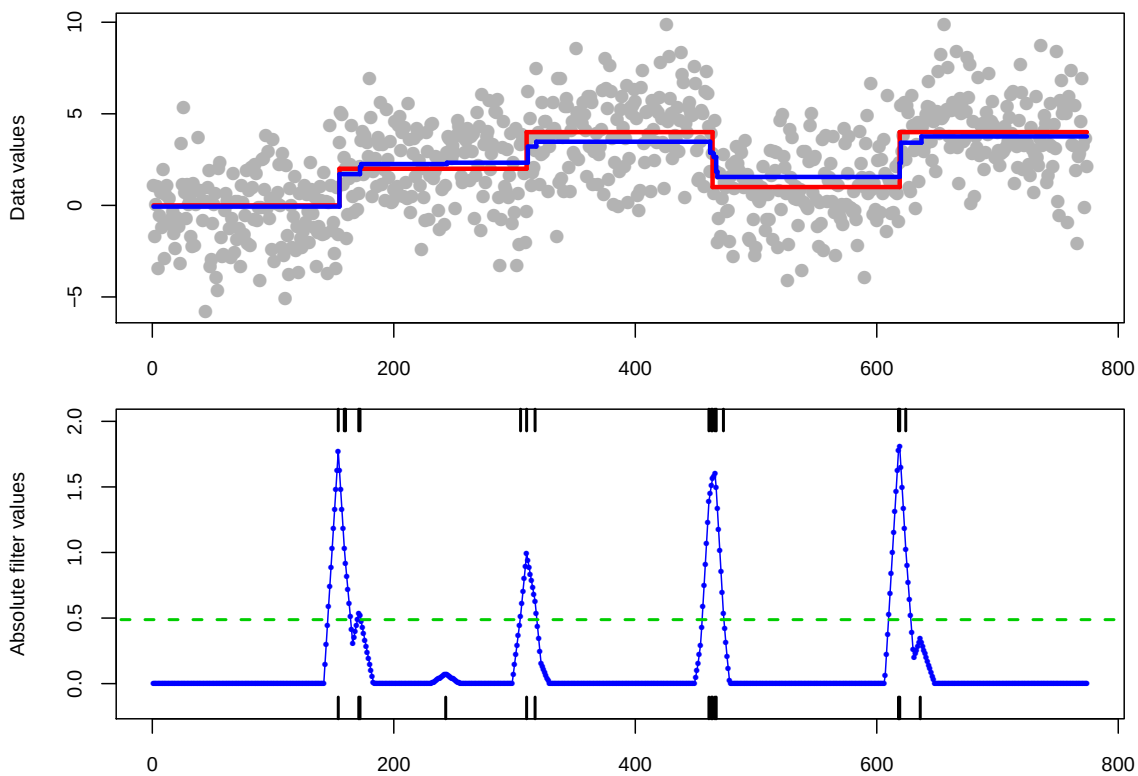


Figure 3.4: In the top plot, an example with $n = 774$ is shown from our simulation setup, where the data y is drawn in gray, the mean θ_0 in red, and the fused lasso estimate $\widehat{\theta}$ in blue. In the bottom plot, the filter values $F_i(\widehat{\theta})$, $i = 1, \ldots, n$ are drawn in blue, and the threshold τ_n is drawn as a horizontal green line. Changepoints before and after filtering are marked by short black lines along the bottom and top x-axes, respectively.

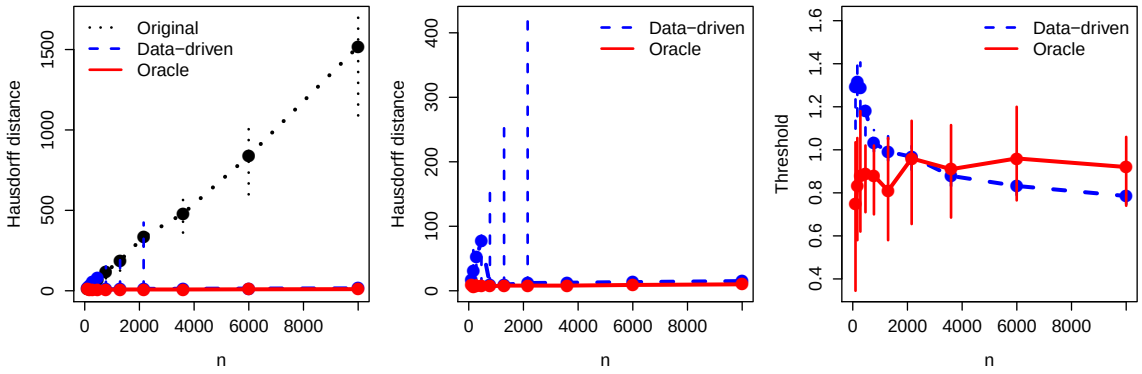


Figure 3.5: In the left panel, the Hausdorff distances between original changepoints, filtered changepoints with a data-driven threshold, and filtered changepoints with an oracle threshold, are plotted (in black, blue, and red, respectively). The results are aggregated across 50 trial runs for each sample size n; the solid dots display the median values, and the vertical segments display the interquartile ranges (25th to 75th percentiles). The middle panel zooms in on the Hausdorff distances for the data-driven and oracle filtering procedures, and the right panel displays the choices of τ_n for these procedures.

Four

Detecting heterogeneity – Post-selection inference for changepoint significance

Paper summary: Changepoint detection methods are used in many areas of science and engineering, e.g., in the analysis of copy number variation data to detect abnormalities in copy numbers along the genome. Despite the broad array of available tools, methodology for quantifying our uncertainty in the strength (or presence) of given changepoints *post-selection* are lacking. Post-selection inference offers a framework to fill this gap, but the most straightforward application of these methods results in low-powered hypothesis tests and leaves open several important questions about practical usability. In this work, we carefully tailor post-selection inference methods towards changepoint detection, focusing on copy number variation data. To accomplish this, we study commonly used changepoint algorithms: binary segmentation, as well as two of its most popular variants, wild and circular, and the fused lasso. We implement some of the latest developments in post-selection inference theory, mainly auxiliary randomization. This improves the power, which requires implementations of MCMC algorithms (importance sampling and hit-and-run sampling) to carry out our tests. We also provide recommendations for improving practical useability, detailed simulations, and an example analysis on array comparative genomic hybridization (CGH) data.

The work in this chapter was done jointly with Sangwon Hyun, Max G'Sell, and Ryan J. Tibshirani, and has been submitted to Biometrics under the title "Valid post-selection inference for segmentation methods with application to copy number variation data."

4.1 Introduction

Changepoint detection is the problem of identifying changes in data distribution along a sequence of observations. We study the canonical changepoint problem, where changes occur only in the mean: let vector $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$ be a data vector with independent

entries,

$$Y_i \sim \mathcal{N}(\theta_i, \sigma^2), \quad i = 1, \ldots, n, \tag{4.1}$$

where the unknown mean vector $\theta \in \mathbb{R}^n$ forms a piecewise constant sequence. That is, for locations $1 \leq b_1 < \cdots < b_t \leq n-1$,

$$\theta_{b_j+1} = \ldots = \theta_{b_{j+1}}, \quad j = 0, \ldots, t.$$

where for convenience we write $b_0 = 0$ and $b_{t+1} = n$. We call $b_1, \ldots, b_t$ *changepoint* locations of θ. Changepoint detection algorithms typically focus on estimating the number of changepoints t (which could possibly be 0), as well as the locations $b_1, \ldots, b_t$, from a single realization Y. Roughly speaking, changepoint methodology (and its associated literature) can be divided into two classes of algorithms: *segmentation* algorithms and *penalization* algorithms. The former class includes *binary segmentation* (BS) (Vostrikova, 1981) and popular variants like *wild binary segmentation* (WBS) (Fryzlewicz, 2014) and *circular binary segmentation* (CBS) (Olshen et al., 2004); the latter class includes the *fused lasso* (FL) (Tibshirani et al., 2005) (also called *total variation denoising* (Rudin et al., 1992) in signal processing), and the *Potts estimator* (Boysen et al., 2009). These two classes have different strengths; see, e.g., Cho and Fryzlewicz (2011); Lin et al. (2017) for more discussion.

Having estimated changepoint locations, a natural follow-up goal would be to conduct statistical inference on the significance of the changes in mean at these locations. Despite the large number of segmentation algorithms and penalization algorithms available for changepoint detection, there has been very little focus on formally valid inferential tools to use *post-selection* – after the changepoints have been selected. In this work, we describe a suite of inference tools to use after a changepoint algorithm has been applied—namely, BS, WBS, CBS, or FL. We work in the framework of *post-selection inference*, also called *selective inference*. The specific machinery that we build off was first introduced in Lee et al. (2016); Tibshirani et al. (2016), and further developed in various works, notably Fithian et al. (2014, 2015); Tian and Taylor (2018), whose extensions we rely on in particular.

Basic inference procedure. The basic inference procedure we consider is as follows.

1. Given data Y, apply a changepoint algorithm to detect some fixed number of changepoints k. Denote the estimated changepoint locations by $\widehat{b}_1, \ldots, \widehat{b}_k$, and their respective changepoint directions (whether the estimated change in mean was positive or negative) by $\widehat{d}_1, \ldots, \widehat{d}_k \in \{-1, 1\}$. Let $I_1, \ldots, I_{k+1}$ denote the partition of $\{1, \ldots, n\}$ formed by $\widehat{b}_{1:k}$. The specifics of the changepoint algorithms that we consider are given in §4.2.1.

2. Form contrast vectors $v_1, \ldots, v_k \in \mathbb{R}^n$, defined so that for arbitrary $y \in \mathbb{R}^n$,

$$v_j^T y = \widehat{d}_j \Big(\frac{1}{|I_{j+1}|} \Big(\sum_{i \in I_{j+1}} y_i \Big) - \frac{1}{|I_j|} \Big(\sum_{i \in I_j} y_i \Big) \Big), \tag{4.2}$$

for $j = 1, \ldots, k$ where $|I_j|$ denotes the cardinality of the set I_j. Hence, $v_j^T Y$ represents the difference between the sample means of segments to right and left of $\widehat{b}_j$,

3. For each $j = 1, \ldots, k$, we test the hypothesis $H_0 : v_j^T \theta = 0$ by rejecting for large values of a statistic $T(Y, v_j)$, which is computed based on knowledge of the changepoint algorithm that produced $\widehat{b}_{1:k}$ in Step 1, and the desired contrast vector (4.2) formed in Step 2. Each statistic yields a p-value under the null (assuming the model (4.1)). The details of $T(Y, v_j)$ are given in Sections 4.2.2 and 4.3.

4. Optionally, we can use Bonferroni correction by multiplying the p-values by k, to account for multiplicity.

It is worth mentioning that several variants of this basic procedure are possible. For example, the number of changepoints k in Step 1 need not be fixed a priori and may be itself estimated from data; the set of estimated changepoints $\widehat{b}_{1:k}$ may be pruned after Step 1 to eliminate changepoints that lie too close to others, and alternative contrast vectors to (4.2) in Step 2 may be used to measure more localized mean changes; these are all briefly described in §4.4. Though not covered in our paper, the p-values from our tests can be inverted to form confidence intervals for population contrasts $v_j^T \theta$ for $j = 1, \ldots, k$ (Lee et al., 2016; Tibshirani et al., 2016).

Contributions. At a more comprehensive level, our contributions in this work are to implement theoretically valid inference tools and provide practical guidance for each combination of the following choices that a typical user might face in a changepoint analysis: the algorithm (BS, WBS, CBS, or FL), number of estimated changepoints k (fixed or data-driven), the null hypothesis model (saturated or selected model, to be explained in §4.2.2), what type of conditioning (plain or marginalized, to be explained in §4.3.3), and the error variance σ^2 (known or unknown). For unknown σ^2, we develop a new hit-and-run sampling algorithm. In §4.4, we summarize the tradeoffs underlying each of these choices.

Finally, as the primary application of our inference tools, we study comparative genomic hybridization (CGH) data, making particular suggestions geared towards this problem throughout the paper. We begin with a motivating CGH data example in the next subsection, and return to it at the end of the paper.

4.1.1 Motivating example: array CGH data analysis

We examine array CGH data from the 14th chromosome of cell line GM01750, one of the 15 datasets from Snijders et al. (2001); more background can be found in Lai et al. (2005) and references therein. Array CGH data are $\log_2$ ratios of dye intensities of diseased to healthy subjects' measurements, mixed across many samples. Normal regions of the gene are thought to have an underlying mean $\log_2$ ratio of zero, and aberrations are regions of

upward or downward departures from zero because the gene in that region has been mutated – duplicated or deleted. The presence and locations of aberrations are well studied in the biomedical literature to be associated with the presence of a wide range of genetically driven diseases – as many types of cancer, Alzheimer, and autism (Bean et al., 2007; Mullighan et al., 2007; Kao et al., 2009; Verhaak et al., 2010). All these aforementioned work use one of the changepoint detection methods we cover in this paper. Accurate inference on top of existing changepoint analyses of array CGH data can serve as an effective screening mechanism for these previous work that can be applied in an automated fashion.

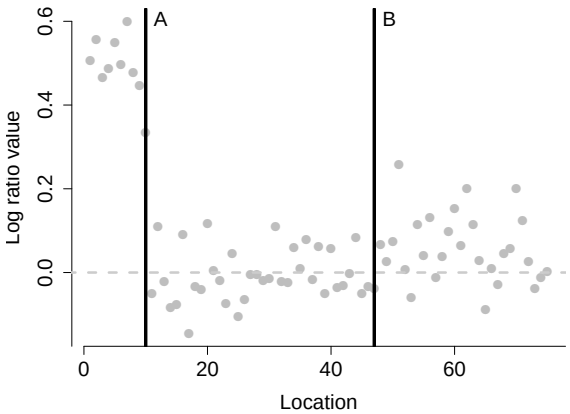
The data is plotted in the left panel of Figure 4.1. Two locations $\widehat{b}_1 < \widehat{b}_2$, marked A and B respectively, were detected by running 2-step WBS. Ground truth in this data set can be defined via an external process called called karyotyping; this is done by Snijders et al. (2001) who finds only one true changepoint at location A. (To be precise, they do not report exact locations of abnormalities, but find a single start-to-middle deviation from zero level.)

Without access to any post-selection inference tools, we might treat locations A and B as fixed, and simply run t-tests for equality of means of neighboring data segments, to the left and right of each location. This is precisely testing the null hypothesis $H_0 : v_j^T \theta = 0$, $j = 1, 2$, where the contrast vectors are as defined in (4.2). P-values from the t-tests are reported in the first row of the table in Figure 4.1: we see that location A has a p-value of $< 10^{-5}$, but location B also has a small p-value of 5×10^{-4}, which is troublesome. The problem is that location B was specifically selected by WBS because (loosely speaking) the sample means to left and right of B are well separated, thus a t-test on location B is likely to be optimistic.

Using the tools we describe shortly, we test $H_0 : v_j^T \theta = 0$, $j = 1, 2$ in two ways: using a *saturated model* and a *selected model* on the mean vector θ. The satured model assumes nothing about θ, while the selected model assumes θ is constant between the intervals formed by A and B. Both tests yield a p-value $< 10^{-5}$ at location A, but only a moderately small p-value at location B. If we were to use the Bonferroni correction at a nominal significance level $\alpha = 0.05$, then we would not reject the null at location B in both cases.

4.1.2 Related work

In addition to the references on general post-selection inference methodology given previously, we highlight the recent work of Hyun et al. (2018a), which studies post-selection inference for the generalized lasso, a generalization of the fused lasso. These authors characterize the polyhedral form of fused lasso selection events, and study inference using contrasts as in (4.2). While writing the current paper, we became aware of the independent contributions of Umezu and Takeuchi (2017), who study multi-dimensional changepoint sequences, but focus problems in which the mean θ has only one changepoint. Aside from these papers, there is little focus on valid inference methods to apply post-detection in changepoint analysis. On the other hand, there is a huge literature on changepoint estimation, and inference for *fixed*



Location	A	B
Karyotype	True	False
Classical t-test	0	5×10^{-4}
Saturated model test	0	0.050
Selected model test	0	0.027

Figure 4.1: Left: array CGH data from the 14th chromosome of fibroblast cell line GM01750, from Snijders et al. (2001). The x-axis denotes the relative index of the genome position, and the y-axis denotes the log ratio in fluorescence intensities of the test and reference samples. The dotted horizontal line denotes a log ratio of 0 for reference. The bold vertical lines denote the locations A and B from running WBS for 2 steps. Right: the p-values using classical (naive) t-tests, saturated model tests, and selected model tests, at each location A and B. The ground truth is also given, as determined by karyotyping. The saturated model test used an estimated noise level σ^2 from the entire 23-chromosome data set. The selected model test was performed in the unknown σ^2 setting.

hypotheses in changepoint problems; we refer to Jandhyala et al. (2013); Aue and Horvath (2013); Horvath and Rice (2014), which collectively summarize the literature.

4.2 Preliminaries

4.2.1 Review: changepoint algorithms

Below we describe the changepoint algorithms that we will study in this paper. We will focus on formulations that run the algorithm for a given number of steps k. In contrast, these algorithms are typically described in the literature as recursively running until internally calculated statistics do not exceed a given threshold level τ. The reason that we choose the former formulation is twofold: first, we feel it is easier for a user to specify a priori a reasonable number of steps k, versus a threshold level τ; second, we can use the method in Hyun et al. (2018a) to adaptively choose the number of steps k and still perform valid inferences. In what follows, we use the notation $y_{a:b} = (y_a, y_{a+1}, \ldots, y_b)$ and $\bar{y}_{a:b} = (b-a+1)^{-1} \sum_{i=a}^{b} y_i$ for a vector y. Similarly, for a set I, $\bar{y}_I = |I|^{-1} \sum_{i \in I} y_i$.

Binary segmentation (BS). Given a data vector $y \in \mathbb{R}^n$, the k-step BS algorithm (Vostrikova, 1981) sequentially splits the data based on the cumulative sum (CUSUM) statistics, defined below. At a step $\ell = 1, \ldots, k$, let $\widehat{b}_{1:(\ell-1)}$ be the changepoints estimated

so far, and let I_j, $j = 1, \ldots, \ell$ be the partition of $\{1, \ldots, n\}$ induced by $\widehat{b}_{1:(\ell-1)}$. Throughout this paper, we use the convention that for $\ell = 1$, $I_1 = \{1, \ldots, n\}$. Intervals of length 1 are discarded. Let s_j and e_j be the start and end indices of I_j. The next changepoint $\widehat{b}_\ell$ and maximizing interval $\widehat{j}_\ell$ are chosen to maximize the absolute CUSUM statistic,

$$\left\{\widehat{j}_\ell, \widehat{b}_\ell\right\} = \underset{\substack{j \in \{1,\ldots,\ell-1\} \\ b \in \{s_j,\ldots,e_j-1\}}}{\operatorname{argmax}} \left|g^T_{(s_j,b,e_j)} y\right|, \quad \text{where}$$

$$g^T_{(s,b,e)} y = \sqrt{\frac{1}{\frac{1}{|e-b|} + \frac{1}{|b+1-s|}}} \left(\bar{y}_{(b+1):e} - \bar{y}_{s:b}\right). \tag{4.3}$$

Additionally, the direction $\widehat{d}_\ell$ of the new changepoint is calculated by the sign of the maximizing absolute CUSUM statistic, $\widehat{d}_\ell = \operatorname{sign}(g^T_{(s_j,b_\ell,e_j)} y)$ for $j = \widehat{j}_{\ell+1}$.

Wild binary segmentation (WBS). The k-step WBS algorithm (Fryzlewicz, 2014) is a modification of BS that calculates CUSUM statistics over randomly drawn segments of the data. Denote by $w = \{w_1, \ldots, w_B\} = \{(s_1, \ldots, e_1), \ldots, (s_B, \ldots, e_B)\}$ a set of B uniformly randomly drawn intervals with $1 \le s_i < e_i \le n$, $i = 1, \ldots, B$. At a step $\ell = 1, \ldots, k$, let J_ℓ to be the index set of the intervals in w which do not intersect with the changepoints $\widehat{b}_{1:(\ell-1)}$ estimated so far. The next changepoint $\widehat{b}_\ell$ and the maximizing interval $\widehat{j}_\ell$ are obtained by

$$\left\{\widehat{j}_\ell, \widehat{b}_\ell\right\} = \underset{\substack{j \in J_\ell \\ b \in \{s_j,\ldots,e_j-1\}}}{\operatorname{argmax}} \left|g^T_{(s_j,b,e_j)} y\right|,$$

where $g^T_{(s,b,e)} y$ is defined in (4.3). Similar to BS, the direction of the changepoint $\widehat{d}_\ell$ is defined by the sign of the maximizing absolute CUSUM statistic.

Circular binary segmentation (CBS). The k-step CBS algorithm (Olshen et al., 2004) specializes in detecting *pairs* of changepoints that have alternating directions. At a step $\ell = 1, \ldots, k$, let $\widehat{a}_{1:(\ell-1)}$, $\widehat{b}_{1:(\ell-1)}$ be the changepoints estimated so far (with the pair a_j, b_j estimated at step j), and let I_j, $j = 1, \ldots, 2\ell + 1$ be the associated partition of $\{1, \ldots, n\}$. Intervals of length 2 are discarded. Let s_j and e_j denote the start and end index of I_j. The next changepoint pair $\widehat{a}_\ell$ and $\widehat{b}_\ell$, and the maximizing interval $\widehat{j}_\ell$, are found by

$$\left\{\widehat{j}_\ell, \widehat{a}_\ell, \widehat{b}_\ell\right\} = \underset{\substack{j \in \{1,\ldots,2(\ell-1)+1)\} \\ a,b \in \{s_j,\ldots,e_j-1\} \,:\, a<b}}{\operatorname{argmax}} \left|g^T_{(s_j,a,b,e_j)} y\right| \quad \text{where} \tag{4.4}$$

$$g^T_{(s,a,b,e)} y = \sqrt{\frac{1}{\frac{1}{|b-a|} + \frac{1}{|e-s-b+a|}}} \Big(\bar{y}_{(a+1):b} - \bar{y}_{\{s:a\} \cup \{(b+1):e\}}\Big). \tag{4.5}$$

As before, the new changepoint direction $\widehat{d}_\ell$ is defined based on the sign of the (modified) CUSUM statistic, $\widehat{d}_\ell = \operatorname{sign}(g^T_{(s_j,a_{\ell+1},b_{\ell+1},e_j)} y)$ for $j = \widehat{j}_{\ell+1}(y)$.

Fused lasso (FL). The fused lasso estimator (Rudin et al., 1992; Tibshirani et al., 2005) is defined by solving the convex optimization problem,

$$\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}|, \tag{4.6}$$

for a tuning parameter $\lambda \geq 0$. The fused lasso can be seen as a k-step algorithm by sweeping the tuning parameter from $\lambda = \infty$ down to $\lambda = 0$. Then, at given values of λ (called knots), the FL estimator sequentially introduces an additional changepoint into the solution of (4.6) (Hoefling, 2010). See Hyun et al. (2018a) for a more in-depth description.

4.2.2 Review: post-selection inference

We briefly review post-selection inference as developed in Lee et al. (2016); Tibshirani et al. (2016); Fithian et al. (2014). Our description here will be cast towards changepoint problems. For clarity, we notationally distinguish between a random vector Y distributed as in (4.1), and y_{obs}, a single data vector we observe for changepoint analysis. When a changepoint algorithm—such as BS, WBS, CBS, or FL—is applied to the data y_{obs}, it selects a particular changepoint model $M(y_{\text{obs}})$. The specific forms of such models are described in §4.3.1; for now we may loosely think of $M(y_{\text{obs}})$ as the estimated changepoint locations and directions made by the algorithm on the data at hand. Post-selection inference revolves around the selective distribution, i.e., the law of

$$v^T Y \mid \Big(M(Y) = M(y_{\text{obs}}),\ q(Y) = q(y_{\text{obs}}) \Big), \tag{4.7}$$

under the null hypothesis $H_0 : v^T \theta = 0$, for any v that is a measurable function of $M(y_{\text{obs}})$, such as in (4.2). Here, $q(Y)$ is a vector of sufficient statistic of nuisance parameters that need to be conditioned on in order to tractably compute inferences based on (4.7). The explicit form of $q(Y)$ differs based on the assumptions imposed on θ under the null model. Broadly, there are two classes of null models we may study: saturated and selected models (Fithian et al., 2014). As shown in the literature, computationally, in either null models, it is important for the selection event $\{y : M(y) = M(y_{\text{obs}})\}$ be polyhedral. This is described in detail in Section 4.3.1, where we show that this holds for BS, WBS, CBS, and FL.

Saturated model. The *saturated model* assumes that Y is distributed as in (4.1) with known error variance σ^2, and assumes nothing about the mean vector θ. We set $q(Y) = \Pi_v^{\perp} Y$, the projection of Y onto the hyperplane orthogonal to v. The selective distribution then becomes the law of

$$v^T Y \mid \Big(M(Y) = M(y_{\text{obs}}),\ \Pi_v^{\perp} Y = \Pi_v^{\perp} y_{\text{obs}} \Big). \tag{4.8}$$

Selected model. The *selected model* again assumes that Y follows (4.1), but additionally assumes that the mean vector θ is piecewise constant with changepoints at the sorted estimated locations $\widehat{c}_{1:k} = \widehat{c}_{1:k}(y_{\text{obs}})$, assuming we have run our changepoint algorithm for k steps. That is, letting s_j and e_j denote the start and end index of interval I_j, we assume

$$\theta_{s_j} = \ldots = \theta_{e_j}, \quad j \in \{1, \ldots, k+1\}.$$

Under this assumption, the law of Y becomes a $(k+1)$-parameter Gaussian distribution. Additionally, with the contrast vector v_j defined as in (4.2), for any fixed $j = 1, \ldots, k+1$, the quantity $v_j^T \theta$ of interest is simply the difference between two of the parameters in this distribution. Let $\mathcal{I}_j = \{1, \ldots, k+1\} \backslash \{j, j+1\}$. Assuming σ^2 is known, the sufficient statistics $q(Y)$ for the nuisance parameters in the Gaussian family are then the sample averages of the appropriate data segments, and the selective distribution becomes the law of

$$\big(\bar{Y}_{I_{j+1}} - \bar{Y}_{I_j}\big) \mid \Big(M(Y) = M(y_{\text{obs}}),\ \bar{Y}_{I_j \cup I_{j+1}} = \big(\bar{y}_{\text{obs}}\big)_{I_j \cup I_{j+1}},\ \bar{Y}_{I_\ell} = \big(\bar{y}_{\text{obs}}\big)_{I_\ell} \text{ for } \ell \in \mathcal{I}_j\Big). \tag{4.9}$$

The appeal of the selected model is that we can properly treat σ^2 as unknown; in this case, we must only additionally condition on the Euclidean norm of y_{obs} to cover this nuisance parameter, and the selective distribution becomes the law of

$$\big(\bar{Y}_{I_{j+1}} - \bar{Y}_{I_j}\big) \mid \Big(M(Y) = M(y_{\text{obs}}),\ \bar{Y}_{I_j \cup I_{j+1}} = \big(\bar{y}_{\text{obs}}\big)_{I_j \cup I_{j+1}},\ \bar{Y}_{I_\ell} = \big(\bar{y}_{\text{obs}}\big)_{I_\ell} \text{ for } \ell \in \mathcal{I}_j, \\ \|Y\|_2 = \|y_{\text{obs}}\|_2\Big). \tag{4.10}$$

4.3 Inference for changepoint algorithms

We describe our contributions that enable post-selection inference for changepoint analyses, beginning with the form of model selection events for common changepoint algorithms. We then describe computational details for saturated and selected model tests, and auxiliary randomization.

4.3.1 Polyhedral selection events

We show that, for each of the BS, WBS, and CBS algorithms, there is a parametrization for their models such that event $\{y : M(y) = M(y_{\text{obs}})\}$ is a polyhedron—in fact a convex cone—of the form $\{y : \Gamma y \geq 0\}$, for a matrix $\Gamma \in \mathbb{R}^{m \times n}$ that depends on $M(y_{\text{obs}})$ (and we interpret the inequality $\Gamma y \geq 0$ componentwise). Throughout the description of the polyhedra for each algorithm, we display the number of rows in Γ since it loosely denotes how "complex" each model selection event is. The same was already shown for FL in Hyun et al. (2018a), and we omit details, but briefly comment on it below. Overall, for a fixed k, the Γ matrices for FL and BS are linear in n, while it is quadratic in n for CBS, and

$O(Bp)$ for WBS using intervals of length p. This number can grow faster than linear in n if $B \geq n$, which is recommended in practice (Fryzlewicz, 2014). All the proofs in this section are provided in Appendix 4.B.

Selection event for BS. We define the model for the k-step BS estimator as

$$M_{1:k}^{\mathrm{BS}}(y_{\mathrm{obs}}) = \big\{\widehat{b}_{1:k}(y_{\mathrm{obs}}),\ \widehat{d}_{1:k}(y_{\mathrm{obs}})\big\},$$

where $\widehat{b}_{1:k}(y_{\mathrm{obs}})$ and $\widehat{d}_{1:k}(y_{\mathrm{obs}})$ are the changepoint locations and directions when the algorithm is run on y_{obs}, as described in Section 4.2.1.

Proposition 31. *Given any fixed $k \geq 1$ and $b_{1:k}, d_{1:k}$, we can explicitly construct Γ where*

$$\big\{y : M_{1:k}^{\mathrm{BS}}(y) = \{b_{1:k}, d_{1:k}\}\big\} = \{y : \Gamma y \geq 0\},$$

where Γ has $2\sum_{\ell=1}^{k}(n-\ell-1)$ rows.

Selection event for WBS. We define the model of the k-step WBS estimator as

$$M_{1:k}^{\mathrm{WBS}}(y_{\mathrm{obs}}, w) = \big\{\widehat{b}_{1:k}(y_{\mathrm{obs}}),\ \widehat{d}_{1:k}(y_{\mathrm{obs}}),\ \widehat{j}_{1:k}(y_{\mathrm{obs}})\big\},$$

where w is the set of B intervals that the algorithm uses, $\widehat{b}_{1:k}(y_{\mathrm{obs}})$ and $\widehat{d}_{1:k}(y_{\mathrm{obs}})$ are the changepoint locations and directions, and $\widehat{j}_{1:k}(y_{\mathrm{obs}})$ are the maximizing intervals. Note that unlike BS, the maximizing intervals $\widehat{j}_{1:k}$ are part of WBS's model.

Proposition 32. *Given any fixed $k \geq 1$ and $\{w, b_{1:k}, d_{1:k}, j_{1:k}\}$, we can explicitly construct Γ where*

$$\big\{y : M_{1:k}^{\mathrm{WBS}}(y, w) = \{b_{1:k}, d_{1:k}, j_{1:k}\}\big\} = \big\{y : \Gamma y \geq 0\big\}.$$

The number of rows in Γ will vary depending on the configuration of w and $b_{1:k}$, but if each of the B intervals in w has length p, it will be at most $2\sum_{\ell=1}^{k}((B-\ell)\cdot(p-1)+(p-2))$.

Selection event for CBS. We define the model for the k-step CBS estimator as

$$M_{1:k}^{\mathrm{CBS}}(y_{\mathrm{obs}}) = \big\{\widehat{a}_{1:k}(y_{\mathrm{obs}}),\ \widehat{b}_{1:k}(y_{\mathrm{obs}}),\ \widehat{d}_{1:k}(y_{\mathrm{obs}})\big\},$$

where now $\widehat{a}_{1:k}(y_{\mathrm{obs}})$ and $\widehat{b}_{1:k}(y_{\mathrm{obs}})$ are the pairs of estimated changepoint locations, and $\widehat{d}_{1:k}(y_{\mathrm{obs}})$ are the changepoint directions, as described in Section 4.2.1.

Proposition 33. *Given any fixed $k \geq 1$ and $\{a_{1:k}, b_{1:k}, d_{1:k}\}$, we can explicitly construct Γ where*

$$\big\{y : M_{1:k}^{\mathrm{CBS}}(y, w) = \{a_{1:k}, b_{1:k}, d_{1:k}\}\big\} = \big\{y : \Gamma y \geq 0\big\}.$$

Let $I_j^{(\ell)}$ denote the jth interval of $B(\ell)$ intervals remaining for an intermediate step $\ell \in \{1, \ldots, k\}$, and let $C(x, 2) = \binom{x}{2}$. Then Γ has a number of rows equal to

$$2\sum_{\ell=1}^{k}\Big[\sum_{j=1}^{B(\ell)} C(|I_j^{(\ell)}|-1, 2) - 1\Big].$$

Selection events for FL, and a brief comparison. The model for the k-step FL estimator is

$$M^{\mathrm{FL}}_{1:k}(y_{\mathrm{obs}}) = \big\{\widehat{b}_{1:k}(y_{\mathrm{obs}}),\ \widehat{d}_{1:k}(y_{\mathrm{obs}}),\ \widehat{R}_{1:k}(y_{\mathrm{obs}})\big\},$$

where $\widehat{b}_{1:k}(y)$ and $\widehat{d}_{1:k}(y)$ are changepoint locations and directions, and $\widehat{R}_\ell(y) \in \mathbb{R}^{n-\ell}, \ell = 1, \ldots, k$ whose elements represent signs of a certain statistic $h_i(y)$ calculated at location i in competition for maximization with $\widehat{b}_\ell$ at step ℓ. These statistics $h_i(y)$ are weighted mean differences at location i and are analogous to CUSUM statistics in BS. Hyun et al. (2018a) makes this representation more explicit, proving that for any fixed $k \geq 1$ and $b_{1:k}, d_{1:k}, R_{1:k}$, we can explicitly construct Γ such that

$$\big\{y : M^{\mathrm{FL}}_{1:k}(y) = \{b_{1:k}, d_{1:k}, R_{1:k}\}\big\} = \{y : \Gamma y \geq 0\},$$

where Γ has the same number of rows as a k-step BS event.

4.3.2 Computation of p-values

Given a precise description of the polyhedral selection event $\{y : M(y) = M(y_{\mathrm{obs}})\}$, we can describe the methods to compute the p-value, i.e. the tail probability of the selective distributions described in §4.2.2. Without loss of generality, all of our descriptions will be specialized to testing the null hypothesis of $H_0 : v^T\theta = 0$ against the one-sided alternative $H_1 : v^T\theta > 0$. For saturated model tests, this exact calculation has been developed in previous work and we review it as it is relevant to our contributions on increasing its power. For selected model tests, an approximation was described in previous work, but we develop a new and more intuitive hit-and-run sampler that has not been implemented before.

Saturated model tests: exact formulae. As shown in Lee et al. (2016) and Tibshirani et al. (2016), the saturated selective distribution (4.8) has a particularly computationally convenient distribution when Y is Gaussian and the model selection event $\{y : M(y) = M(y_{\mathrm{obs}})\}$ is a polyhedral set in y. In this case, the law of (4.8) is a *truncated Gaussian* (TG), whose truncation limits depend only on $\Pi_v^\perp y_{\mathrm{obs}}$, and can be computed explicitly. Its tail probability can be computed in closed form (without Monte Carlo sampling). That is, the probability that $v^TY \geq v^Ty_{\mathrm{obs}}$ under the law of (4.8) is exactly equal to

$$(\Phi(\mathcal{V}_{\mathrm{up}}/\tau) - \Phi(v^Ty_{\mathrm{obs}}/\tau))/(\Phi(\mathcal{V}_{\mathrm{up}}/\tau) - \Phi(\mathcal{V}_{\mathrm{lo}}/\tau)) \tag{4.11}$$

where $\Phi(\cdot)$ represents the standard Gaussian CDF, $\tau = \sigma^2\|v\|_2^2$, $\rho = \Gamma v/\|v\|_2^2$ and

$$\mathcal{V}_{\mathrm{lo}} = v^Ty_{\mathrm{obs}} - \min_{j:\rho_j>0}\big(\Gamma y_{\mathrm{obs}}\big)_j/\rho_j, \quad \text{and} \quad \mathcal{V}_{\mathrm{up}} = v^Ty_{\mathrm{obs}} - \max_{j:\rho_j<0}\big(\Gamma y_{\mathrm{obs}}\big)_j/\rho_j. \tag{4.12}$$

This above equation is commonly referred as the TG statistic. Since this statistic is a pivot, it is the p-value used for the saturated model test.

Algorithm 2: MCMC hit-and-run algorithm for selected model test with unknown σ^2

Choose a number M of iterations and set $y^{(0)} = y_{\text{obs}}$.
for $m \in \{1, \ldots, M\}$ **do**
  Uniformly sample two unit vectors s and t in the nullspace of A.
  Compute the set $\mathcal{I} \subseteq [-\pi/2, \pi/2]$ that intersects the set

$$\Big\{y \ : \ y = y^{(m-1)} + r(\omega)\sin(\omega) \cdot s + r(\omega)\cos(\omega) \cdot t \quad \text{for any } \omega \in [-\pi/2, \pi/2]\Big\},$$

  for the radius function $r(\omega) = -2(y^{(m-1)})^T(\sin(\omega) \cdot s + \cos(\omega) \cdot t)$, with the polyhedral set implied by the selected model $M(y_{\text{obs}})$ based on §4.3.1.
  Uniformly sample $\omega^{(m)}$ from $\mathcal{I}$ and form the next sample

$$y^{(m)} = y^{(m-1)} + r(\omega^{(m)})\sin(\omega^{(m)}) \cdot s + r(\omega)\cos(\omega^{(m)}) \cdot t.$$

end
Return the approximate for the tail probability of (4.10), $\sum_{m=1}^{M} \mathbf{1}[v^T y^{(m)} \geq v^T y_{\text{obs}}]/M$.

Selected model tests: hit-and-run sampling. To compute the p-value for selected model tests, Fithian et al. (2015) proposed a hit-and-run strategy for sampling from the distribution for the known σ^2 setting, (4.9). This was implemented by the authors, and we briefly review the details in Appendix 4.C. For the unknown σ^2 setting, Fithian et al. (2014) developed an importance sampling strategy for sampling the distribution (4.10). However, we find that an alternative and intuitive hit-and-run strategy can be adapted to the unknown σ^2 setting and implement this as a new algorithm.

Given a changepoint $j \in \{1, \ldots, k\}$, observe that we can design a segment test contrast v where sampling from (4.10) is equivalent to sampling uniformly from the set

$$\Big\{v^T Y : M(Y) = M(y_{\text{obs}}),\ \|Y\|_2 = \|y_{\text{obs}}\|_2,\ \bar{Y}_{I_j \cup I_{j+1}} = \big(\bar{y}_{\text{obs}}\big)_{I_j \cup I_{j+1}},\ \bar{Y}_{I_\ell} = \big(\bar{y}_{\text{obs}}\big)_{I_\ell} \text{ for } \ell \in \mathcal{I}_j\Big\}. \tag{4.13}$$

Note that the above set no longer depends on θ or σ^2. This is because we conditioned all the relevant sufficient statistics under the selected model. Our hit-and-run sampler then sequentially draws samples $v^T Y$ from the above set. For notational convenience, observe that the last k constraints in (4.13) can be rewritten as $AY = Ay_{(\text{obs})}$ for some matrix $A \in \mathbb{R}^{k \times n}$. Our new hit-and-run algorithm is then shown in Algorithm 2.

4.3.3 Randomization and marginalization

We apply the ideas of randomization in Tian and Taylor (2015) that improve the power of selective inference to changepoint algorithms and devise explicit samplers. We investigate two specific forms of randomization: randomization over additive noise and randomization over random intervals. We specialize the following descriptions to saturated models. We note that similar randomization of selected model inferences is also possible but is doubly computationally burdensome.

Marginalization over additive noise. Tian and Taylor (2015) shows that performing inference based on the selected model $M(y_{\text{obs}} + w_{\text{obs}})$ where w_{obs} is additive noise and then marginalizing over W leads to improved power. Here, w_{obs} is a realization of a random component W sampled from $\mathcal{N}(0, \sigma^2_{\text{add}} I_n)$, where $\sigma^2_{\text{add}} > 0$ is set by the user. Fithian et al. (2014) provides a mathematical framework for pursuing such randomization, stating that less conditioning results in an increase in Fisher information. For additive noise, the above model selection event is:

$$\{y : \Gamma(y + w_{\text{obs}}) \geq 0\} = \{y : \Gamma y \geq -\Gamma w_{\text{obs}}\}.$$

This means the new polyhedron formed by the model selection event based on perturbed data $y_{\text{obs}} + w_{\text{obs}}$ is slightly shifted.

Porting the ideas of Tian and Taylor (2015) to our setting, to test the one-sided null hypothesis $H_0 : v^T\theta = 0$, we want to compute the following tail probability of the marginalized selective distribution,

$$T(y_{\text{obs}}, v) = \mathbb{P}\Big(v^T Y \geq v^T y_{\text{obs}} \;\Big|\; \Big(M(Y + W) = M(y_{\text{obs}} + W),\ \Pi_v^{\perp} Y = \Pi_v^{\perp} y_{\text{obs}}\Big)\Big). \quad (4.14)$$

It is hard to directly compute this. However, the formulas in (4.11) and (4.12) give us exact formulas to compute the non-marginalized tail-probabilities,

$$T(y_{\text{obs}}, v, w_{\text{obs}}) = \mathbb{P}\Big(v^T Y \geq v^T y_{\text{obs}} \;\Big|\; \Big(M(Y{+}W) = M(y_{\text{obs}}{+}W),\ \Pi_v^{\perp} Y = \Pi_v^{\perp} y_{\text{obs}},\ W = w_{\text{obs}}\Big)\Big).$$

The following proposition shows that we can compute $T(y_{\text{obs}}, v)$ by reweighting instances of $T(y_{\text{obs}}, v, w_{\text{obs}})$ via importance sampling. Here, let $E_1 = \mathbf{1}[M(Y + W) = M(y_{\text{obs}} + W)]$ and $E_2 = \mathbf{1}[\Pi_v^{\perp} Y = \Pi_v^{\perp} y_{\text{obs}}]$.

Proposition 34. *Let Ω denote the support of the random component W. If the distribution of W is independent of the random event E_2,* (4.14) *can be exactly computed as*

$$T(y_{\text{obs}}, v) = \int_{\Omega} T(y_{\text{obs}}, v, w_{\text{obs}}) \cdot a(w_{\text{obs}})\, dP_W(w_{\text{obs}}) = \frac{\int_{\Omega} \Phi\big(\mathcal{V}_{up}/\tau\big) - \Phi\big(v^T y_{\text{obs}}/\tau\big)\, dP_W(w_{\text{obs}})}{\int_{\Omega} \Phi\big(\mathcal{V}_{up}/\tau\big) - \Phi\big(\mathcal{V}_{lo}/\tau\big)\, dP_W(w_{\text{obs}})}. \quad (4.15)$$

where the weighting factor is $a(w_{\text{obs}}) = \mathbb{P}(W = w_{\text{obs}} | E_1, E_2)/\mathbb{P}(W = w_{\text{obs}})$.

Algorithm 3: Marginalizing over additive noise

```
Choose a number T of trials.
for t ∈ {1, ..., T} do
    Sample the additive noise w_j from N(0, σ²_add I_n).
    Compute k(w_t) and g(w_t).
end
Return the approximate for the tail probability (4.15), Σ_{t=1}^T k(w_t) / Σ_{t=1}^T g(w_t).
```

The first equality in (4.15) demonstrates the reweighting of $T(y_{\text{obs}}, v, w_{\text{obs}})$, but the second equality gives a sampling strategy where we approximate the integrals. Algorithm 3 describes this, where for one realization w_{obs}, we let $k(w_{\text{obs}})$ and $g(w_{\text{obs}})$ denote the integrand of the last term's numerator and denominator in (4.15) respectively. As mentioned in (4.11), these can be computed exactly.

Marginalization over WBS intervals. In contrast to the above setting where W represents Gaussian noise, in wild binary segmentation described in §4.2.1, W represents the set of B randomly drawn intervals. Observe that Proposition 34 still applies to this setting, where $M(y_{\text{obs}} + w_{\text{obs}})$ is now replaced with $M(y_{\text{obs}}, w_{\text{obs}})$, as described in §4.3.1. However, unlike the additive noise setting, the maximizing intervals $\widehat{j}_{1:k}$ in the model $M(y_{\text{obs}}, w_{\text{obs}})$ are embedded in the construction of the matrix Γ representing the polyhedra. This prevents a naive sampling of B new intervals. To overcome this, let $\{W_{\widehat{j}_1}, \ldots, W_{\widehat{j}_k}\}$ be the maximizing intervals. We sample a new set of all other intervals, W_ℓ for $\ell \in \{1, \ldots, B\} \backslash \{\widehat{j}_1, \ldots, \widehat{j}_k\}$. Specifically, for each of such intervals $W_\ell = (s_\ell, \ldots, e_\ell)$, s_ℓ and e_ℓ are sampled uniformly between 1 to n where $s_\ell < e_\ell$. After all $B - k$ intervals are resampled, a check is performed to ensure that $\{W_{\widehat{j}_1}, \ldots, W_{\widehat{j}_k}\}$ are still the maximizing intervals when WBS is applied again to y_{obs}. The full algorithm is similar to Algorithm 3 and hence deferred to Appendix 4.C.

4.4 Practicalities and extensions

The above sections formalize the mechanisms to perform selective inference with respect to the basic procedure highlighted in §4.1. We now briefly summarize all the combination of choices mentioned in this work that the user faces based on the methods developed in the above sections and their practical impact.

4.4.1 Practical considerations

There are some practical choices that the user needs to make when implementing the procedure. Here, we summarize these choice, as alluded to in §4.1.

- **Algorithm** (BS, WBS, CBS and FL): FL and BS have similar mechansims, but BS has a simpler mechanism and a less complex selection event, potentially giving higher post-selection conditional power. CBS is specialized for pairs of changepoints, and WBS specializes in localized changepoint detection compared to BS, but both have higher computational burden due to their more complex polyhedra.

- **Conditioning** (Plain or marginalized): Marginalizing over a source of randomness yields tests with higher power than plain inference, but at two costs: increased computational burden due to MCMC sampling being required, and worsened detection ability when using additive noise marginalization. Also, the marginalized p-values are subject to the sampling randomness, and the number of trials T needed to reduce the p-values' intrinsic variability scales with σ^2_{add}.

- **Number of estimated changepoints** k (Fixed or data-driven): As currently described in §4.2.1, we described methods to find a fixed number of changepoints k. However, we can adopt stopping rules from Hyun et al. (2018a) to adaptively choose k. This increases the complexity of the polyhedra compared to those in §4.3.1, leading to lower statistical power than its fixed-k counterpart. This is shown in Appendix 4.D.

- **Assumed null model** (Saturated or selected): As mentioned in §4.2.2, selected model tests are valid under a stricter set of assumptions but often yield higher power. Computationally, saturated model tests are often simpler to perform than selected model tests due to the closed form expression of the tail probability.

- **Error variance** σ^2 (Known or unknown): Saturated model tests require σ^2 to be known. In practice, we need to estimate it in-sample from a reasonable changepoint mean fitted to the same data, or estimated out-of-sample on left-out data. Selected model tests have the advantage of not requiring knowledge of σ^2, but require a larger computational burden, as mentioned in §4.3.2.

4.4.2 Extensions

As mentioned in Hyun et al. (2018a), there are many practically-motivated extensions to the baseline procedure mentioned in §4.1 to either improve power or interpretability. We highlight these below. All of these extensions will still give proper Type-I error control under the appropriate null hypotheses.

- **Designing linear contrasts**: The user can make many types of contrast vectors v to fit their analysis, in addition to the segment test contrasts (4.2), as long as it measurable with respect to $M(y_{\text{obs}})$. One example is the spike test from (Hyun et al., 2018a) of single location mean changes. For CNV analysis, it could be useful to test regions between an adjacent pair of changepoints away from the immediately surrounding regions. Also, a step-sign plot (a plot that shows the locations and direction of the changepoints, but not their magnitude) can help the user design contrasts (Hyun et al., 2018a).
- **Post-processing the estimated changepoints**: Multiple detected changepoints too close to one another can hurt the power of segment tests. Post-processing the estimated changepoints based on decluttering (Hyun et al., 2018a) or filtering (Lin et al., 2017) so the new set of changepoints are well-separated can lead to contrasts that yield higher power. We show empirical evidence of this improving power of the fused lasso, in Appendix 4.C.6.
- **Pre-cutting**: We can also modify all the algorithms in §4.2.1 to start with an initial existing set of changepoints. This is useful in CGH analyses, when it is not meaningful to consider segments that start in one chromosome and end in another. By pooling information in this manner from separate chromosomal regions, the pre-cut analysis is an improvement over conducting separate analyses in individual chromosomes.

4.5 Simulations

4.5.1 Gaussian simulations

In this section, we show simulation examples to demonstrate properties of the segmentation post-selection inference tools presented in the current paper. The mean θ consists of two alternating-direction changepoints of size δ in the middle as in (4.16), chosen to be a realistic example of mutation phenomena as observed in array CGH datasets (Snijders et al., 2001). Here, the sample size $n = 200$ is chosen to be in the scale of the chromosomal data. We model this using the equation below,

$$\textbf{Middle mutation:}\ \text{for } i \in 1, \ldots, n, \quad y_i \sim \mathcal{N}(\theta_i, 1), \ \ \theta_i = \begin{cases} \delta & \text{if } 101 \leq i \leq 140 \\ 0 & \text{if otherwise,} \end{cases} \tag{4.16}$$

for the signal size $\delta \in \{0, 0.25, 0.5, 1, 2, 4\}$ with noise level $\sigma^2 = 1$.

Methodology. In the following simulations, we consider the following four estimators (BS, WBS, CBS and FL), each run for two steps. We perform saturated model tests on each estimator, but only perform selected model tests on BS and FL for simplicity, for both known and unknown noise parameter σ^2. We use the basis procedure outlined in §4.1 with a significance level of $\alpha = 0.05$. Throughout the entire simulation suite, the empirical standard

deviation in each of the power curves and detection probabilities is less than 0.02. For each method, for each signal-to-noise size δ, we run more than 250 trials.

Calculating power. Since the tests are performed only when a changepoint is selected, it is necessary to separate the detection ability of the estimator from power of the test. To that end, we define the following quantities,

$$\text{Conditional power} = \frac{\#\text{ correctly detected \& rejected}}{\#\text{ correctly detected}} \tag{4.17}$$

$$\text{Detection probability} = \frac{\#\text{ correctly detected}}{\#\text{ tests conducted}} \tag{4.18}$$

$$\text{Unconditional power} = \text{Detection} \times \text{Conditional power} \tag{4.19}$$

The overall power of an inference tool can only be assessed by examining the conditional and unconditional power together. We consider a detection to be correct if it is within ± 2 of the true changepoint locations.

Power comparison across signal sizes δ. For saturated model tests, we perform additive-noise inferences using Gaussian $\mathcal{N}(0, \sigma^2_{\text{add}})$ with $\sigma_{\text{add}} = 0.2$ for BS, FL, and CBS. For WBS, we employ the randomization scheme as described in §4.3.3 with $B = n$. With the metrics in (4.18)-(4.19), we examine the performance of the four methods. The solid lines in Figure 4.2 show the "plain" method where model selection based on $M(y_{\text{obs}})$. The dotted lines show the marginalized counterparts where the model selection is $M(y_{\text{obs}}, W)$, margnialized over W.

WBS and CBS have higher conditional and unconditional power than BS. This is expected since the former two are more adept for localized change-points of alternating directions. FL noticeably under-performs in power compared to segmentation methods. This is partially caused by FL's detection behavior, and can be explained by examining alternative measures of detection and improved with post-processing. This investigation is deferred to Appendix 4.C.6. The marginalized versions of each algorithm have noticeably improved power, but almost unnoticeably worse detection than their non-randomized, plain versions (middle panel of Figure 4.2). Combined, in terms of unconditional power, marginalized inferences clearly dominate their plain counterparts.

Selected model inference simulations are shown in Figure 4.3. Surprisingly, there is an almost inconceivable drop in power from unknown σ^2 to known σ^2. Compared to the saturated model tests in Figure 4.2, there is smaller power gap between FL and BS. Also, selected model tests appear to have higher power than saturated model tests. In general however, it is hard to compare the power of saturated and selected models due to the clear difference in model assumptions.

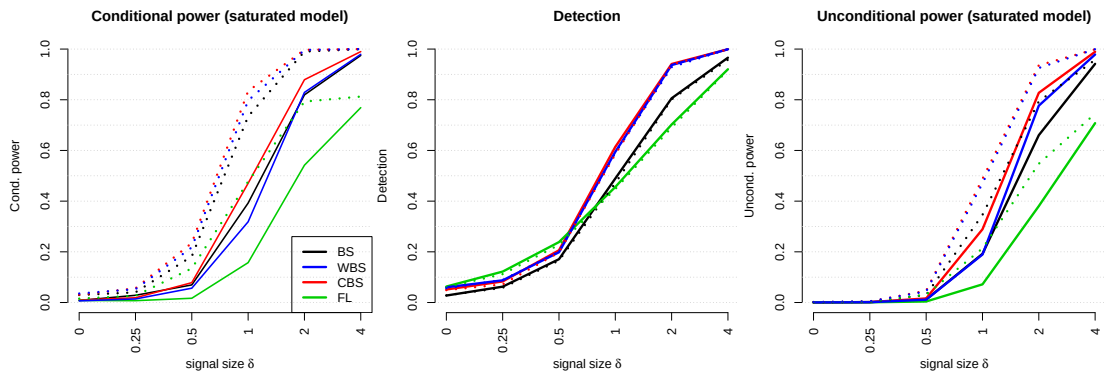


Figure 4.2: Data was simulated from two settings over signal size $\delta \in (0, 4)$ with $n = 200$ data points. Several two-step algorithms (WBS, SBS, CBS, FL) were applied, and post-selection segment test inference was conducted on the resulting two detected changepoints from each method. The dotted lines are the marginalized versions of each test.

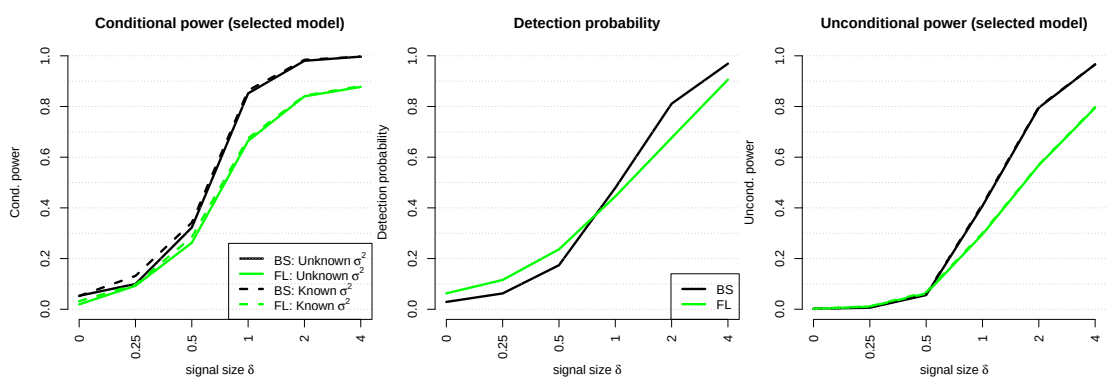


Figure 4.3: Setup similar to Figure 4.2 but for selected model tests. Only BS (black) and FL (green) are shown. but the selected model test is applied to both known (dashed line) and unknown noise parameter σ^2 (solid line).

As an aside, additional simulations to verify that our procedure gives uniform p-values under the global null $\theta = 0$ as well as comparison against sample splitting are given in Appendix 4.C.3.

4.5.2 Pseudo-real simulation with heavy tails

We present pseudo-real datasets based on a single chromosome – chromosome 9 in GM01750 – in order to investigate how heavy-tailed distributions affect our inferences. We only present saturated model tests for brevity. From the original data, we estimate a 1-changepoint mean θ, shown in the bold red line in Figure 4.4, and residuals r, both based on a fitted 1-step wild binary segmentation model. The QQ plot shows that these residuals have heavier tails than a Gaussian (panel A of Figure 4.4), and are close in distribution to a Laplacian. This motivates us to generate synthetic data $y = \theta + \epsilon$ by adding noise ϵ in three ways:

1. Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ (black),
2. Laplace noise $\epsilon \sim \text{Laplace}(0, \sigma/\sqrt{2})$ (green), and
3. Bootstrapped residuals, $\epsilon = b(r)$, where $b(\cdot)$ samples the residuals with replacement (red).

We then investigate the behavior of saturated model tests after a 3-step binary segmentation across all three types of noises when the null hypothesis $H_0 : v^T\theta = 0$ is true. To set σ^2 for these saturated model tests, we compute the empirical variance after fitting a pre-cut 10-step wild binary segmentation across the entire cell line. The results are shown in Figure 4.4. Exactly valid null p-values would follow the theoretical $U(0, 1)$ distribution, optimistic (superuniform) p-values would lie below the diagonal, and conservative (subuniform) p-values would lie above the diagonal. We see that the inferences are exactly valid with Gaussian noise but is optimistic with both Laplacian noise and bootstrapped residuals (panel B of Figure 4.4).

To overcome this optimism, we modify the *bootstrap substitution method* (Tibshirani et al., 2018). Let β denote $\bar{\theta}$, the grand mean of θ. Originally, the authors' main idea is to approximate the law of $v^T Y$ used to construct the TG statistic (4.11) with the bootstrapped distribution of $v^T(Y - \beta)$ by bootstrapping the residuals, $y - \bar{y}$. Here, the empirical grand mean $\bar{y}$ represents the simplest model with no changepoints. While this estimate will usually restore validity, it is expected to produce overly conservative p-values if there exist *any* changepoints (panel C of Figure 4.4).

Hence, we instead consider the bootstrapped distribution of $v^T(Y - \theta)$, by bootstrapping the residuals, $y - \widehat{\theta}$, where $\widehat{\theta}$ is a piecewise constant estimate of θ. For our instance, we use a k-step binary segmentation model to estimate $\widehat{\theta}$, where we choose k using two-fold cross validation from a two-fold split of the data y into odd and even indices. This procedure is not valid in general and should be used with caution. In order to combat the main risk of

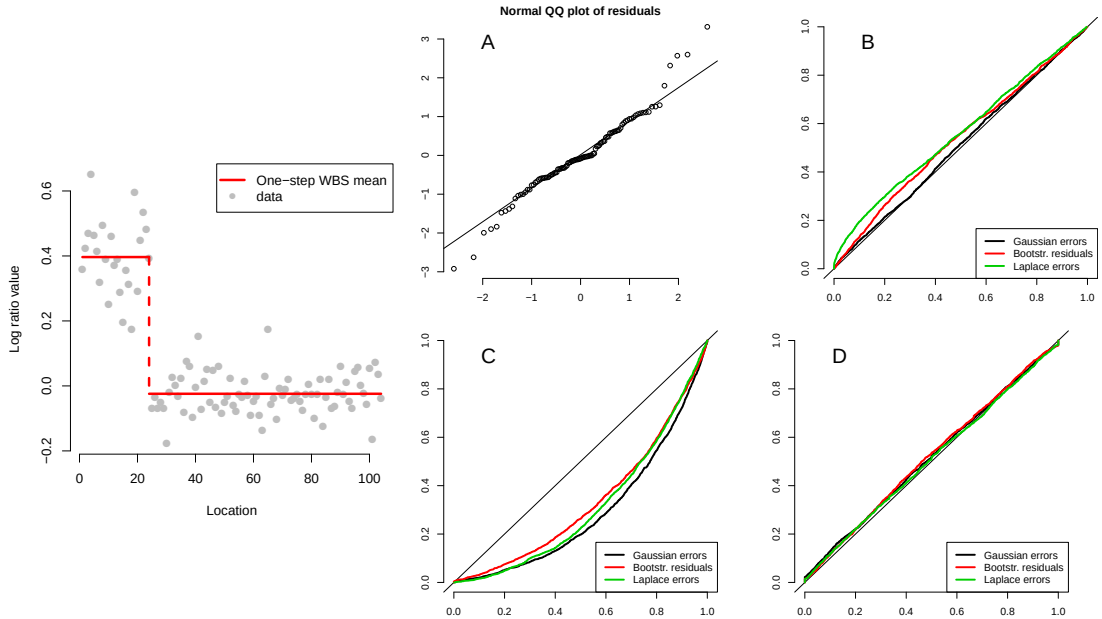


Figure 4.4: (Left) Bootstrapped residuals added to the artificially constructed mean, generated from chromosome 9 in GM01750. (Panel A): QQ plot of residuals. The remaining 3 panels show the p-values of saturated model tests under three different noise models, Gaussian (black), bootstrapped residuals (red) and Laplacian (green). (Panel B): Application of vanilla saturated model tests (no modifications). (Panel C): P-values after using the bootstrap substitution method (Tibshirani et al., 2018). (Panel D): P-values after using our modified bootstrap substitution method that involves bootstrapping $y - \widehat{\theta}$ instead of $y - \bar{y}$.

over-fitting of $\widehat{\theta}$, we may further modify this procedure by excluding shorter segments in $\widehat{\theta}$ prior to bootstrapping. For our dataset, these potential downsides do not seem to come to fruition in practice. At the sample size $n \simeq 100$ and signal-to-noise ratio of our current dataset, the resulting p-values in both heavy-tailed and Gaussian data are convincingly uniform (panel D of Figure 4.4).

4.6 Copy Number Variation (CNV) data application

The datasets we study in this paper are originally from Snijders et al. (2001), and have been studied by numerous works in the statistics literature, e.g. Lai et al. (2008); Hao et al. (2013). Each dataset consists of individual cell lines with $2,000$ measurements or more across 23 chromosomes. Our analysis focuses on middle-to-middle duplication, the setting that was studied in §4.5.

In our analysis, we use a 4-step wild binary segmentation and perform marginalized saturated model tests on two cell lines GM01524 and GM01750 in Figure 4.5. Recall that the 14th chromosome of the latter cell line was shown in Figure 4.1. As decribed in §4.4, we pre-cut both analyses at chromosome boundaries since the ordering of chromosomes 1 through 23 is essentially arbitrary. In GM01524, we can see that the our choice of methods – segment test inferences on changepoints recovered from pre-cut wild binary segmentation, after decluttering – deems two changepoint locations A and B of alternating directions in chromosome 6 to be significant, and two other locations to be spurious, at the signifance level $\alpha = 0.05$ after Bonferroni correction. This result is consistent with karyotyping results of a single middle-to-middle duplication. Likewise, in GM01750, the wild binary segmentation inference correctly identified the two start-to-middle duplications in chromosomes 9 and 14 which were confirmed with karyotyping, and correctly invalidated the rest.

4.7 Conclusions

We have described an approach to conduct post-selection inference on changepoints detected by common segmentation algorithms, using the same data for detection and testing. Through simulations, we demonstrated the detection probability and power over signal-to-noise ratios in a variety of settings, as well as our tools' robustness to heavy-tailed data. Finally, we demonstrated the application in array CGH data, where we show that our methods effectively provide a statistical filter that retains the changepoints that validated by karyotyping and discards the rest. Future work in this area could improve the practical applicability of these methods such extending these methods to next-generation sequencing.

4.A Code

The code to perform estimation as well as saturated model tests are in `https://github.com/robohyun66/binseginf`, while the code to perform selected model tests are additionally in `https://github.com/linnylin92/selectiveModel`.

4.B Additional proofs

4.B.1 Proof of Proposition 31, (BS)

Proof. When $k = 1$, $2(n-2)$ linear inequalities characterize the single changepoint model $\{b_1, d_1\}$:

$$d_1 \cdot g^T_{(1,b_1,n)} y \geq g^T_{(1,b,n)} y, \quad \text{and} \quad d_1 \cdot g^T_{(1,b_1,n)} y \geq -g^T_{(1,b,n)} y, \quad b \in \{1, \ldots, n-1\} \backslash \{b_1\}.$$

Now by induction, assume we have constructed a polyhedral representation of the selection event up through step $k-1$. All that remains is to characterize the kth estimated changepoint

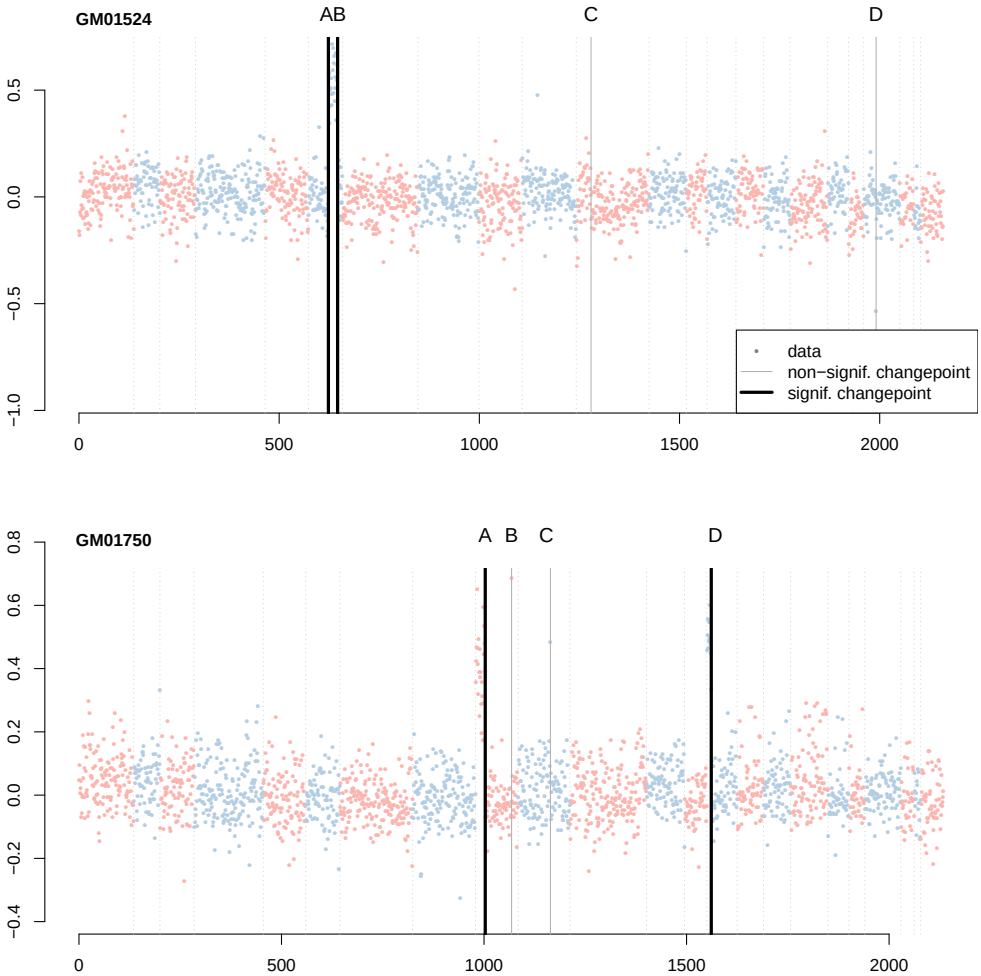


Figure 4.5: "Pre-cut" changepoint inference using saturated model tests for wild binary segmentation marginalized over random intervals conducted on two cell lines, from Snijders et al. (2001). Data points are colored in two alternating tones, to visually depict the chromosomal boundaries. For each cell line, the letters A through D denote the estimated changepoints, $\widehat{b}_1$ through $\widehat{b}_4$ respectively. The bolded lines denote changepoints that were rejected under the null hypothesis $H_0 : v^T\theta = 0$ at a Type-I error control level $\alpha = 0.05$ after Bonferroni-correction. (Top): The analysis for the cell line GM01524, with all 23 chromosomes shown. (Bottom): The same setup as above, but for the cell line GM01750.

and direction $\{b_k, d_k\}$ by inequalities that are linear in y. This can be done with $2(n-k-1)$ inequalities. To see this, assume without a loss of generality that the maximizing interval is $j_k = k$; then $\{b_k, d_k\}$ must satisfy the $2(|I_k|-2)$ inequalities

$$d_k \cdot g^T_{(s_k,b_k,e_k)} y \geq g^T_{(s_k,b,e_k)} y \quad \text{and} \quad d_k \cdot g^T_{(s_k,b_k,e_k)} y \geq -g^T_{(s_k,b,e_k)} y, \quad b \in \{s_k, \ldots, e_k - 1\} \backslash \{b_k\}.$$

For each interval I_ℓ, $\ell = 1, \ldots, k-1$, we also have $2(|I_\ell|-1)$ inequalities

$$d_k \cdot g^T_{(s_k,b_k,e_k)} y \geq g^T_{(s_\ell,b,e_\ell)} y \quad \text{and} \quad d_k \cdot g^T_{(s_k,b_k,e_k)} y \geq -g^T_{(s_\ell,b,e_\ell)} y, \quad b \in \{s_\ell, \ldots, e_\ell - 1\}.$$

The last two displays together completely determine $\{b_k, d_k\}$, and as $\sum_{\ell=1}^k |I_\ell| = n$, we get our desired total of $2(n-k-1)$ inequalities. □

4.B.2 Proof of Proposition 32, (WBS)

Proof. The construction of Γ is basically the same as that for BS in Proposition 31; the only difference is that, at step k, the inequalities defining the new rows of Γ are based on the intervals w_{j_k} and w_ℓ, $\ell \in J_k \backslash \{j_k\}$, instead of I_{j_k} and I_ℓ, $\ell \neq j_k$, respectively. To compute the upper bound on the number of rows m, observe that in step $\ell \in \{1, \ldots, k\}$, there are at most $B - \ell + 1$ intervals remaining. Among these, the interval j_k contributes $p-2$ inequalities, and the remaining $B - \ell$ intervals contributes $p-1$ inequalities. □

4.B.3 Proof of Proposition 33, (CBS)

Proof. The proof follows similarly to the proof of Proposition 31. Observe that for any $k' < k$, the model $M^{\text{CBS}}_{1:k'}(y_{\text{obs}})$ is strictly contained in the model $M^{\text{CBS}}_{1:k}(y_{\text{obs}})$. Hence, we can proceed using induction, and let b_i for $i \in \{1, \ldots, k\}$ denote $\widehat{b}_i$ for simplicity, and do the same for a_i, d_i and j_i. Let $C(x, 2) = \binom{x}{2}$ for simplicity as well.

For $k = 1$, the following $2 \cdot (C(n-1, 2) - 1)$ inequalities characterize the selection of the changepoint model $\{a_1, b_1, d_1\}$,

$$d_1 \cdot g^T_{(1,a_1,b_1,n)} y \geq g^T_{(1,r,t,n)} y, \quad \text{and} \quad d_1 \cdot g^T_{(1,a_1,b_1,n)} y \geq -g^T_{(1,r,t,n)} y,$$

for all $r, t \in \{1, \ldots, n-1\}$ where $r < t$, $r \neq a_1$ and $t \neq b_1$.

By induction, assume we have constructed the polyhedra for the model, $M^{\text{CBS}}_{1:(k-1)}(y_{\text{obs}}) = \{a_{1:(k-1)}, b_{1:(k-1)}, d_{1:(k-1)}\}$. To construct $M^{\text{CBS}}_{1:k}(y_{\text{obs}})$, all that remains is to characterize the kth parameters $\{a_k, b_k, d_k\}$. To do this, assume that j_k corresponds with the interval I_k having the form $\{s_k, \ldots, e_k\}$. Within this interval, we form the first $2 \cdot (C(|I_{j_k}| - 1, 2) - 1)$ inequalities of the form,

$$d_k \cdot g^T_{(s_k,a_k,b_k,e_k)} y \geq g^T_{(s_k,r,t,e_k)} y \quad \text{and} \quad d_k \cdot g^T_{(s_k,a_k,b_k,e_k)} y \geq -g^T_{(s_k,r,t,e_k)} y$$

for all $r, t \in \{s_k, \ldots, e_k - 1\}$ where $r < t$ and $r \neq a_k$ and $t \neq b_k$. The remaining inequalities originate from the remaining intervals. For each interval I_ℓ, for $\ell \in \{1, \ldots, 2k-1\} \backslash \{j_k\}$, let I_ℓ have the form $\{s_\ell, \ldots, e_\ell\}$. We form the next $2 \cdot C(|I_\ell| - 1, 2)$ inequalities of the form

$$d_k \cdot g^T_{(s_k, a_k, b_k, e_k)} y \geq g^T_{(s_\ell, r, t, e_\ell)} y \quad \text{and} \quad d_k \cdot g^T_{(s_k, a_k, b_k, e_k)} y \geq -g^T_{(s_\ell, r, t, e_\ell)} y$$

for all $r, t \in \{s_\ell, \ldots, e_\ell - 1\}$ where $r < t$. □

4.B.4 Proof of Proposition 34, (Marginalization)

Proof. For concreteness, we write the proof where W represents additive noise, but the proof generalizes to the setting where W represents random intervals easily. First write $T(y_{\text{obs}}, v)$ as an integral over the joint density of W and Y,

$$\begin{aligned} T(y_{\text{obs}}, v) &= P(v^T Y \geq v^T y_{\text{obs}} | M(Y + W) = M(y_{\text{obs}} + W), \Pi_v^\perp Y = \Pi_v^\perp y_{\text{obs}}) \\ &= \int \mathbf{1}(v^T y \geq v^T y_{\text{obs}}) f_{W,Y|E_1,E_2}(w, y) dw dy. \end{aligned} \tag{4.20}$$

Then the joint density $f_{W,Y|E_1,E_2}(w, y)$ partitions into two components, whose latter component (a probability mass function) can be rewritten using Bayes rule. For convenience, denote $g(w) = \mathbb{P}(E_1 | W = w, E_2)$.

$$\begin{aligned} f_{W,Y|E_1,E_2}(w, y) dy dw &= f_{Y|W=w,E_1,E_2}(y) \cdot f_{W|E_1,E_2}(w)\ dy\ dw \\ &= f_{Y|W=w,E_1,E_2}(y) \cdot \frac{\mathbb{P}(E_1 | W = w, E_2) f_{W|E_2}(w)}{\mathbb{P}(E_1 | E_2)}\ dy\ dw \\ &= f_{Y|W=w,E_1,E_2}(y) \cdot \frac{g(w) f_W(w)}{\int g(w') f_W(w') dw'}\ dy\ dw, \end{aligned}$$

where we used the independence between W and E_2 in the last equality. With this, $T(y_{\text{obs}}, v)$ from (4.20) becomes:

$$T(y_{\text{obs}}, v) = \int \mathbf{1}(v^T y \geq v^T y_{\text{obs}}) \cdot g(w) \cdot \frac{f_{W|E_2}(w)}{\int g(w') f_W(w') dw'} \cdot f_{Y|W=w,E_1,E_2}(y)\ dy\ dw.$$

Now, rearranging, we get:

$$\begin{aligned} T(y_{\text{obs}}, v) &= \int \underbrace{\left[\int \mathbf{1}(v^T y \geq v^T y_{\text{obs}}) \cdot f_{Y|W=w,E_1,E_2}(y) dy \right]}_{T(y_{\text{obs}}, v, w)} \underbrace{\frac{g(w)}{\int g(w') f_W(w') dw'}}_{a(w)} f_W(w) dw \\ &= \int T(y_{\text{obs}}, v, w) a(w)\ f_W(w)\ dw. \end{aligned} \tag{4.21}$$

This proves the first equality in Proposition 34. To show what the weighting factor $a(w)$ equals, observe that by applying Bayes rule to the numerator of $a(w_{\text{obs}})$, and rearranging:

$$a(w) = \frac{g(w)}{\int g(w')f_W(w')\,dw'} = \frac{\mathbb{P}(E_1|E_2, W=w)}{P(E_1|E_2)} = \frac{\mathbb{P}(W=w|E_1,E_2)}{\mathbb{P}(W=w|E_2)}$$
$$= \frac{\mathbb{P}(W=w|E_1,E_2)}{\mathbb{P}(W=w)}.$$

Finally, to show the seocnd equality in Proposition 34, observe that we can also represent $a(w)$ as

$$a(w) = \frac{g(w)}{\mathbb{E}[g(w)]} \tag{4.22}$$

by definition, where the denominator is the expectation taken with respect to the random variable W. Leveraging the geometric theorems of Lee et al. (2016); Tibshirani et al. (2016), it can be shown that

$$g(w) = P\Big(M(Y+W) = M(y_{\text{obs}}+W) \mid \Pi_v^{\perp}Y = \Pi_v^{\perp}y_{\text{obs}}\Big) = \Phi(\mathcal{V}_{\text{up}}/\tau) - \Phi(\mathcal{V}_{\text{lo}}/\tau). \tag{4.23}$$

Also from the same references as well as stated in §4.3.3, we know that

$$T(y_{\text{obs}}, v, w) = \frac{\Phi(\mathcal{V}_{\text{up}}/\tau) - \Phi(v^T y_{\text{obs}}/\tau)}{\Phi(\mathcal{V}_{\text{up}}/\tau) - \Phi(\mathcal{V}_{\text{lo}}/\tau)} \tag{4.24}$$

Putting (4.22), (4.23) and (4.24) together into (4.21), we complete the proof by obtaining

$$T(y_{\text{obs}}, v) = \frac{\int T(y_{\text{obs}}, v, w)g(w)f_W(w)dw}{\int g(w)f_W(w)dw} = \frac{\int \Phi(\mathcal{V}_{\text{up}}/\tau) - \Phi(v^T y_{\text{obs}}/\tau)f_W(w)dw}{\int \Phi(\mathcal{V}_{\text{up}}/\tau) - \Phi(\mathcal{V}_{\text{lo}}/\tau)f_W(w)dw}.$$

□

4.C Additional algorithmic details

4.C.1 Selected model tests, hit-and-run sampling for known σ^2

The following is the hit-and-run sampler to estimate the tail probability of the law of (4.8). This is for the known σ^2 setting, which differs from the setting described in the main text in §4.3.2. This was briefly described in Fithian et al. (2015) but the authors have later implemented it in ways not originally described in the above work to make it more efficient. We do not claim novelty for the following algorithm, but simply state it for completion. The original code can be found the repository `https://github.com/selective-inference`, and we reimplemented it to suite our coding framework and simulation setup.

We specialize our description to test the null hypothesis $H_0 : v_j^T\theta = 0$ against the one-sided alternative $H_1 : v_j^T\theta > 0$. There are some notation to clarify prior to describing the algorithm. Let $v_j \in \mathbb{R}^n$ denote the vector such that

$$v_j^T y = \bar{y}_{I_{j+1}} - \bar{y}_{I_j}.$$

As in §4.3.2, let $A \in \mathbb{R}^{k\times n}$ denote the matrix such that the last k equations in the above display are satisfied if and only if $AY = Ay_{\text{obs}}$. Based on §4.3.1, observe that our goal reduces to sampling from the n-dimensional distribution

$$Y \sim \mathcal{N}(0, \sigma^2 I_n), \quad \text{conditioned on} \quad \Gamma Y \geq 0,\ AY = Ay_{\text{obs}}. \tag{4.25}$$

where I_n is the $n \times n$ identity matrix.

The first stage of the algorithm *removes the nullspace* of A in the following sense. Construct any matrix $B \in \mathbb{R}^{n\times n}$ such that it has full rank and the last k rows are equal to A. Then, consider the following n-dimensional distribution.

$$Y' \sim \mathcal{N}(0, \sigma^2 B^T B), \quad \text{conditioned on} \quad \Gamma B^{-1} Y' \geq 0,\ (Y')_{(n-k+1):n} = Ay_{\text{obs}}. \tag{4.26}$$

Note that $B^{-1}Y'$ has the same law as (4.25). Observe that the above distribution is a conditional Gaussian, meaning we can remove the last conditioning event. Towards that end, let Γ'' denote the first $n-k$ columns of the matrix ΓB^{-1}, and let u'' denote the last k columns of ΓB^{-1} left-multiplying Ay_{obs}. Also, consider the following partitioning of the matrix $B^T B$,

$$\sigma^2 B^T B = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix},$$

where B_{11} is a $(n-k)\times(n-k)$ submatrix, B_{12} is a $(n-k)\times k$ submatrix, and B_{22} is a $k \times k$ submatrix. Then, consider the following $n-k$-dimensional distribution.

$$Y'' \sim \mathcal{N}\Big(B_{12}B_{22}^{-1}(Ay_{\text{obs}}),\ B_{11} - B_{12}B_{22}^{-1}B_{12}^T\Big), \quad \text{conditioned on} \quad \Gamma'' Y'' \geq -u''. \tag{4.27}$$

Note that Y'' has the same law as the first $n-k$ coordinates of (4.26).

The next stage of the algorithm *whitens* the above distribution so its covariance is the identity. Let μ'' and Σ'' denote the mean and variance of the unconditional form of the above distribution (4.27). Let Θ be the matrix such that $\Theta\Sigma''\Theta^T = I_n$. This must exist since Σ'' is positive definite. Consider the following $n-k$ dimensional distribution,

$$Z \sim \mathcal{N}(0, I_n), \quad \text{conditioned on} \quad \Gamma''\Theta^{-1}Z \geq -u'' - \Gamma''\mu''. \tag{4.28}$$

Note that $\Theta^{-1}Z + \mu''$ has the same law as (4.27). Hence, we have constructed linear mapping F and G between (4.25) and (4.28) such that $F(Y) \overset{d}{=} Z$, and $G(Z) \overset{d}{=} Y$.

Algorithm 4: MCMC hit-and-run algorithm for selected model test with known σ^2

Choose a number M of iterations.
Set $z^{(0)} = F(y_{\text{obs}})$, as described in the text.
Generate p unit directions $g_1, \ldots, g_p$, each vector of length n.
Compute $U = \Gamma''\Theta^{-1}z^{(0)} + u'' + \Gamma''\mu''$, which represents the "slack" of each constraint.
Compute the p vectors, $\rho_i = \Gamma''\Theta^{-1}g_i$ for $i \in \{1, \ldots, p\}$.
for $m \in \{1, \ldots, M\}$ **do**
  Select an index i uniformly from 1 to p.
  Compute the truncation bounds

$$\mathcal{V}_{\text{lo}} = g_i^T z^{(m-1)} - \min_{j:(\rho_i)_j > 0} U_j/(\rho_i)_j, \quad \text{and} \quad \mathcal{V}_{\text{up}} = g_i^T z^{(m-1)} - \max_{j:(\rho_i)_j < 0} U_j/(\rho_i)_j.$$

  Sample $\alpha^{(m)}$ from a Gaussian with mean $g_i^T z^{(m-1)}$ and variance 1, truncated to lie between $\mathcal{V}_{\text{lo}}$ and $\mathcal{V}_{\text{up}}$.
  Form the next sample

$$z^{(m)} = z^{(m-1)} + \alpha^{(m)} g_i, \quad \text{and} \quad y^{(m)} = G(z^{(m)}).$$

  Update the slack variable,

$$U \leftarrow U + \alpha^{(m)}\rho_i.$$

end
Return the approximate for the tail probability of (4.9), $\sum_{m=1}^{M} \mathbf{1}[v^T y^{(m)} \geq v^T y_{\text{obs}}]/M$.

In order to set up a hit-and-run sampler, generate p unit vectors $g_1, \ldots, g_p$. (The choice of p is arbitrary, and the specific method of generating these p vectors is also arbitrary.) Our hit-and-run sampler with move in the linear directions dictated by $g_1, \ldots, g_p$. We are now ready to describe the hit-and-run sampler in Algorithm 4, which leverages many of the same calculations in (4.11) and (4.12). The similarity arises since $\Pi_{g_i}^{\perp} Z = \Pi_{g_i}^{\perp}(Z + g_i)$ by definition of projection.

The computational efficiency of the above algorithm comes from the fact that little multiplication needs to be done with the polyhedron matrix $\Gamma''\Theta^{-1}$, a potentially huge matrix. U and $\rho_1, \ldots, \rho_p$, each vectors of the same length, carry all the information needed about polyhedron throughout the entire procedure of generating M samples.

4.C.2 Marginalization over WBS intervals

Below is the pseudo-code for the marginalization over WBS intervals, described in §4.3.3.

Algorithm 5: Marginalizing over random intervals

```
Choose a number T of trials.
for t ∈ {1, ..., T} do
    Sample the non-maximizing intervals w_ℓ = (s_ℓ, ..., e_ℓ) for ℓ ∈ {1, ..., B}\{ĵ_{1:k}}
      where s_ℓ, e_ℓ are uniformly drawn from 1 to n and s_ℓ < e_ℓ.
    Check to see that {ĵ_{1:k}} are still the indices of the maximizing intervals. If not,
      return to the previous step.
    Compute k(w_t) and g(w_t).
end
Return the approximate for the tail probability (4.15),
```

$$\frac{\sum_{t=1}^{T} k(w_t)}{\sum_{t=1}^{T} g(w_t)}.$$

4.C.3 Additional simulation results

4.C.4 Type-I error control verification

We examine all our statistical inferences under the global null where $\theta = 0$ to demonstrate their validity – uniformity of null p-values, or type I error control. Specifically, any simulations from the no-signal regime $\delta = 0$ from the middle mutation (4.16) can be used. When there is no signal, the null scenario $v^T\theta = 0$ is always true so we expect all p-value to be uniformly distributed between 0 and 1. We verify this expected behavior in Figure 4.6. We notice that the methods that require MCMC (marginalized saturated and selected model tests) requires more trials to converge towards the uniform distribution compared to their counterparts that have exact calculations.

4.C.5 Comparison with sample splitting

Sample splitting is another valid inference technique. After splitting the dataset in half based on even and odd indices, we run a changepoint algorithm on one dataset and conduct classical one-sided t-test on the other. This is the most comparable test, as it does not assume σ^2 is known and conducts a one-sided test of the null $H_0 : v^T\theta = 0$. Instead of ± 2 slack used for calculating detection in selective inference detection (dotted and dashed lines), ± 1 was used for sample splitting inference (solid line). The loss in detection accuracy in the middle panel of Figure 4.7 shows the downside of halving data size for detection. Unconditional power for marginalized saturated model tests and selected model tests are noticeably higher than the other two.

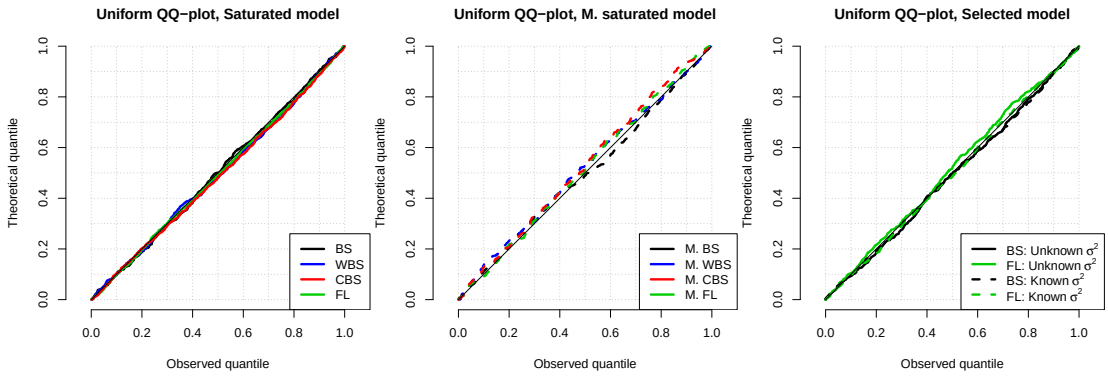


Figure 4.6: All plots showing the p-values of various statistical inferences under the global null, with colors of lines given according to Figure 4.2 and 4.3. (Left): Saturated model tests, specifically BS (black), WBS (blue), CBS (red) and FL (green). (Middle): Marginalized variants of the left plot. (Right): Selected model tests, specifically BS (black) and FL (green), either with unknown σ^2 (solid) or known σ^2 (dashed).

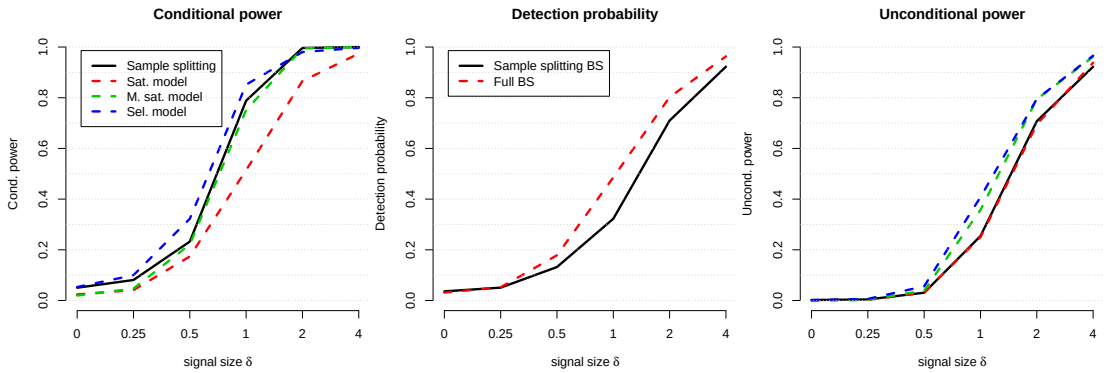


Figure 4.7: Setup similar to Figure 4.2 but comparing sample splitting (black solid), plain saturated model test (red dashed), additive noise marginalized saturated model test (green dashed), and selected model test with unknown σ^2 (blue dashed), all using a 2-step binary segmentation. (Middle): Detection probability for the binary segmentation applied on the sample split dataset (black solid) or the full dataset (red dashed). (Right): Unconditional power, computed by multiplying the conditional power curve and its relevant detection probability curve.

4.C.6 Power comparison using unique detection

Fused lasso was appeared to have a large drop in power compared to segmentation algorithms. In addition to these three measures shown in §4.5, for multiple changepoint problems like middle mutations it is useful to measure performance using an alternative measure of detection called unique detection. This is useful because some algorithms – mainly fused lasso, but to also binary segmentation to some extent, primarily in later steps – admit "clumps" of nearby points. If this clumped detection pattern occurs in early steps, the algorithm requires more steps than others to fully admit the correct changepoints. In this case, detection alone is not an adequate metric, and unique detection can be used in place.

$$\text{Unique detection probability} = \frac{\#\text{changepoints which were approximately detected}}{\#\text{number of true changepoints.}} \tag{4.29}$$

In plain words, unique detection is measuring how many of the true changepoint locations have been approximately recovered.

We present a simple case study. In addition to a 2-step fused lasso, imagine using a 3-step fused lasso, but with post-processing. For post-processing, declutter by centroid clustering with maximum distance of 2, and test the $k_0 < 3$ changepoints, pitting the resulting segment test p-values against $0.05/k_0$. A 2-step fused lasso's detection does not reach 1 even at high signals ($\delta = 4$) because of the aforementioned clumped detection behavior. The resulting segment tests are also not powerful, since the segment test contrast vectors consist of left and right segments which do not closely resemble true underlying piecewise constant segments in the data. However, when detection is replaced with unique detection, two things are noticeable. First, decluttered lasso's detection performance is noticeably improved when going from 2 to 3 steps. Also, when unconditional power is calculated using unique detection, binary segmentation does not have as large of an advantage over the the several variants of fused lasso. This is shown in Figure 4.8. We see from the right figure (compared to the left) that the a "decluttered" version of 2- or 3-step fused lasso has much closer unconditional power to binary segmentation.

4.C.7 Power comparison with different mean shape

The synthetic mean discussed here consists of a single upward changepoint piece-wise constant mean, as shown in (4.30) and Figure 4.9 (right). This is chosen to be another realistic example of the mutation phenomenon as observed in array CGH datasets from Snijders et al. (2001), in addition to the case shown in the main text. We focus on the *duplication* mutation scenario, but the results apply similarly to deletions. As before, the sample size $n = 200$ was chosen to be in the scale of the data length in a typical array CGH dataset in a single chromosome. For saturated model tests, WBS no longer outperforms binary segmentation in power. This is expected since there is only a single changepoint not accompanied by opposing-direction changepoints.

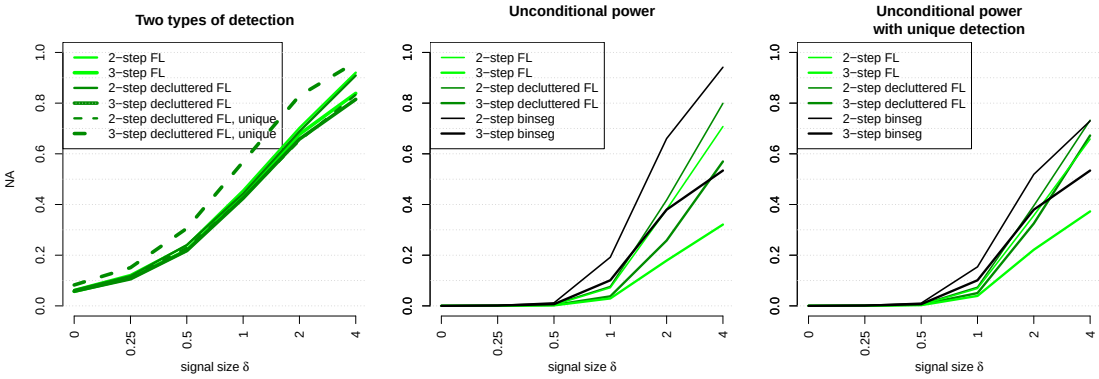


Figure 4.8: (Left): Various detections for FL, either using 2 or 3 steps, and either using decluttering or not. (Middle): The unconditional power of various segmentation algorithms. (Right): The unconditional power, but defined as the conditional power multiplied by the unique detection probability.

$$\textbf{Edge mutation:} \quad y_i \sim \mathcal{N}(\theta_i, 1), \;\; \theta_i = \begin{cases} \delta & \text{if } 161 \leq i \leq 200 \\ 0 & \text{if otherwise} \end{cases} \tag{4.30}$$

4.C.8 Sample splitting (continued)

The results in Figure 4.7 were based on approximate detection where, for methods used on the entire dataset of length n, we defined a detection event as estimating ± 2 of the true changepoint locations. For sample splitting, this was defined as estimate ± 1 of the true changepoint location based on half the dataset. This choice of approximate detection is somewhat arbitrary, and it is informative to see if the results would change if we considered only exact detection. We can see from Figure 4.11 that randomized TG p-values have comparable power with sample splitting inferences, among tests that are regarding exactly the right changepoints.

4.D Model size selection using information criteria

Throughout the paper we assume that the number of algorithm steps k is fixed. Hyun et al. (2018a) introduces a stopping rule based on information criteria (IC) which can be characterized as a polyhedral selection event. The IC for the sequence of models $M_{1:\ell}, \ell = 1, \ldots, n-1$ is

$$J(M_{1:\ell}) = \|y - \widehat{y}_{M_{1:\ell}(y)}\|_2^2 + p\big(M_{1:\ell}(y)\big). \tag{4.31}$$

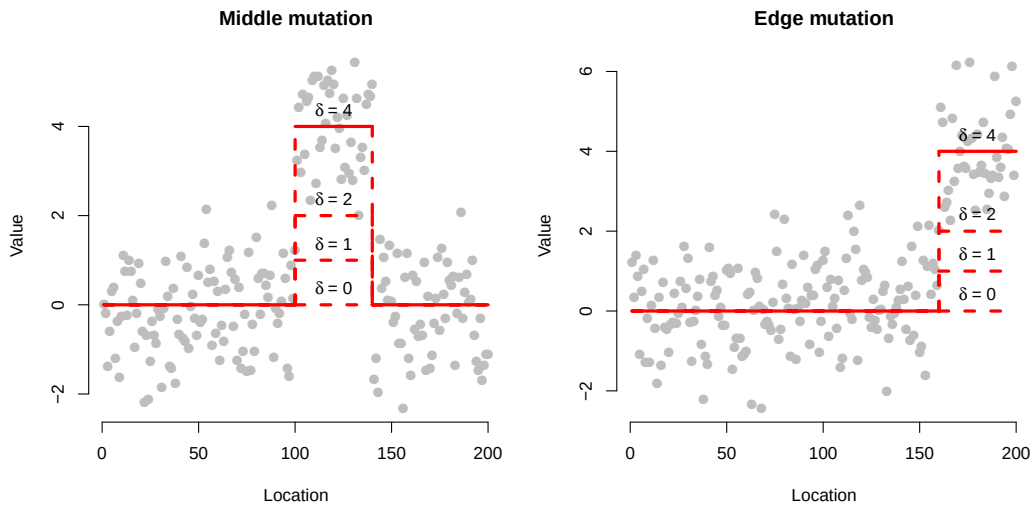


Figure 4.9: (Left) Example of simulated Gaussian data for middle mutation as defined in (4.16) with $\delta = 4$, with data length $n = 200$ and noise level $\sigma = 1$. The possible mean vectors θ for $\delta = 0, 1, 2$ are also shown. (Right) Analogous to the left figure, but representing edge mutations defined in (4.30).

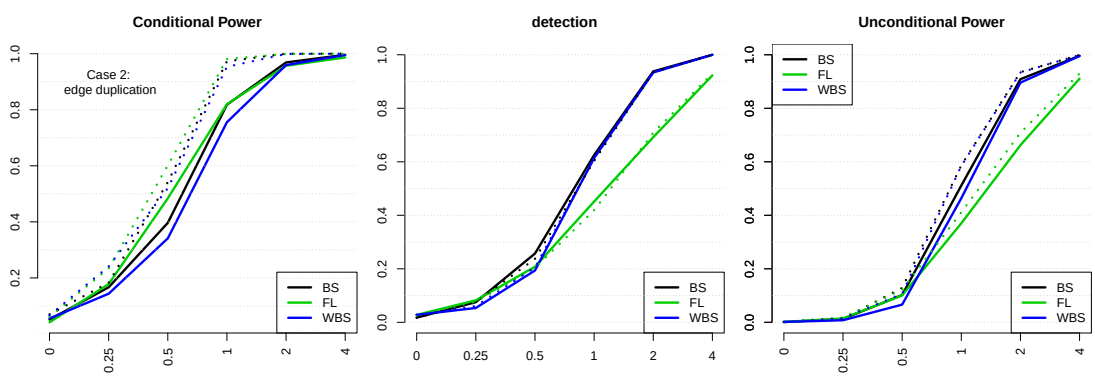


Figure 4.10: Same setup as Figure 4.2 but for edge-mutation data.

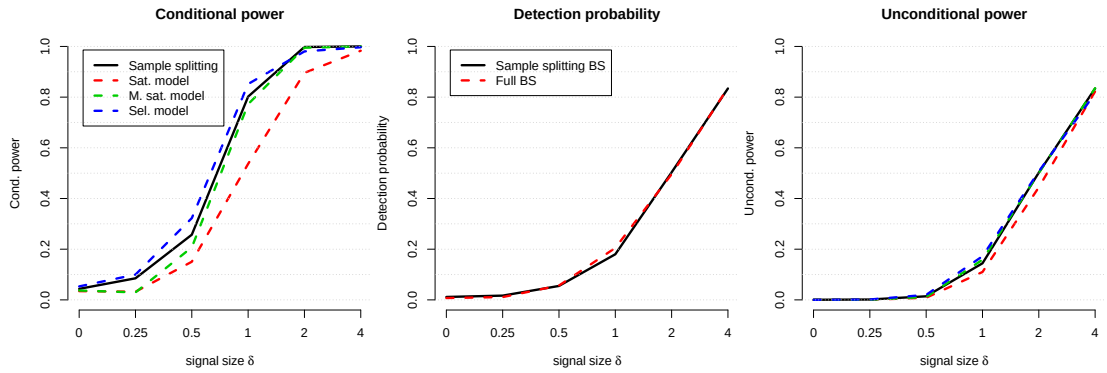


Figure 4.11: The same setup as in Figure 4.7 but with exact detection.

We omit the dependency on y when obvious. We use the BIC complexity penalty $p(M_k) = \sigma^2 \cdot k \cdot \log(n)$ for this paper. Also define $S_\ell(y) = \text{sign}\left(J(M_{1:\ell}) - J(M_{1:(\ell-1)})\right)$ to be the sign of the difference in IC between step $\ell - 1$ and ℓ. This is a $+1$ for a rise and -1 for a decline. A data-dependent stopping rule $\widehat{k}$ is defined as

$$\widehat{k}(y) = \min\{k : S_k(y) = S_{k+1}(y) = \ldots = S_{k+q}(y) = 1\} \tag{4.32}$$

which is a local minimization of IC, defined as the first time q consecutive rises occur. As discussed in Hyun et al. (2018a), $q = 2$ is a reasonable choice for the changepoint detection. To carry out valid selective inference, we condition on the selection event $\mathbf{1}[S_{1:(k+q)}(y) = S_{1:(k+q)}(y_{\text{obs}})]$, which is enough to determine $\widehat{k}$. A k-step model for k chosen by (4.32) can be understood to be $M_{1:\widehat{k}}(Y) = M_{1:k}(y_{\text{obs}})$. The corresponding selection event $P_{M_{1:\widehat{k}}}$ is with the additional halfspaces, as outlined in Hyun et al. (2018a). Simulations in Figure 4.12 show that introducing IC stopping is valid, by controlled type-I error, but comes at the cost of considerable power loss.

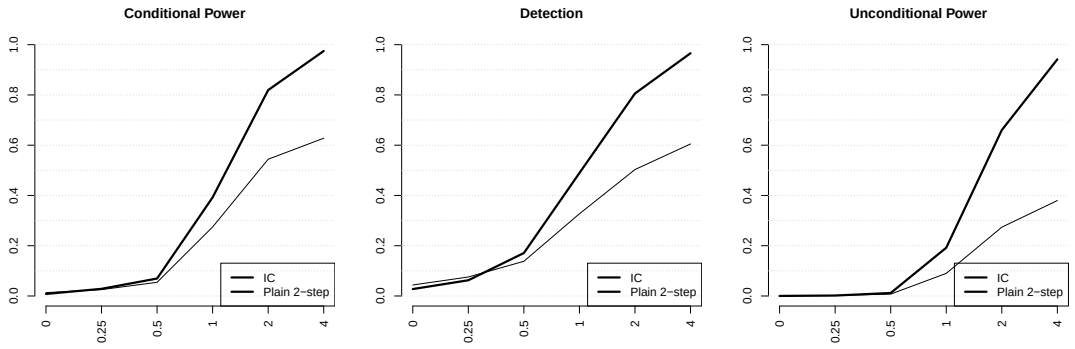


Figure 4.12: Similar setup as Figure 4.2. In the middle-mutation data example from (4.16). IC-stopped binary segmentation inference (bold line) is compared to a fixed 2-step binary segmentation inferences (thin line). We can see that the power and detection are considerably lower. The average number of steps taken per each δ on x-axis ticks are $1.34, 1.86, 3.02, 3.64, 3.77, 3.72$, respectively.

Five

Modeling heterogeneity - Exponential-family embedding

Paper summary: Single-cell RNA-seq data enable scientists to study cell developmental trajectories, but the statistical properties of these methods are not well developed. In this article, we study the statistical properties of embedding each cell into a lower dimensional space, an important component of cell trajectory estimation methods. Specifically, we develop *eSVD* (exponential-family SVD), which estimates an embedding for each cell with respect to a hierarchical model where the inner product between latent vectors is the natural parameter of an exponential family random variable. Our estimation procedure uses an alternating minimization approach, and we prove its the identifiability conditions and convergence rate, in line with other theoretical works in the nonconvex optimization literature. Our method is similar to other matrix factorization methods, but we adapt its underlying algorithm and statistical theory to be more amendable for single-cell analyses.

We apply eSVD via Gaussian distributions where the standard deviations are proportional to the means to analyze a single-cell dataset of oligodendrocytes in mouse brains (Marques et al., 2016). While previous results are not able to distinguish the lineages among the mature oligodendrocyte cell types, our diagnostics and results demonstrate there are two major developmental lineages that diverge at mature oligodendrocytes.

The work in this chapter was done jointly with Jing Lei and Kathryn Roeder, and has been accepted to JASA Applications and Case Studies under the title, "Exponential-family embedding with application to cell developmental trajectories for single-cell RNA-seq data."

5.1 Introduction

Single-cell RNA-seq data give scientists an unprecedented opportunity to analyze the dynamics among individual cells based on their gene expressions, but many analysis require first embedding each cell in a lower-dimensional space in order to make downstream methods more statistically or computationally tractable. For example, this low-dimensional embedding can be used to visualize high-dimensional data, to control for ancestry across different subpopulations, to cluster cells into cell-types, to impute values deemed as dropouts, or to estimate trajectories to understand how cells develop over time. Typically, these embeddings are computed from an n by p gene expression matrix, where each of the n rows represent a different cell and each of the p columns represent a different gene. However, the most commonly-used embedding is based on the singular value decomposition (SVD), which implicitly assumes that distribution of each entry is a Gaussian random variable where the variance is fixed regardless of its mean. However, this assumption is often violated in sequencing data where the variance of each cell's observations can vary dramatically with their mean expression level (Love et al., 2014; Hicks et al., 2017). This consideration motivates us to develop the *eSVD* (exponential-family SVD), a generalization of SVD, to embed the cells into a lower-dimensional space with respect to any one-parameter exponential family distribution, allowing the scientist to have much broader modeling flexibility. In this article, we formalize this statistical idea and design a non-convex estimator based on alternating minimization to estimate this low-dimensional embedding, provide theory to prove its consistency and convergence properties, and apply it to single-cell RNA-seq data to obtain better downstream analysis results.

To demonstrate the shortcomings of SVD throughout our paper, we focus on analyzing oligodendrocytes – cells that enable rapid transmission of signals by producing myelin and provide metabolic support to neurons in the central nervous system. These cells are intriguing to study due to their constant development throughout a subject's lifetime, unlike most other cells that stop development during the adult years of an organism (Menn et al., 2006). As mentioned in Marques et al. (2016), understanding how oligodendrocytes develop can lead to new insights into the causes for myelin disorders such as multiple sclerosis. We discuss the oligodendrocyte dataset and present a preliminary analysis in §5.2, where we provide various diagnostics demonstrating the SVD embedding's lack of fit. To better understand this phenomenon, we review the hierarchical model that SVD implicitly assumes in §5.3. Specifically, suppose a hierarchical model where each cell and each gene has its own low-dimensional latent random vector. In the language of exponential family distributions, this model assumes that the cell's expression of a particular gene is a one-parameter exponential family random variable whose relevant natural parameter is the inner product of the two corresponding latent vectors. As mentioned above, SVD assumes specifically a Gaussian distribution with constant variance. However, as we will review later, there is a rich line of

work that extend hierarchical models of this type to analyze single-cell data (Pierson and Yau, 2015; Townes et al., 2017; Durif et al., 2017; Risso et al., 2018), of which eSVD is a continuation of.

eSVD builds upon algorithmic ideas developed in the field of matrix factorization, a field which studies how to estimate a fixed but unknown low-rank matrix of natural parameters, a task similar to the one we are pursuing. Specifically, eSVD uses alternating minimization, a popular and computationally efficient approach to solve the nonconvex optimization problem at hand, described in detail in §5.4. However, despite the vast amount of algorithmic developments in matrix factorization, only a small subset of methods are amenable for theoretical investigations of their statistical properties such as convergence rates. This is a common criticism of alternating-minimization based methods, as stated in Liu et al. (2018b). To address these critiques, we design eSVD different from other methods in the literature to improve upon its usability while enabling us to study its statistical properties detailed in §5.5. To ensure that these theoretical ideas empirically improve the analysis of single-cell data, we compare eSVD to more recent embedding methods developed in the biostatistics community in §5.6 using synthetic data.

We apply eSVD to improve our former analysis of oligodendrocytes in §5.7, where we show that our new embedding helps to estimate cell developmental trajectories that match with the latest scientific findings. These trajectories, also called lineages, explain the heterogeneity among the oligodendrocytes by describing the smooth transition of gene expression among individual cells along a continuum, reflecting the cells' gradual transcriptional changes during development (Trapnell et al., 2014). Although early research suggest oligodendrocytes develop along a single trajectory (Kessaris et al., 2006), recent works suggest that oligodendrocytes could potentially branch out into various mature types (Marques et al., 2016; van Bruggen et al., 2017; Marques et al., 2018). Our improved analysis match these findings – we show the eSVD embedding estimates two distinct trajectories. We develop a novel visualization tool to show our developmental trajectory findings, and conclude in §5.8 with practical extensions and theoretical questions left open for future work. While we focus on using the eSVD embedding to estimate cell developmental trajectories in our paper, we emphasize that this embedding can be used for other applications highlighted earlier in this section, and provide additional analyses on other single-cell datasets in the appendix.

5.2 Preliminary analysis

We analyze a dataset of oligodendrocytes from mice brains collected by Marques et al. (2016) as a prototypical example to demonstrate shortcomings of the SVD embedding when applied to single-cell data. This dataset, henceforth called the Marques dataset, contains the gene expression of 5,069 oligodendrocytes categorized into six major cell types that are manually labeled based on marker genes, shown in Figure 5.1. The 5,069 cells in the dataset are

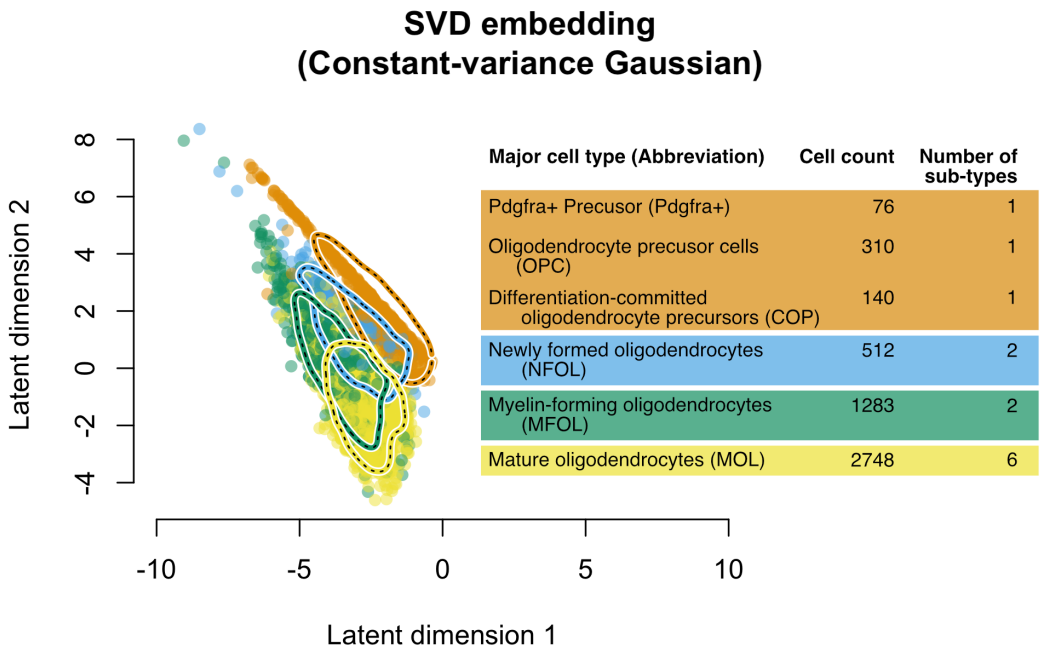


Figure 5.1: SVD embedding of the oligodendrocytes from Marques et al. (2016) after preprocessing, shown alongside a table summarizing the cell types. The six major cell types are listed in the table with the number of cells in each type, along with how they are differentiated into the thirteen different cell sub-types. The rows are organized from the "youngest" cell types to "most mature" cell types from top to bottom. The first three major cell types are colored orange. while the latter three are colored blue, green and yellow respectively. The first two latent dimensions are shown on the left, along with contours of the estimated densities to visualize high-density regions (one for each color of points).

clustered in Marques et al. (2016) into thirteen cell sub-types using a biclustering algorithm (Zeisel et al., 2015). These cell sub-types were then grouped into six major cell types using a biclustering algorithm (Zeisel et al., 2015), which were later manually labeled based on cell-type specific marker genes (Zhang et al., 2014). The full details of our preprocessing details can be found in Appendix 5.B, including our procedure for selecting informative genes for all the remaining analysis in this paper. The goal of this preliminary analysis is to previous various diagnostics and modeling concerns we wish to remedy prior downstream analysis in the remaining sections of the paper.

We review the SVD embedding, as it provides motivation for eSVD in the next section. Let $A \in \mathbb{R}^{n \times p}$ represent the observed single-cell data matrix with rank m, where n is the

number of cells and p is the number of genes. Here, A is a matrix of non-negative values, where entry A_{ij} measures how many instances of genetic material for gene j is observed for cell i after appropriate pre-processing. Let the SVD of A be denoted as $\widehat{U}\widehat{D}\widehat{V}^\top$ where $\widehat{U} \in \mathbb{R}^{n\times m}$ and $\widehat{V} \in \mathbb{R}^{p\times m}$ are both orthonormal matrices and $\widehat{D} \in \mathbb{R}^{m\times m}$ is a diagonal matrix. For a given latent dimension $k \leq m$, the SVD embedding for each cell $i \in \{1, \ldots, n\}$ (denoted as $\widehat{X}_i \in \mathbb{R}^k$) and each gene $j \in \{1, \ldots, p\}$ (denoted as $\widehat{Y}_j \in \mathbb{R}^k$) becomes

$$\widehat{X}_i = \left(\frac{n}{p}\right)^{1/4} \cdot \left(\sqrt{\widehat{D}_{1,1}} \cdot \widehat{U}_{i,1}, \ldots, \sqrt{\widehat{D}_{k,k}} \cdot \widehat{U}_{i,k}\right), \quad i = 1, \ldots, n \tag{5.1}$$

$$\widehat{Y}_j = \left(\frac{p}{n}\right)^{1/4} \cdot \left(\sqrt{\widehat{D}_{1,1}} \cdot \widehat{V}_{j,1}, \ldots, \sqrt{\widehat{D}_{k,k}} \cdot \widehat{V}_{j,k}\right), \quad j = 1, \ldots, p. \tag{5.2}$$

The first two dimensions of such an embedding is shown in Figure 5.1. Later in this article, we will show that this embedding implicitly assumes a constant variance Gaussian distribution in §5.3, and show that this particular formulation for identifiability concerns that will be discussed in detail in §5.4.

Now, we show that while the SVD embedding (or its equivalent reparameterizations) are commonly used in the literature, it does not model the data well. First, we visualize the quality of fit of the SVD embedding by purposefully omitting a small subset of randomly selected entries in A and estimating the embedding as a missing data problem. We can then assess the quality of fit of the embedding by comparing the values of these purposefully omitted entries in A to their predicted values. Figure 5.2 demonstrates this diagnostic, where the left plot shows the observed values in A that were not omitted (i.e., the "training set") verses their respective predicted values, while the right plot shows the observed values that were purposefully omitted (i.e., the "testing set") verses their respective predicted values. We see that while the fitted values in the training set closely predict the observed values, the fitted values for the testing set are wildly off in comparison. Furthermore, we see that the variance among the testing set does not seem to constant, as the prediction error grows as the predicted values grows. This missing-value diagnostic is commonly used as both a goodness of fit heuristic as well as a model selection tool in work such as Li et al. (2016), and we will return to it in detail in §5.4. Next, to further investigate if this constant-variance is suitable for modeling the oligodendrocytes, we plot the standard deviation verses mean expression for each gene, as well as the weighted average expression of each of the six cell types in Figure 5.3. Both plots suggest that the variance in gene expression increases with its mean. Combined, all the empirical diagnostics inspire us to develop a better embedding that have desirable statistical properties such as identifiability and consistency. In the next section, we review the optimization problem that the SVD embedding solves and see how it can be extended to one-parameter exponential families more generally, which will motivate our method, the eSVD.

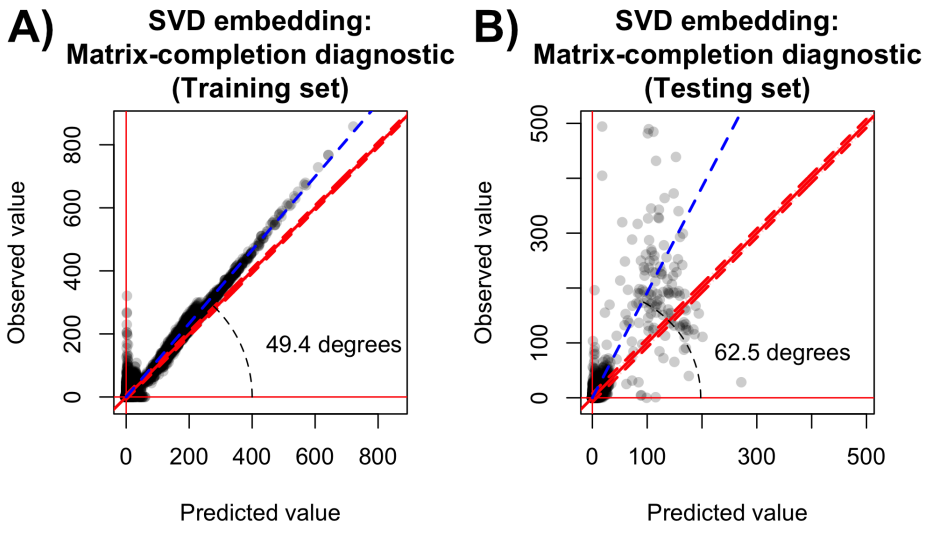


Figure 5.2: Diagnostic based on matrix completion to assess the fit using SVD embedding, fit using softImpute (Mazumder et al., 2010). The red diagonal band is centered around the identity function (ideal mean function) and marks the 10th to 90th quantiles of the constant-variance Gaussian model (based on the empirical variance) for different values of the predicted mean. The blue dotted line represents the principal angle between the observed values and their predicted value counterparts, where we mark its divergence from the identity function's 45°. More details of this diagnostic is discussed in §5.4, while details of the fitting process using softImpute can be found in Appendix 5.B. (A) Diagnostic plot showing the observed values that were not omitted (i.e., the "training set") verses their respective predicted values. (B) Diagnostic plot showing the observed values that were purposefully omitted (i.e., the "testing set") verses their respective predicted values.

5.3 Statistical model and background

In this section, we explain the latent hierarchical model that we investigate in this article, and its relation to other works in the matrix factorization and single-cell biostatistics community.

5.3.1 Statistical model

In this article, we model the entries of the single-cell RNA-seq dataset $A \in \mathbb{R}^{n \times p}$ as conditionally independent random variables drawn from a random dot product model – a hierarchical model where each of the n cells and each of the p genes has its own corresponding low-dimensional latent vector. Specifically, for an appropriate one-parameter exponential family distribution F parameterized by its natural parameter θ_{ij}, we impose the random

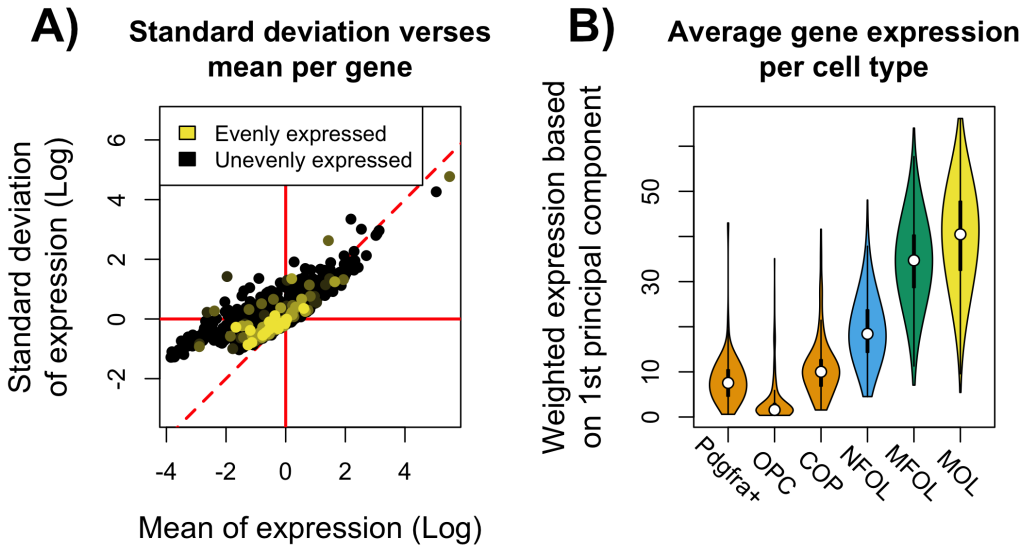


Figure 5.3: (A) The standard deviation of the expression verses the mean expression across all the cells (on the logarithm scale), where each point represents one of the 983 genes in the preprocessed single-cell RNA-seq dataset. The color of each point depends on how evenly the gene is expressed among each of the six oligodendrocyte cell types show in Figure 5.1. The solid red horizontal and vertical lines and the dashed red line denoting the line $y = x$ are for visual reference. (B) Violin plot of the average expression of the genes reweighted according to the first principal component amomg the six oligodendrocyte cell types. The first three cell types are colored orange while the last three cell types are each colored a different color. More details about how the data was preprocessed and how both these plots were made are in Appendix 5.B.

dot product model used by other biostatistic works,

$$\begin{aligned} X_1, \ldots, X_n &\overset{i.i.d.}{\sim} G, \quad \text{and} \quad Y_1, \ldots, Y_p \overset{i.i.d.}{\sim} H, \\ A_{ij} &\sim F\big(\theta_{ij} = X_i^\top Y_j\big), \quad \text{for } (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, p\}, \end{aligned} \tag{5.3}$$

where G and H represent two latent k-dimensional distributions. We assume all the variables X_i's and Y_j's are jointly independent, and A_{ij}'s are independent conditioned on X_i's and Y_j's. Let the density of the exponential family distribution F be

$$p(A_{ij} \mid \theta_{ij}) = h(A_{ij}) \exp\big(\eta(A_{ij})^\top T(\theta_{ij}) - g(\theta_{ij})\big), \tag{5.4}$$

where $g(\cdot)$ is a known log-partition function for F with a domain $\mathcal{R}$, $\eta(\cdot)$ is a known natural parameter function, and $T(\cdot)$ is a known sufficient statistic function. For notational convenience, we denote $X \in \mathbb{R}^{n \times k}$ and $Y \in \mathbb{R}^{p \times k}$ as the matrices that collect all the latent

vectors $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_p$ row-wise, and denote $\Theta = XY^\top \in \mathbb{R}^{n \times p}$ as the natural parameter matrix that collects all elements θ_{ij}. For the above model to be valid, we require the following assumption.

Assumption 1 (Bounded inner product)**.** *Let $\mathcal{R}$ denote the domain of the natural parameters for the distribution F. Assume that for any $X_i \sim G$ and $Y_j \sim H$,*

$$\mathbb{P}(X_i^\top Y_j \in \mathcal{R}) = 1, \quad \textit{almost surely.}$$

The most common choice is setting F to be the Gaussian distribution with a constant variance. Other works such as Durif et al. (2017) and Risso et al. (2018) consider the Poisson and Negative Binomial distribution specifically. While we design eSVD to work for any one-parameter exponential family distribution, we highlight the *curved Gaussian* distribution for F, which we will use to analyze the oligodendrocytes later in this article.[1] That is, for a fixed parameter $\tau > 0$, consider the density,

$$p(A_{ij} \mid \theta_{ij}) = \frac{\tau \exp(-\tau^2/2)}{\sqrt{2\pi}} \exp\Big(\begin{bmatrix} -\tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{bmatrix}^\top \begin{bmatrix} \theta_{ij} \\ \theta_{ij}^2 \end{bmatrix} + \log(\theta_{ij}) \Big). \tag{5.5}$$

To convert the natural parameter into its canonical parameter, it can be shown that the above distribution has mean $\mu_{ij} = 1/\theta_{ij}$. This implies that this distribution is essentially $A_{ij} \sim N(\mu_{ij}, \mu_{ij}^2/\tau^2)$, where the standard deviation is proportional to the mean. Here, $\mathcal{R} = \mathbb{R}_+$, the positive half-line. This curved exponential family distribution is relevant since in practice, if $\tau \geq 2$, this distribution would reflect the phenomenon that genes with larger expression also exhibit larger variance, but yet most of the distribution's mass is still positive.

Given the above model, our goal is to estimate the random vectors $X_1, \ldots, X_n$ since these latent vectors encode the cell developmental trajectories.

5.3.2 Matrix factorization

One of the focuses of the matrix factorization field is to minimize the following negative log-likelihood loss function for an exponential family distribution over all rank-k matrices X and Y shown in (5.4),

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \Big[g(X_i^\top Y_j) - \eta(A_{ij})^\top T(X_i^\top Y_j) \Big], \tag{5.6}$$

constrained to $X_i^\top Y_j \in \mathcal{R}$ for all pairs (i, j). The above loss function is nonconvex, but if F is the the constant-variance Gaussian distribution, the above display is proportional to

$$\frac{1}{np} \sum_{(i,j)} (A_{ij} - X_i^\top Y_j)^2,$$

[1] We call it a curved Gaussian distribution since this distribution is a curved exponential family distribution.

which is specifically what SVD minimizes. This particular hierarchical model is convenient to use since SVD provides a closed-form solution to the corresponding nonconvex optimization problem (Maezika, 2016). Specifically, the SVD embedding would be the vectors displayed in (5.1) and (5.2). For any other exponential family distribution, such as the curved Gaussian distribution mentioned in (5.5), we need to develop more sophisticated estimators to optimize this nonconvex loss.

To the best of our knowledge, the first statistical results for estimators that minimized a loss similar to (5.6) for general exponential family distributions come from Gunasekar et al. (2014) and Lafond (2015). There, the authors minimize the loss over the natural parameter matrix Θ and add a trace penalization term to encourage the estimate to be low-rank. While this formulation yields a convex optimization problem, it computationally requires solving a semi-definite program which hinders the analysis of large datasets. This consideration has motivated researchers to investigate the statistical properties of estimators that minimize the nonconvex function (5.6) directly. Specifically, *alternating minimization* is a suitable candidate to minimize (5.6), where each iteration alternates between optimizing either one of two low-rank matrices X and Y while treating the other fixed. This algorithmic strategy pre-dates the convex relaxation approach (see Collins et al. (2002),Jain et al. (2013), Udell et al. (2016) and the references within), but the statistical properties of such estimators have only recently been characterized rigorously. These theoretical results are summarized in Chi et al. (2019) and the references within. To accommodate the restriction in Assumption 1, works such as Yu et al. (2020) and Wang et al. (2016) adapt the theoretical framework to study alternating projected gradient descent instead. However, all the aforementioned theoretical results do not directly apply the random dot product model (5.3) due to the additional source of randomness induced by the hierarchical structure. Our theory will account for this additional source of randomness. Additional discussion contrasting eSVD with other estimators in the literature is deferred to Appendix 5.C.

5.3.3 Relation to other work in biostatistics

A distinguishing feature of the random dot product model discussed in (5.3) is that the latent vectors X_i and Y_j are all treated as random. Random dot product models of this form are commonly used in the biostatistics literature to model single-cell RNA-seq data, and these models often include other random effects that influence A_{ij}. For example, most of the methods such as pCMF (Durif et al., 2017) allow researchers to incorporate *dropout* into the model – a characteristic of single-cell data where a substantial fraction of the gene expression for any cell are recorded as exactly 0 due to low amounts of RNA in the cell (Kharchenko et al., 2014). Other methods such as ZINB-WaVE (Risso et al., 2018) go further and allow covariate information such as gene length and cell size. Most recently, Lopez et al. (2018) use deep autoencoders to estimate the latent embedding.

A common criticism of these more sophisticated models focuses on their lack of theoretical

analysis. Concerns such as identifiability are typically not addressed rigorously, which cause ambiguity on understanding the performance of the estimators of such models. Our analysis of eSVD in the random dot product model shown in (5.3) is able to overcome this issue by drawing upon connections to the network literature. Specifically, our random dot product model is similar to those used in latent position random graphs studied in the network literature (see Hoff et al. (2002); Athreya et al. (2017); Nielsen and Witten (2018) and the references within). Hence, we draw inspiration from Lei (2018) on how to address these identifiability concerns and develop proof techniques in this article.

While all the aforementioned works use an embedding where $X_i^\top Y_j$ controls the distribution of A_{ij} loosely speaking, there are other embeddings often used in single-cell analyses. For example, certain cell trajectory methods first embed all the cells via independent component analysis (ICA) (Trapnell et al., 2014; Street et al., 2018). There is not much statistical theory for such embeddings however. UMAP (Becht et al., 2019) is another popular embedding method, but this method is commonly used purely for visualization as the estimated latent embedding does not reflect any statistical quantity. Also, other work such as Liu et al. (2018b) and Zhang et al. (2018) estimate a low-rank covariance matrix for various exponential-family distributions, as opposed to our goal to estimate a low-rank matrix of natural parameters.

5.4 Method: eSVD (Exponential-family SVD)

We describe eSVD in this section, which is designed to be a general framework to minimize (5.6) for any choice of a one-parameter exponential family distribution F. To keep its presentation clear, we describe some of the more nuanced implementation details in Appendix 5.C. We also describe an important diagnostic which provides the user a tool to decide which choice of F best suits the data at hand. This diagnostic can be also used as a tuning heuristic if the choice of exponential family distribution relies on choosing a tuning parameter such as τ in the curved Gaussian model (5.5).

5.4.1 eSVD

We lay down some notation needed to explain our method. Denoting a generic matrix and its SVD by $X = UDV^\top$, let $\text{LeftSVD}(X) = U$, mapping matrices to the matrix of their left singular vectors. Next, we use overhead bars like $\overline{X}$ and $\overline{Y}$ to denote orthonormal matrices scaled to have spectral norm either $\sqrt{n}$ or $\sqrt{p}$ respectively for explicitness.

Similar to other nonconvex matrix factorization methods (Wang et al., 2016; Yu et al., 2020), our method requires an initial estimate of the rank-k matrix of natural parameters, $\widehat{\Theta}'$, where k is pre-determined. The statistical theory later will explicitly state the requirements that $\widehat{\Theta}'$ needs to fulfill. We provide a concrete initialization method based on Wang et al.

(2016) in the Appendix 5.C.1. Given this initial estimate, consider its SVD $\widehat{\Theta}' = UDV^\top$. To start the alternating minimization stage of our method, we set $\overline{Y}^{(0)} = V$.

After initialization, eSVD then refines the estimate by performing alternating minimization. Specifically, for iterations $t = 0, \ldots, T-1$,

$$X^{(t)} = \operatorname*{argmin}_{X \in \mathbb{R}^{n \times k}} \mathcal{L}_n(X, \overline{Y}^{(t)}) \ : \ X_i^\top \overline{Y}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j), \tag{5.7}$$

$$\overline{X}^{(t+1)} = \sqrt{n} \cdot \mathrm{LeftSVD}(X^{(t)}), \tag{5.8}$$

$$Y^{(t+1)} = \operatorname*{argmin}_{Y \in \mathbb{R}^{p \times k}} \mathcal{L}_n(\overline{X}^{(t+1)}, Y) \ : \ (\overline{X}_i^{(t+1)})^\top Y_j \in \mathcal{R}, \quad \forall (i,j), \tag{5.9}$$

$$\overline{Y}^{(t+1)} = \sqrt{p} \cdot \mathrm{LeftSVD}(Y^{(t+1)}). \tag{5.10}$$

After all T iterations, eSVD outputs the final estimate after a reparameterization. That is, letting $\widehat{\Theta}^{(T)} = \overline{X}^{(T)}(Y^{(T)})^\top$ have a rank-k SVD of $\widehat{U}\widehat{D}\widehat{V}^\top$, the final estimates are

$$\widehat{X}_i = \left(\frac{n}{p}\right)^{1/4} \cdot \left(\sqrt{\widehat{D}_{1,1}} \cdot \widehat{U}_{i,1}, \ldots, \sqrt{\widehat{D}_{k,k}} \cdot \widehat{U}_{i,k}\right), \quad i = 1, \ldots, n, \tag{5.11}$$

$$\widehat{Y}_j = \left(\frac{p}{n}\right)^{1/4} \cdot \left(\sqrt{\widehat{D}_{1,1}} \cdot \widehat{V}_{j,1}, \ldots, \sqrt{\widehat{D}_{k,k}} \cdot \widehat{V}_{j,k}\right), \quad j = 1, \ldots, p. \tag{5.12}$$

This is the same reparameterization used in (5.1) and (5.2).

Usability. In our implementation of our above estimator, we allow F to be the constant-variance Gaussian, curved Gaussian, Poisson or Exponential distribution. Furthermore, if the user wants to use our estimator for a different exponential family distribution F, all she needs to pass into our implementation is the computation of the loss function (5.6) and its gradients as well as information about the domain $\mathcal{R}$.

Remarks about algorithmic design. We make a few remarks about our algorithm to explain its design and relation to other methods. Unlike work such as Gunasekar et al. (2014), we do not estimate Θ directly, where a convex penalty is appended to the objective function to encourage a low rank solution. Instead, by optimizing over X and Y directly, where $XY^\top = \Theta$, we enforce that our estimate of Θ is at most rank k. However, optimizing over X and Y directly raises identifiability issues, since any constant $\delta > 0$, $\mathcal{L}_n(\delta X, Y/\delta) = \mathcal{L}_n(X, Y)$. Work such as Ge et al. (2017) append a penalty term

$$\frac{1}{8}\|X^\top X - Y^\top Y\|_F^2,$$

while Zhao et al. (2015) use the QR-decomposition between iterations, unlike our choice of the LeftSVD($\cdot$) operator. In practice, we found all these choices to behave similarly. The

factors $\sqrt{n}$ and $\sqrt{p}$ in (5.8) and (5.10) are for theoretical reasons to ensure the spectrum of the Hessian is well-controlled and to ensure the values to not underflow if n or p are too large empirically. Also, the final reparameterizations in (5.11) and (5.12) are designed such that the sample second-moment matrices of $\widehat{X}$ and $\widehat{Y}$ are both equal and diagonal, i.e.,

$$\frac{1}{n}\widehat{X}^\top\widehat{X} = \frac{1}{p}\widehat{Y}^\top\widehat{Y}.$$

Lastly, to perform the constrained optimization (5.7) and (5.9), we use Frank-Wolfe (Jaggi, 2013), which we found more stable compared to using alternating projected gradient approaches such as in Wang et al. (2016) where $X^{(t)}$ and $Y^{(t)}$ are separately updated by taking one projected gradient step with respect to the loss function. While there are theoretical guidelines of choosing step-sizes related to the convexity and smoothness for this method, we found these choices often led to poor empirical performance.

5.4.2 Matrix completion diagnostic and tuning heuristic

We provide the following diagnostic to determine which choice of F is most appropriate for our data. Inspired by network cross-validation work such as Li et al. (2016), we use matrix completion to determine the quality of our model fit. To do this, we omit a small percentage of the entries of A when estimating the embedding and compare the observed but omitted values to their predicted expected value counterparts. To compute this expected value, recall the basic property of exponential-family distributions (5.4) where the derivative of the log-partition function $g(\cdot)$ yields the expected value. This is formalized in the algorithm below.

1. For bootstrap trials $b = 1, \ldots, B$:
   a) Randomly sample m of the entries of A, denoted as $\mathcal{O} = \{(i_1, j_1), \ldots, (i_m, j_m)\}$, which will be omitted for this matrix completion diagnostic. Here, m can be any small number, such as $\lceil 0.01 \cdot (np) \rceil$.
   b) Estimate the latent vectors by $\widehat{X}$ and $\widehat{Y}$ according to Subsection 5.4.1 where the objective function $\mathcal{L}_n$ omits the entries in $\mathcal{O}$ and is parameterized to the desired distribution of F.
   c) Compute v_1, defined as the first eigenvector of the matrix formed by the omitted observed values $A_{\mathcal{O}} = \{A_{i_1,j_1}, \ldots, A_{i_m,j_m}\}$ and their predicted expected value counterparts $g'(\widehat{X}\widehat{Y}^\top)_{\mathcal{O}} = \{g'(\widehat{X}_{i_1}^\top \widehat{Y}_{j_1}), \ldots, g'(\widehat{X}_{i_m}^\top \widehat{Y}_{j_m})\}$.
   d) Compute model fit quality, $q^{(b)}$ defined as the angle between v_1 and the vector $(1, 1)$, representing the identity function.
2. Average the model fit qualities across all trials, $q^{(1)}, \ldots, q^{(B)}$.

If we try the above diagnostic for multiple distributions for F, the distribution that yields the smallest average of $q^{(1)}, \ldots, q^{(B)}$ is deemed the most appropriate model for A. In this way, we can try this diagnostic as a tuning heuristic to select the dimensionality of the latent space k, or parameters for exponential-family distributions such as τ in the curved Gaussian distribution (5.5). Observe that we define the model fit quality $q^{(b)}$ as the angle between the first (uncentered) principal component vector[2] between between excluded observed values $A_{\mathcal{O}}$ and predicted values $(\widehat{X}\widehat{Y}^\top)_{\mathcal{O}}$ is to 45°, represented by the vector $(1, 1)$. Having an eigenvector's angle close to 45° means that on average, the predicted values model the observed value well. We do not use MSE to define the model fit quality since the variance can grow with the expected value for various exponential family distributions such as our curved Gaussian model (5.5).

5.5 Statistical theory

We derive the statistical rate of convergence when eSVD is applied to the model described in §5.3. The overall rate is divided into two components. First, we analyze the convergence of $\widehat{\Theta}$ to Θ when conditioned on the latent vectors in X and Y. Second, we analyze the convergence of $\widehat{X}$ to X, given some additional identifiability conditions, for a generic estimator $\widehat{\Theta}$. By combining the two components, applied to the curved Gaussian model, we can derive the overall rate of convergence for eSVD as it pertains to the analysis in this paper. The proofs for all the results are in Appendix 5.H. Our analysis draws inspirations from Zhao et al. (2015) which also studies alternating minimization, but only for the constant-variance Gaussian model.

Throughout this section, conditioned on X and Y (the matrix of random latent vectors for each cell and gene described in (5.5) that we are trying to estimate), let the SVD of Θ be denoted as

$$\Theta = \widetilde{U}\widetilde{D}\widetilde{V}^\top, \tag{5.13}$$

where we denote the singular values as $\widetilde{d}_1, \ldots, \widetilde{d}_k$. In addition, for a generic matrix A, let $\|A\|_F$ denote its Forbenius norm. We use standard asymptotic notation throughout this section. For two sequences a_n and b_n, let $a_n = O(b_n)$ denote that a_n/b_n is bounded for large enough n, and for two random sequences A_n and B_n, let $A_n = O_P(B_n)$ denote that A_n/B_n is bounded in probability for large enough n.

5.5.1 Estimation of matrix of natural parameters

We operate under the correctly-specified model setting, where eSVD uses the log-likelihood of F. Our first set of assumptions ensure that the exponential family distribution F is

[2] By this, we mean v_1 is equivalent to doing PCA on the matrix formed by the columns $A_{\mathcal{O}}$ and $(\widehat{X}\widehat{Y}^\top)_{\mathcal{O}}$ where we purposefully do not center the columns of this matrix, and extract the first principal component vector.

well-behaved. We define notation to distinguish between sample and population optimizers, borrowing terminology from Balakrishnan et al. (2017). Let $M_n^X(\overline{Y}^{(t)})$ and $M_n^Y(\overline{X}^{(t+1)})$ denote $X^{(t)}$ and $Y^{(t+1)}$ respectively in (5.7) and (5.9). In the nonconvex literature, these denote the *sample minimization operators*. Likewise, we define the population loss function,

$$\mathcal{L}(X,Y) = \frac{1}{np}\sum_{(i,j)}\Big[g(X_i^\top Y_j) - \mathbb{E}\big[\eta(A_{ij})\big]^\top T(X_i^\top Y_j)\Big], \tag{5.14}$$

and the corresponding *population minimization operators*,

$$M^X(\overline{Y}^{(t)}) = \operatorname*{argmin}_{X\in\mathbb{R}^{n\times k}} \mathcal{L}(X,\overline{Y}^{(t)}) \ : \ X_i^\top \overline{Y}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j), \tag{5.15}$$

$$M^Y(\overline{X}^{(t+1)}) = \operatorname*{argmin}_{Y\in\mathbb{R}^{p\times k}} \mathcal{L}(\overline{X}^{(t+1)},Y) \ : \ (\overline{X}_i^{(t+1)})^\top Y_j \in \mathcal{R}, \quad \forall (i,j). \tag{5.16}$$

To handle identifiability issues, let us define the set of matrix pairs,

$$\{\overline{\mathcal{X}}^*, \mathcal{Y}^*\} = \Big\{\{\overline{X}, Y\} \ : \ \Theta = \overline{X}Y^\top\Big\}.$$

By the SVD of Θ in (5.13), we can see that the set $\overline{\mathcal{X}}^*$ represents all matrices that are equal to $\sqrt{n}\cdot\widetilde{U}$ up to rotation. Similarly, we can define the pair of spaces $\{\mathcal{X}^*, \overline{\mathcal{Y}}^*\}$.

Assumption 2 (Strong convexity and gradient Lipschitz)**.** *Assume that the population negative log-likelihood function $\mathcal{L}(\cdot,\cdot)$ is μ-strongly convex and its gradient is L-Lipschitz for $L \geq \mu > 0$ after fixing one of the input matrices. Specifically, for any matrices $X, X' \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{p\times k}$ satisfying Assumption 1,*

$$\frac{\mu}{2}\|X - X'\|_F^2 \leq \mathcal{L}(X',Y) - \mathcal{L}(X,Y) - \Big\langle X' - X, \nabla_X \mathcal{L}(X,Y)\Big\rangle \leq \frac{L}{2}\|X - X'\|_F^2,$$

and a similar assumption holds for any matrices $Y, Y' \in \mathbb{R}^{p\times k}$ and $X \in \mathbb{R}^{n\times k}$.

Assumption 3 (Gradient Lipschitz with respect to alternating variable)**.** *Assume that there exists a $S > 0$ such that for any orthonormal matrices $\overline{Y}$ and $\overline{X}$ rescaled to have spectral norm $\sqrt{p}$ and $\sqrt{n}$ respectively, any pairs $\{X^*, \overline{Y}^*\} \in \{\mathcal{X}^*, \overline{\mathcal{Y}}^*\}$ and $\{\overline{X}^*, Y^*\} \in \{\overline{\mathcal{X}}^*, \mathcal{Y}^*\}$ satisfy*

$$\|\nabla_X \mathcal{L}(X^*, \overline{Y}) - \nabla_X \mathcal{L}(X^*, \overline{Y}^*)\|_F \leq S\|\overline{Y} - \overline{Y}^*\|_F,$$
$$\|\nabla_Y \mathcal{L}(\overline{X}, Y^*) - \nabla_Y \mathcal{L}(\overline{X}^*, Y^*)\|_F \leq S\|\overline{X} - \overline{X}^*\|_F.$$

For the below assumptions, let

$$B_F(r; \overline{\mathcal{Y}}^*) = \Big\{Y' \in \mathbb{R}^{p\times k} : \text{there exists } \overline{Y}^* \in \overline{\mathcal{Y}}^* \text{ such that } \|Y' - \overline{Y}^*\|_F \leq r\Big\},$$

in other words, the union of Forbenius balls of matrices around any $\overline{\mathcal{Y}}^*$ with radius r.

Assumption 4 (Uniform statistical error)**.** *Conditioned on X and Y, assume that with probability at least $1 - c/\min\{n,p\}$ for some universal constant c that*

$$\sup_{Y' \in \mathbb{B}_F(r;\, \overline{\mathcal{Y}}^*)} \|M^X(Y') - M_n^X(Y')\|_F \leq \varepsilon_{unif}(n,p),$$
$$\sup_{X' \in \mathbb{B}_F(r;\, \overline{\mathcal{X}}^*)} \|M^Y(X') - M_n^Y(X')\|_F \leq \varepsilon_{unif}(n,p),$$

where $r = \widetilde{d}_k \mu / (4 \max\{n,p\}^{1/2} S)$ and $\varepsilon_{stat}(n,p)$ is some function of n and p (and possibly other quantities). Assume that

$$\varepsilon_{unif}(n,p) \leq \frac{\widetilde{d}_k}{4 \max\{n,p\}^{1/2}}.$$

In Assumptions 2 and 3, the strongly convexity and smoothness enable fast convergence. These assumptions are common in work that study matrix factorization, i.e. Wang et al. (2016) and Yu et al. (2020). Likewise, Assumption 4 ensures the sample optimizer does not deviate too far from the population optimizer, commonly used in work such as Balakrishnan et al. (2017). Later in this section, we explain that Assumptions 2, 3 and 4 are satisfied in the curved Gaussian model (5.5) in the appropriate setting.

Assumption 5 (Initialization condition)**.** *Conditioned on X and Y, assume that for some pair of matrices $\overline{Y}^* \in \overline{\mathcal{Y}}^*$, the initial estimate $\overline{Y}^{(0)}$ satisfies*

$$\|\overline{Y}^* - \overline{Y}^{(0)}\|_F \leq \frac{\widetilde{d}_k}{4 \max\{n,p\}^{1/2}} \frac{\mu}{S}.$$

Assumption 5 is an initialization condition similar to Wang et al. (2016) and Yu et al. (2020) that ensures the alternating minimization steps can allow $\widehat{\Theta}^{(T)}$ to converge towards Θ. This is described in the following proposition.

Proposition 35. *Under Assumptions 1 and 2-5, conditioned on X and Y, if*

$$\frac{4(np)^{1/2}}{\widetilde{d}_k} \frac{S}{\mu} < 1,$$

then for the number of iterations T large enough, with probability at least $1 - 2c/\min\{n,p\}$, eSVD described in (5.7)-(5.10) *achieves the rate*

$$\|\Theta - \widehat{\Theta}^{(T)}\|_F = C \cdot \Big(\frac{\max\{n^{1/2}, \widetilde{d}_1/n^{1/2}\} \cdot (np)^{1/2}}{\widetilde{d}_k - 4(np)^{1/2} S/\mu} \cdot \varepsilon_{unif}(n,p) \Big),$$

for some universal constants c and C.

To apply Proposition 35 to a particular model, one needs to compute the terms μ, L, S, and $\varepsilon(n,p)$ needed for Assumptions 2-4. This often requires positing assumptions in the context of the particular model that imply Assumptions 2-4. We demonstrate this below with minor modifications.

Application to the curved Gaussian model. Using broader assumptions that imply Assumptions 1, 2 and 3 as well as slightly modification of eSVD to make Assumption 4 easier to analyze, we can apply Proposition 35 to the setting where F is the curved Gaussian distribution (5.5) and $\mathcal{L}_n(\cdot,\cdot)$ is its corresponding negative log-likelihood. More specifically, assume the for each iteration $t = 0, \ldots, T-1$ uses a different set of observed matrices A. This simplifying assumption has been used in other work such as Wang et al. (2015) where this is no closed-formed solution to (5.7) or (5.9). Then, roughly speaking, if $n = O(p)$ and $p = O(n)$, then

$$\|\Theta - \widehat{\Theta}^{(T)}\|_F = O_P\Big(n^{3/4}\log^{1/4}(n)\Big). \tag{5.17}$$

We defer the details, including specific descriptions of the broader assumptions, to Appendix 5.D for brevity.

5.5.2 Estimation of latent vectors

Given the convergence rate for estimating Θ, we can next ask how how well we estimate the latent vectors $X_1, \ldots, X_n$ or $Y_1, \ldots, Y_p$ themselves. In fact, our theorem below requires enforcing assumptions on G and H for identifiability reasons (see Assumption 7) and holds generically for *any* method to estimate Θ as long as $\|\Theta - \widehat{\Theta}\|_F = O_P(\epsilon)$ for some rate function ϵ. We use the following notation to define the population second moment matrices of X_i and Y_j for any $i = 1, \ldots, n$ and $j = 1, \ldots, p$ as

$$\mathbb{E}[X_i X_i^\top] = C_X^* = \Phi^*\Lambda^*\Phi^{*\top}, \quad \text{and} \quad \mathbb{E}[Y_j Y_j^\top] = C_Y^* = \Psi^*\Gamma^*\Psi^{*\top},$$

with their corresponding eigen-decompositions.

Assumption 6 (Sub-exponential distribution of latent vectors)**.** *Assume that the square of X_i and Y_j in the model* (5.3) *are multivariate sub-exponentially distributed. That is, there exists a constant D such that for any vector $V \in \mathbb{R}^k$ and any $c \geq 1$,*

$$\Big(\mathbb{E}\Big[\big|\langle X_i, V\rangle^2 - \mathbb{E}\big[\langle X_i, V\rangle\big]^2\big|^c\Big]\Big)^{1/c} \leq Dc,$$

and a similar assumption holds for Y_j with also the same constant D.

Assumption 7 (Second moment properties)**.** *The population second moment matrices C_X^* and C_Y^* are equal and are both diagonal matrices, where $(C_X^*)_{i,i} \geq (C_X^*)_{j,j}$ for any*

$1 \le i < j \le k$. *Furthermore, assume there exists positive numbers* $c_1 \le c_2$ *and* $1 < \alpha \le \beta$ *such that for all* $\ell = 1, \ldots, k$, *the eigenvalues satisfiy*

$$c_1 \ell^{-\alpha} \le \lambda_\ell^* \le c_2 \ell^{-\alpha}, \quad \textit{and} \quad \lambda_\ell^* - \lambda_{\ell+1}^* \ge c_1 \ell^{-\beta}.$$

Assumption 6 enables sharp rates for estimating the second-moment matrix C_X^* and C_Y^*, while the second part of Assumption 7 enables our estimator to accurately estimate its eigenvalues and eigenvectors. Both assumptions are common in work that study the spectrum, i.e., Lei (2018). The first part of Assumption 7 is an identifiability condition, which we show in Appendix 5.C can always be satisfied after some reparameterization.

Proposition 36. *Assume the model in* (5.3) *where Assumptions 6 and 7 hold. If the estimator* $\widehat{\Theta}$ *satisfies* $\|\widehat{\Theta} - \Theta\|_F \le \epsilon$ *conditioned on* X *and* Y, *and* $k = o(\min\{n, p\})$, *then up to sign*[3], *eSVD achieves the rate after reparamterizations* (5.11) *and* (5.12)

$$\frac{1}{n}\|X - \widehat{X}\|_F^2 = O_P\Big(\max\Big\{\frac{k^{4\beta-\alpha+4}}{\min\{n,p\}}, \ \frac{k^{2\beta-\alpha+2}\max(\epsilon^2, \epsilon)}{np}\Big\}\Big). \tag{5.18}$$

Discussion of rate. Notice that rate (5.18) gets worse the larger β is, representing a smaller eigen-gap according to Assumption 7. Also, typically the latter term within the maximization in (5.18) is the dominant term, unless $\epsilon = O((np/\min\{n,p\})^{1/2})$. Also, a similar rate (5.18) holds for estimating Y.

Application to curved Gaussian model. As discussed in §5.5.1, if we plug in the rate of convergence applied to the curved Gaussian model shown in (5.17) after inheriting its assumptions, for the setting $n = O(p)$ and $p = O(n)$, we obtain roughly

$$\frac{1}{n}\|X - \widehat{X}\|_F^2 = O_P\Big(\frac{\log^{1/2}(n)}{n^{1/2}}\Big).$$

We defer the details such as the assumptions to Appendix 5.D for brevity.

5.6 Numerical study

In this section, we study the performance of eSVD and compare it to other methods using synthetic data. Our setup for all the simulations in this section are as follows: following the model (5.3), we set the known $k = 2$ and sample $X_1, \ldots, X_n$ i.i.d. uniformly from four connected linear segments with additive Gaussian noise, as illustrated in Figure 5.4. These four segments loosely represent four cell types. We also sample $Y_1, \ldots, Y_p$ i.i.d. from a

[3] We use "up to sign" similar to Fan et al. (2018b), where each column of $\widehat{X}$ can be multiplied by ± 1 since the SVD is not unique.

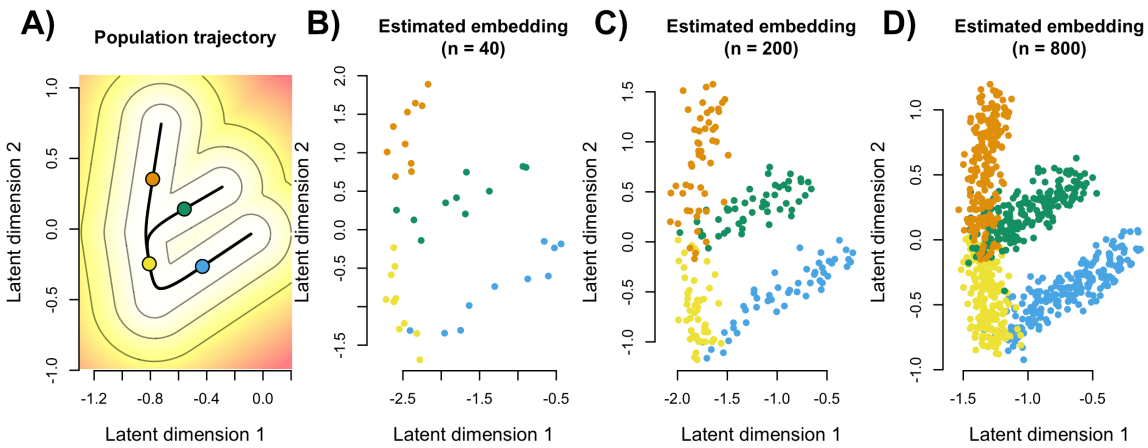


Figure 5.4: (A) The two-dimensional population density of G, visualized as a heat map with contour lines along with the true trajectories (black lines). The mean vector for each of four cell types are labeled in a different color (blue, yellow, green, orange). (B to D) The estimated embedding $\widehat{X}_{1:n}$ of the synthetically-generated A for varying levels of n, (i.e., number of cells or number of rows), colored by the true cell-type, with the estimated trajectories overlaid ontop.

mixture of two Gaussians that are well-separated. These sampling procedures represent G and H respectively, up to identifiability conditions. We enforce that $X_i^T Y_j < 0$ for all pairs (i, j). The distribution family F however changes between different simulations. We do not use `R` packages such as Splatter (Zappia et al., 2017) to generate our synthetic data since we want to have precise control over the true embedding. The full details of the simulation setups and usage of various estimators in this section are in Appendix 5.E.

Asymptotic convergence to true embedding. In this first simulation suite, we demonstrate that the estimated embedding converges towards the true latent embedding. Specifically, we generate $A \in \mathbb{R}^{n \times p}$ where each entry A_{ij} is sampled independently from the curved Gaussian (5.5) with a natural parameter $\theta_{ij} = X_i^\top Y_j$ and $\tau = 2$, and fit eSVD using the correctly specified model. Figure 5.4 is an illustration that demonstrates the asymptotic properties of eSVD. Specifically, we see that the distribution of the embedding $\widehat{X}_1, \ldots, \widehat{X}_n$ converges towards G as n increases. We provide a thorough simulation study of the convergence rates in Appendix 5.E.

Comparison of different embedding methods. For our second simulation suite, we demonstrate that the estimated relative positions of the cells' latent positions derived from eSVD are more accurate than those estimated by other methods even when the distribution F is misspecified. Here, we fix $n = 200$ and $d = 240$. We compare eSVD to five other methods used to embed single cells: SVD, ZINB-WaVE (Risso et al., 2018), pCMF (Durif et al.,

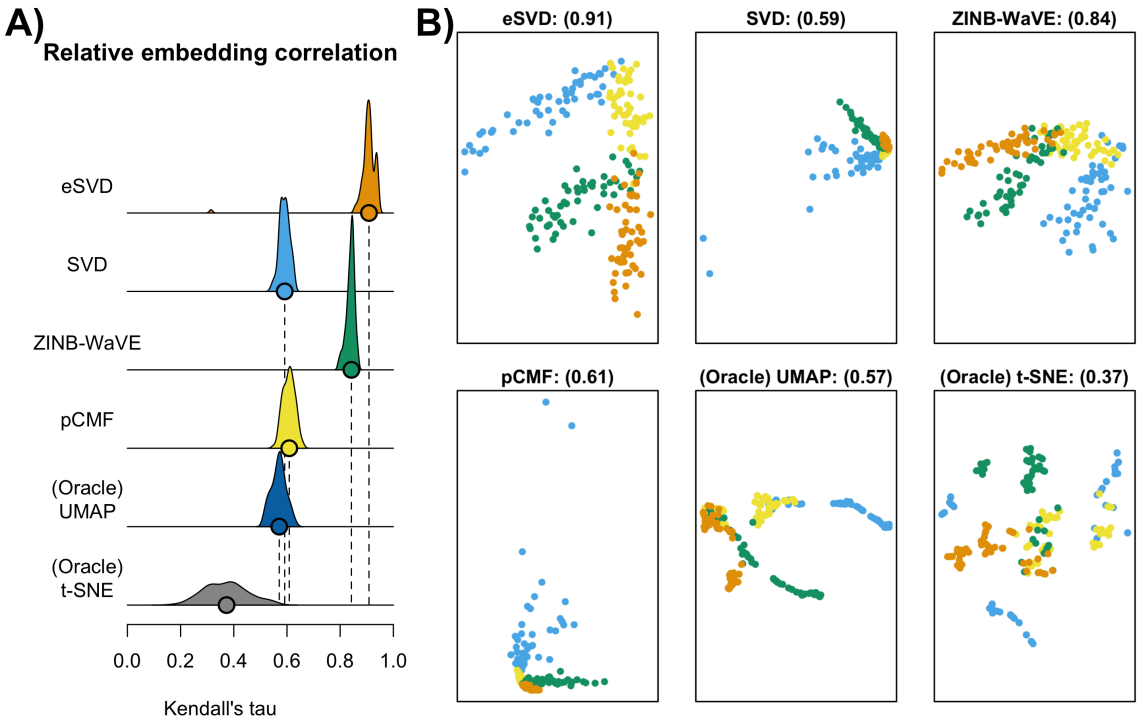


Figure 5.5: (A) The density plot of each of the six embedding methods's accuracy (eSVD, SVD, ZINF-WaVE, pCMF, UMAP and t-SNE), based on our Kendall's tau correlation metric. The circles along each method's x-axis denotes the median accuracy across the 200 trials. (B) For a particular instance of A, the six estimated embedding, showing the two latent dimensions (first latent dimension on the x-axis). The number in parenthesis denotes the accuracy of the embedding with respect to the true embedding. The coloring of the samples persists from Figure 5.4.

2017), UMAP (Becht et al., 2019) and t-SNE (Maaten and Hinton, 2008), The second and third methods are explained in Appendix 5.E. Importantly, SVD implicitly assumes F is a constant-variance Gaussian distribution as mentioned in §5.1, while ZINB-WaVE and pCMF assume F is a Negative Binomial and Poisson distribution respectively.

We simulate data from a negative binomial model in this simulate suite, which is the distribution family that is most commonly used to model RNA-seq data (Love et al., 2014). Specifically, we sample the observed count matrix A conditionally independent on $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_p$ where

$$A_{ij} \sim \text{Negative Binomial}\big(50, \exp(-X_i^\top Y_j\big). \tag{5.19}$$

Then, when we estimate the embedding using eSVD, we use the matrix-completion diagnostic mentioned in §5.4 to select the most appropriate value of the dispersion parameter π from

the set $\{5, 50, 100\}$.

We find that on average across 50 trials, our method estimates the relative latent positions of each cell to be more accurate than other methods (Figure 5.5A). To define our notion of accuracy, consider each cell i and its Euclidean distance to all other $n-1$ cells in the latent space in both the true and estimated embedding. We then compute the Kendall's tau correlation between these two vectors, which only relies on the ranks of the distances, and then average this value over all n cells. Hence, a high mean Kendall's tau value suggests that the four different cell types we posited remain relatively well separated. We call this notion of accuracy as the "relative embedding correlation." Figure 5.5B compares the different estimated embeddings to the true embedding as an illustration. We see that eSVD and ZINB-WaVE both estimate embeddings where the four cell types are relatively in the correct configuration, and their accuracy are quite high. In this example, the other four embeddings do not perform as well.

Investigation using other generative models. We defer our third simulation suite to Appendix 5.E, where we empirically compare eSVD to other methods when F is *misspecified*. While the theorems we developed in §5.5 are no longer valid under these settings, we are nonetheless interested if our method roughly estimates the relative positions of the cells' embedding correctly. As we elaborate in Appendix 5.E, our takeaway message is that there are generative models where any of the six methods is the most appropriate. Hence, it is important to use diagnostics such as the one we provided in §5.4 to understand which method is most suitable for the data at hand.

5.7 Single-cell analysis

We return to the Marques dataset (Marques et al., 2016), as described in §5.2, to determine if the embedding based on the curved Gaussian model (5.5) is more appropriate than that based on the constant-variance Gaussian model for modeling oligodendrocytes and if so, continue our motivating task of investigating the developmental trajectories. As alluded to in §5.2 the six major cell types in Figure 5.1 have a determined order, starting from Pdgfra+ precursors and ending at the mature oligodendrocytes. However, our goal in the analysis to follow is to estimate the trajectories among the cell sub-types constrained to this order. For example, in Marques et al. (2016), the authors estimated one developmental trajectory connecting all the first five cell major types starting from the Pdgfra+ precursors after embedding the cells into a lower-dimension space, but do not definitively conclude how the six mature oligodendrocyte cell sub-types differentiate. Instead, they relied on region-based analyses to hypothesize that these six sub-types differentiate into five different trajectories.

5.7.1 Details of estimating cell developmental trajectory

We provide more details on how we estimate the developmental trajectories based on the low-dimensional embedding $\widehat{X}_1, \ldots, \widehat{X}_n$. As alluded to before, the trajectories show how these different cell sub-types develop from one another, assuming the latent vectors $\widehat{X}_i$ for each cell i gradually changes along the trajectories. We use Slingshot (Street et al., 2018) (with minor modifications) to estimate these cell developmental trajectories. Roughly speaking, Slingshot is a two-step algorithm that requires the latent vectors to already be clustered, where we treat each cell sub-type as a cluster. In the first stage, Slingshot estimates the number of trajectories and ordering of the cell sub-types based on minimizing the distances between cell sub-type centers via a minimum spanning tree. In the second stage, Slingshot fits principal curves (Hastie and Stuetzle, 1989) that passes through the cell sub-type centers in the estimated ordering, while deploying algorithmic tricks to merge different principal curves whose lineages share cell sub-types. These principal curves can be thought as smooth curves that pass through high density regions in the latent dimensions. More details about Slingshot and our modifications of it to make it more suitable for our dataset are given in Appendix 5.F.

We briefly mention that the original study (Marques et al., 2016) uses a different algorithm, Monocle (Trapnell et al., 2014), to estimate the cell developmental trajectories. We use Slingshot instead as it is the current state-of-the-art method based on extensive benchmarking comparisons in Saelens et al. (2019). As we will show below, despite using a different method to estimate the cell trajectories, we reach the same scientific conclusion as in (Marques et al., 2016) if we were to embed the cells according the constant-variance Gaussian model.

5.7.2 Analysis using constant-variance Gaussian distribution

Building on the analysis in §5.2, we perform a trajectory analysis using the SVD embedding shown in (5.1) and (5.2), which assumes the constant-variance Gaussian model. Applying Slingshot directly to this embedding results in two trajectories, both heavily overlapping one another (Figure 5.6A). These results are similar to Marques et al. (2016) in two ways. First, the authors show that all cells develop from Pdgfra+ precursors to myelin-forming oligodendrocytes in the same way, which we estimate as well. Second, the authors do not definitively conclude if the mature oligodendrocytes diverge in their development. Our trajectories themselves also leave this ambiguity unresolved due to the heavy overlap between the two trajectories. However, we perform the following additional visual diagnostic to quantify if these two trajectories are well approximated by a single trajectory in actuality.

To formalize to what degree the different trajectories are the same, we use bootstrap resampling to construct a uniform uncertainty tube around each trajectory. These tubes capture the variance of each estimated trajectory, and plotting these tubes is a useful descriptive tool. This is an important component of our analysis since Slingshot is sensitive

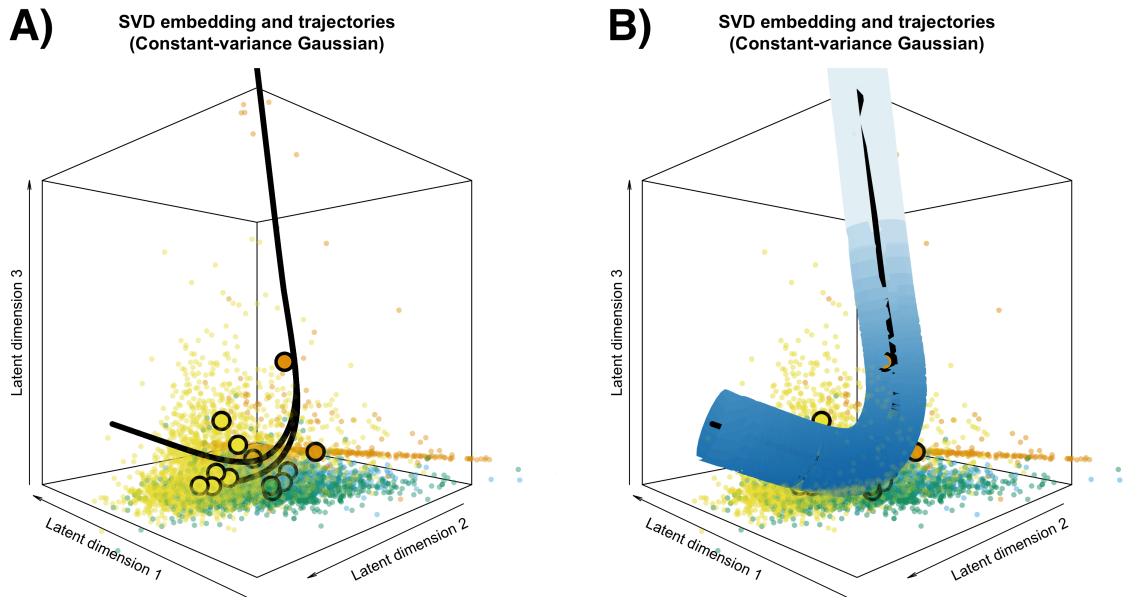


Figure 5.6: (A) Three-dimensional plot of the estimated latent positions via SVD with the two estimated cell developmental trajectories laid on top, corresponding to the data shown in the Figure 5.1. The thirteen bolded points correspond to the cluster centers of the thirteen cell sub-types, where the color scheme persists from Figure 5.1. (B) The uncertainty tube overlaid on top of Figure A.

to small perturbations in the data due to its graph-based strategy to estimate the ordering of the cell sub-types. Hence, small variations can dramatically change the number of estimated trajectories or ordering of cell sub-types within those trajectories. Our procedure samples with replacement from all embedded cells within each of the thirteen cell sub-types. For each bootstrap sample, we apply Slingshot to estimate a new set of trajectories. We then compute the ℓ_2 distance between the new trajectories and the original trajectories. After applying this procedure multiple times, the 95% quantile of the ℓ_2 distances determines the uniform radius of the uncertainty tube, centered around the original trajectory. More details of this procedure are in Appendix 5.F. Based on this construction, all five trajectories lie in the uncertainty tube of the longest trajectory (Figure 5.6B). Hence, we conclude there is effectively one trajectory that connects all thirteen cell sub-types. This result can explain why embeddings based on less flexible statistical models such as in Marques et al. (2016) have difficulty explaining how mature oligodendrocytes differentiate in their trajectory analysis.

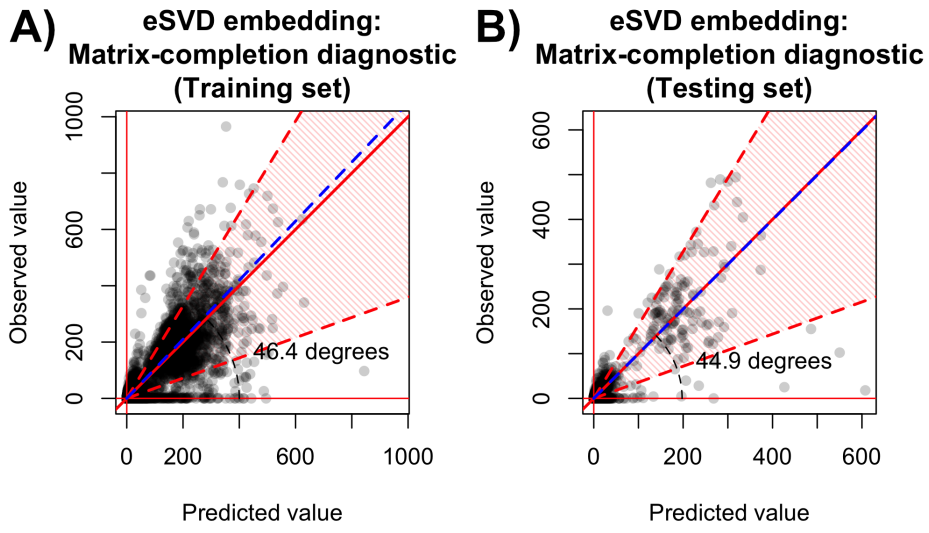


Figure 5.7: (A) Diagnostic plot showing the observed values that were not omitted (i.e., the "training set") verses their respective predicted values. (B) Diagnostic plot showing the observed values that were purposefully omitted (i.e., the "testing set") verses their respective predicted values. Both plots are comparable to those in Figure 5.2, except the fit shown here is estimated via eSVD for the the curved Gaussian model with $k = 5$ and $\tau = 2$. The 10th to 90th quantiles of the curved Gaussian model are more pronounced here, shown in the shaded red region.

5.7.3 Analysis using curved Gaussian model

The above conclusions, however, rest on the questionable constant-variance Gaussian distributional assumption (see Figure 5.3). As we've seen in Figure 5.2 in §5.2 that our matrix-completion suggests that this assumption is not suited for modeling the oligodendrocyte dataset at hand.

This finding motivates us to analyze the data using eSVD to embed each cell with respect to the curved Gaussian distribution (5.5), and re-examine the resulting diagnostics. Based on our tuning heuristic, the curved Gaussian distribution with $k = 5$ and $\tau = 2$ best fits the data, determined among the candidate values of $\{3, 4, 5\}$ and $\{0.5, 1, 2, 4\}$ respectively. When we plot the diagnostic using eSVD in Figure 5.7, we obtain results that suggest a much better fit compared to that of SVD. Not only is the principal angle close to 45° in the testing set, meaning eSVD models the mean function more appropriately, but also a larger fraction of the observations lie within the 10% to 90% quantile region of the distribution. We conclude that the curved Gaussian model is more appropriate than the constant-variance Gaussian model for modeling our oligodendrocyte dataset. The eSVD embedding can be compared to the SVD embedding (Figure 5.8A vs. Figure 5.1), where we similarly mark the

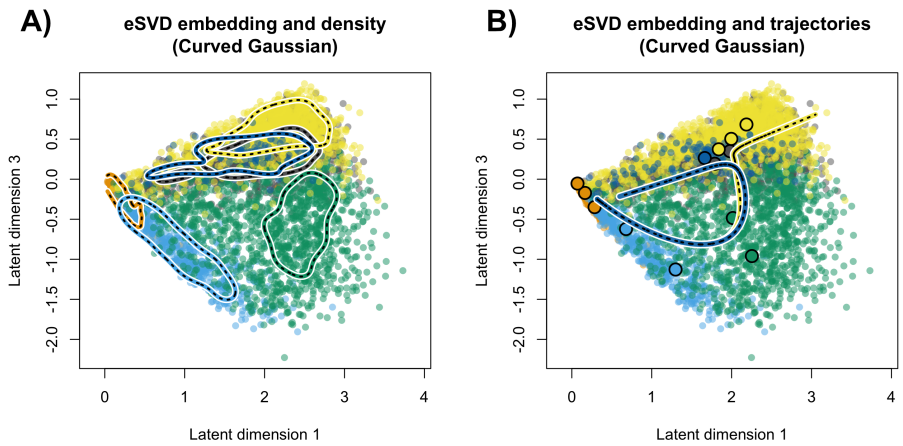


Figure 5.8: eSVD embedding using curved Gaussian distribution with $k = 5$ and $\tau = 2$. The coloring of cells persists from Figure 5.1, with exception of mature oligodendrocytes which now are colored as gray, yellow or blue based on the estimated trajectories shown in Plot B. The first and third latent dimensions are shown. (A) The embedding along with contours of the estimated densities to visualize high-density regions. This plot is comparable to Figure 5.1. (B) The embedding along with both estimated cell developmental trajectories, colored in yellow and blue. These correspond with the three of six mature oligodendrocytes cell sub-types unique to one trajectory (colored in yellow) and the mature oligodendrocytes cell sub-type unique to the other trajectory (colored in blue). The two remaining mature oligodendrocytes are common to both trajectories, prior to the branching (colored in gray). The thirteen bolded points correspond to the cluster centers of the thirteen cell sub-types.

high density regions of various cell types using contour lines. We see that the cells within each major cell type are relatively spread out, a promising feature that suggest eSVD was able to capture the variance among the cells and can support developmental theories of gradual transcriptional change.

When we apply Slingshot to the eSVD embedding, we find that we still retain the conclusion that all cells from Pdgfra+ precursors to myelin-forming oligodendrocytes develop in the same way, similar to Marques et al. (2016), as shown in (Figure 5.8B. However, unlike that work, we are now able to observe a substantial differentiation among the mature oligodendrocytes into two distinct trajectories. Specifically, within this major cell type, only two of the six mature oligodendrocytes sub-types are shared between the two trajectories. Among the five remaining mature oligodendrocytes sub-types, four sub-types branch off in one trajectories while one sub-types branch into the other trajectory. Similar to before, Figure 5.9 displays the embeddings in three-dimensions along with the uncertainty tubes. We show the additional plots of the embedding and trajectory estimates in Appendix 5.G.

We see that in Figure 5.9B, even with the uncertainty tube, the two trajectories that we

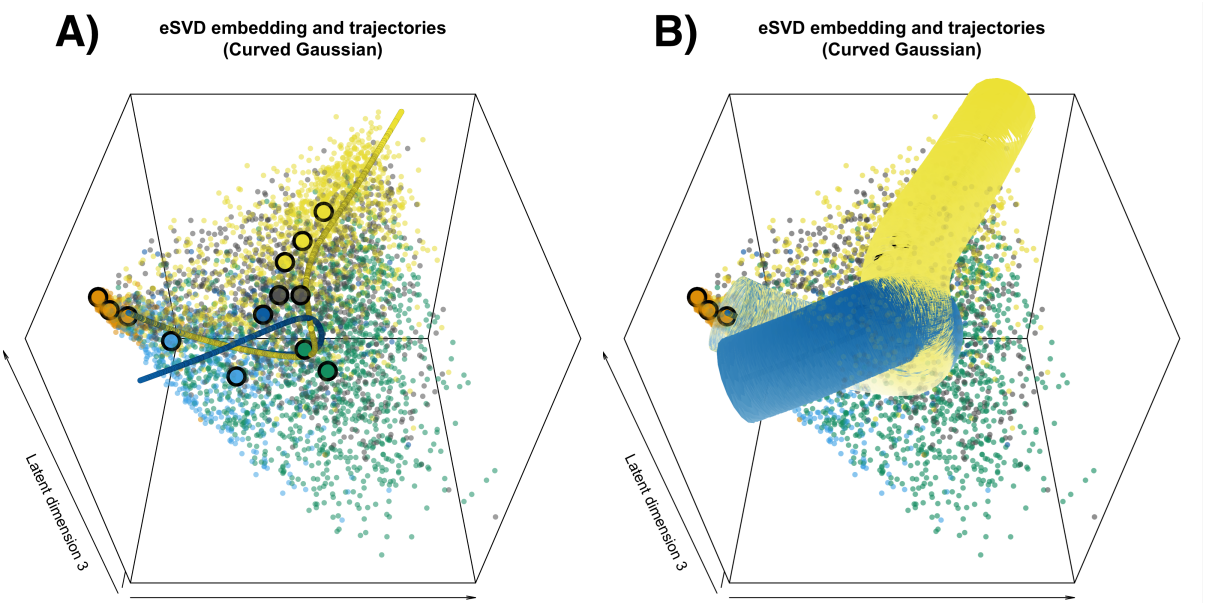


Figure 5.9: (A) Three-dimensional plot of the estimated latent positions via eSVD for curved Gaussian distributions with $\tau = 2$ with the estimated cell developmental trajectory laid on top, corresponding to the data shown in Figure 5.8B. The thirteen bolded points correspond to the cluster centers of the thirteen cell sub-types, where the color scheme persists from Figure 5.8. (B) The uncertainty tubes overlaid on top of Figure A. Both plots are comparable to Figure 5.6.

estimate using the eSVD embedding are still well separated at the mature oligodendrocytes. This is contrast with the analysis using SVD where all the estimated trajectories lied within one uncertainty tube (Figure 5.6B). Hence, from the diagnostic shown in Figure 5.8, we conclude that the curved Gaussian distribution is more appropriate for the Marques data, and using this model results in estimating that the oligodendrocytes develop in two distinct trajectories. This is an improvement from the Marques et al. (2016) analysis which suggested five trajectories but was not able to determine how distinct these trajectories were. Our comparison of results between using the SVD or eSVD embeddings can help explain why previous scientific findings suggested that oligodendrocytes effectively a single developmental trajectory, while newer analyses based on more flexible statistical models might suggest multiple trajectories.

5.8 Discussion

In this article, we develop an estimator to embed the cells in a single-cell RNA-seq dataset into a lower dimensional space with respect to a hierarchical model where the inner product of two latent vectors is the natural parameter of a one-parameter exponential family distribution F.

This embedding method can greatly improve the estimation of cell developmental trajectories overall since it can handle distributions beyond the constant-variance Gaussian distribution, both in theory and practice. While the spirit of such embedding is not new, our contribution is two-fold. First, we develop eSVD, an efficient estimator that avoids using semidefinite programs, and solidify its statistical properties such as identifiability and consistency. Second, we apply our estimator to analyze the oligodendrocytes in mouse brains, showing results that coincide with recent scientific hypotheses (van Bruggen et al., 2017; Marques et al., 2018).

For future work, our work can be extended to analyze statistical embeddings that model the dropout effect directly, such as Pierson and Yau (2015), Townes et al. (2017) and Risso et al. (2018). In addition, models such as the one in Risso et al. (2018) allow additional covariates to play a role in the embedding, such as the cell size or gene length. We also plan to work on extending our hierarchical latent model to clustering and imputation uses, much like Lopez et al. (2018). Unlike Lopez et al. (2018) however, since we have statistical theory on what eSVD's embedding converges towards, we hope to also theoretically analyze the statistical performance of these downstream methods.

5.A Code and reproducibility

The code for the method, simulation, and data analysis, as well as the original data used, can be found at `https://github.com/linnylin92/esvd`, in the `eSVD`, `simulation`, `main` and `data` folders respectively. The dataset we analyzed was originally collected by Marques et al. (2016), found at the Gene Expression Omnibus with accession number GSE75330 (`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75330`).

5.B Formal description of analysis pipeline

5.B.1 Main analysis pipeline

The following procedure describes how we preprocess the data prior to the preliminary analysis in §5.2 as well as how the analysis was performed in §5.7.

1. **Screening genes**: Since not all 23,556 genes are informative for our analysis, we use the following two methods to select genes, based loosely on Zhu et al. (2019).

    - **Sparse principal component analysis** (Witten et al., 2009): This method uses a tuning parameter that controls the ℓ_1-norm of the eigenvectors, of which we try 10 values spaced exponentially between 0 and $\log(23,556)$. We choose the model such that the first $K = 5$ sparse eigenvectors involve more than 500 genes and would capture more than 90% of the variance.

- **DESCEND** (Wang et al., 2018): We find genes with a Gini index with a normalized difference (compared to the mean) of 50.

Together, this results in 983 unique genes being selected. In the following parts of the analysis, we will denote the preprocessed data as $A \in \mathbb{R}_+^{n \times p}$ where there are $n = 5069$ cells and $p = 983$ genes.

2. **Rescaling**: We normalize each cell by its read-depth (i.e., divide each row by its sum) and then multiply the entire matrix by a scalar such that the maximum value is 1000. This is to prevent small numbers from underflowing our method later on.

3. **Tuning the dimensionality of the embedding k and the nuisance parameter of the curved Gaussian distribution τ**: Omitting four entries per row and per column of A (for a total of 24,193 unique entries) for each trial of b over a total of $B = 3$ trials, we use the matrix completion heuristic outlined in Subsection 5.4.2 to select τ from potential values of $k \in \{3, 4, 5\}$ and $\tau \in \{0.5, 1, 2, 4\}$ (for a total of 12 different parameter settings). Based on which principal angle is closes to 45°, we end up selecting $k = 5$ and $\tau = 2$. Note that it is important to refit across the different values of k since non-linear embeddings are not typically nested, as discussed in Durif et al. (2017).

4. **Embedding via eSVD**: We apply eSVD where F is the the curved Gaussian distribution with $k = 5$ and $\tau = 2$ to minimize (5.6), as prescribed in Subsection 5.4.1. The initialization procedure is described in Appendix 5.C.1.

5. **Estimating the cell developmental trajectories and uncertainty tube**: Using the thirteen cell sub-types from Marques et al. (2016) as the cluster labels, we apply our modified Slingshot and construct the uncertainty tubes, as alluded to in §5.7 and detailed in Appendix 5.F.

5.B.2 Additional details of analysis in §5.2

We describe additional details of the analysis used to produce the various figures in §5.2.

Details for Figure 5.1. The scatter plot is produced the SVD embedding described in (5.1) and (5.2) for $k = 2$, where the coloring of the points is based on the cell-type information provided in Marques et al. (2016). The contour of the densities estimated based on the `MASS::kde2d` function (using the default bandwith), where the level of the contour is chosen to be the 92.5% quantile of the estimated density across the grid of points in this two-dimensional space. This quantile level is chosen solely based on the suitably of the figure, and provides the reader a sense of the density of the points that is otherwise hard to gauge based on only the scatterplot.

Details for Figure 5.2. SoftImpute (Mazumder et al., 2010) requires the dimensionality of the latent space k and a tuning parameter λ to determine the severity of the spectral regularization. To choose this, we try 50 different value of λ from 1 to the value given in `softImpute::lambda0` (a function that computes the largest value of λ that still yields the all-0 estimated matrix), as well as $k \in \{3, 4, 5\}$ for a total of 150 different parameter settings. We then choose λ and k based on the matrix completion heuristic outlined in Subsection 5.4.2, where F is set to be the Gaussian distribution with constant variance. Importantly, this results in choosing $k = 3$, which is used when fitting the SVD embedding used for downstream analysis in Subsection 5.7.2. (Note that this is different from $k = 5$ used for the eSVD embedding shown in Subsection 5.7.3.)

Details for Figure 5.3. Figure A in Figure 5.3 is created by computing the logarithm of the column-wise (i.e., gene-wise) mean and standard deviation of A, the preprocessed single-cell RNA-seq dataset. The color of the point is based on the ANOVA p-value that tests if values in each column of A(i.e., the expression of each gene) is equal across all 6 major cell types shown in Figure 5.1, with blacker points denoting a p-value closer to 0 and more yellow points denoting p-values closer to 1.

Figure B in Figure 5.3 is created by first computing the first principal component of A (i.e., the leading eigenvector of the empirical covariance matrix of A), setting all the negative entries to 0, and then renormalizing all the entries of the resulting vector to sum to 1. We set all the negative values to 0 so the resulting vector can meaningfully represented a weighted average. We then compute the inner product between the resulting vector and each row of A, and then plot a violin plot based on the grouping the resulting inner product by the six major cell types shown in in Figure 5.1.

5.C Discussion on estimator

5.C.1 Initialization method

We first define notation needed to describe the initialization method, inspired by Wang et al. (2016). For a given one-parameter exponential family distribution, let $g^{-1}(\cdot)$ be the inverse function of the $g(\cdot)$, the log-partition function for F, which is guaranteed to exist by the convexity of $g(\cdot)$. Furthermore, for a generic matrix $\Theta \in \mathbb{R}^{n\times p}$ that is rank k, let $\mathcal{L}_n(\Theta)$ be equivalent to the loss function $\mathcal{L}_n(X, Y)$ defined in (5.6) for any $X \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{p\times k}$ such that $\Theta = XY^\top$. Lastly, define $\Pi_k(\cdot)$ be the projection operator (based on alternating between truncating the singular values and thresholding) to project a given matrix into the set

$$\Big\{\Theta \in \mathbb{R}^{n\times p} \ : \ \text{rank}(\Theta) \le k, \quad \text{and} \quad \theta_{ij} \in \mathbb{R} \quad \forall (i,j)\Big\}. \tag{5.20}$$

We initialize our estimate, $\widehat{\Theta}^{(0)}$ to be $g^{-1}(A)$, where the function $g^{-1}(\cdot)$ is applied entrywise. Afterwards, we perform projected gradient descent. That is, for $t = 0, \ldots, T' - 1$ iterations, for a stepsize $\gamma > 0$, we iterate,

$$\widehat{\Theta}^{(t+1)} = \Pi_k\Big(\widehat{\Theta}^{(t)} - \gamma \nabla \mathcal{L}_n(\widehat{\Theta}^{(t)})\Big).$$

Let $\widehat{\Theta}^{(T')} = \widehat{\Theta}'$ defined in Subsection 5.4.1.

Determining γ. We found that the initialization without any projected gradient steps works well in practice. However, in our implementation, we use only a few project gradient steps where within each iteration, γ is selected within each iteration via binary search to be the largest value such that the objective function $\mathcal{L}_n$ decreases. There is no theoretical guarantee for such a heuristic however. We hope to provide a more concrete initialization procedure in the future that is well supported by theory.

Lack of convergence. Unfortunately, in our experiments, we have found that there are instances where the above projection (5.20) does not converge empirically. While there is a rich body of literature studying the intersection of many convex sets (see Kundu et al. (2017) and Tibshirani (2017) and the references within), the set of all rank-k matrices is not convex. In instances where this occurs, we terminate the above initialization procedure, and instead fit a k-block model to $g^{-1}(\Theta)$ based applying k-means to both the first k left and right singular vectors separately. This procedure is reminiscent of those used to fit stochastic block models described in various papers such as Li et al. (2016). We hope to investigate a more principled initialization scheme in future work, especially one that integrates well with our theoretical results.

We do use non-negative matrix factorization (NMF) of $g^{-1}(\Theta)$ since the computational complexity of most non-negative matrix factorization methods is of a similar order to our alternating minimization descent. See Subsection 5.C.4 for a more detailed discussion of how NMF compares to eSVD.

5.C.2 Identifiability condition

Here, the following proposition shows that the first part of Assumption 7 can be interpreted as an identifiability condition, since it can always be satisfied after performing an appropriate linear transformation.

Proposition 37. *Given two k-dimensional distributions G and H each with at least two moments where the population second moment matrices are full rank, consider two independent random variables $X' \sim G$ and $Y' \sim H$. Then there exists a linear and invertible transformation R such that the population second moment matrices of $X = RX'$ and $Y = R^{-\top}Y'$ are the same, i.e.,*

$$\mathbb{E}[XX^\top] = \mathbb{E}[YY^\top].$$

Furthermore, both population second moment matrices above are diagonal matrices.

The proof is in Appendix 5.H. Note that since R is invertible, we guarantee that the random vectors X and Y preserve the distribution of their inner product, i.e.,

$$\mathbb{P}\big((X')^\top Y' \leq t\big) = \mathbb{P}\big(X^\top Y \leq t\big), \quad \forall t \in \mathcal{R}.$$

Hence, Assumption 7 can be interpreted as an identifiability condition, since we can only estimate G and H only up to this transformation that ensures their population second moment matrices match.

5.C.3 Usage for common one-parameter exponential-family distributions

eSVD can be extended to any one-parameter exponential family distribution, so here, we derive all the necessary ingredients (calculation of the objective function and gradient with respect to X and Y) for most common one-parameter exponential family distributions. We explain the pipeline to derive how to fit a given one-parameter exponential family distribution F into the eSVD framework.

1. **Writing distribution in exponential-family form**: With a given one-parameter exponential distribution F in mind, write the probability density function (or probability mass function) in the form described in (5.4). That is, determine the functions $g(\cdot)$ (the log-partition function for F), $\eta(\cdot)$ (the natural parameter function) and $T(\cdot)$ (the sufficient statistic function) such that

$$p(A_{ij} \mid \theta_{ij}) = h(A_{ij}) \exp\big(\eta(A_{ij})^\top T(\theta_{ij}) - g(\theta_{ij})\big).$$

2. **Determine the domain**: Next, based on the log-partition function $g(\cdot)$, determine its domain $\mathcal{R}$.

3. **Determining the objective function**: Next, with the functions $g(\cdot)$, $\eta(\cdot)$ and $T(\cdot)$ explicitly derived, plug them into the objective function,

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \Big[g(X_i^\top Y_j) - \eta(A_{ij})^\top T(X_i^\top Y_j)\Big].$$

4. **Calculating the gradients**: Lastly, derive the gradient of the above objective function with respect to X and Y.

Using the above pipeline, we provide the derivations for the following five distributions as an example for readers. We only write down the gradients with respect to X, as the gradients with respect to Y are similar.

- **Gaussian**: For a Gaussian with known variance σ^2, the density is

$$p(A_{ij} \mid \theta_{ij}) = \Big[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(\frac{-A_{ij}^2}{2\sigma^2}\Big)\Big] \cdot \exp\Big(\Big[A_{ij}/\sigma^2\Big]^\top \Big[\theta_{ij}\Big] - \frac{\theta_{ij}^2}{2\sigma^2}\Big),$$

where $\theta_{ij} = \mu_{ij}$, and the domain of θ_{ij} is $\mathcal{R} = \mathbb{R}$. Hence, the objective function becomes

$$\mathcal{L}_n(X, Y) = \frac{1}{np} \sum_{(i,j)} \Big[\underbrace{\frac{(X_i^\top Y_j)^2}{2\sigma^2}}_{=a^2} - \underbrace{\Big[A_{ij}/\sigma^2\Big]^\top \Big[X_i^\top Y_j\Big]}_{=2ab}\Big].$$

However, we can complete the square to further simplify this expression. Adding the term $A_{ij}^2/(2\sigma^2)$ into the summation yields a new objective function,

$$\mathcal{L}_n(X, Y) \propto \frac{1}{np} \sum_{(i,j)} \Big[\Big(\frac{(X_i^\top Y_j)}{\sqrt{2}\sigma} - \frac{A_{ij}}{\sqrt{2}\sigma}\Big)^2\Big] \propto \frac{1}{np} \sum_{(i,j)} \Big[\Big((X_i^\top Y_j) - A_{ij}\Big)^2\Big].$$

Then the gradient with respect to X is a $n \times k$ matrix where the ith row is

$$\nabla_{X_i} \mathcal{L}_n(X, Y) \propto \frac{1}{np} \sum_{j=1}^{p} \Big[2\Big((X_i^\top Y_j) - A_{ij}\Big) \cdot Y_j\Big].$$

- **Curved Gaussian**: We do the full derivation. For a curved Gaussian with parameter τ (i.e., $N(\mu_{ij}, \mu_{ij}^2/\tau^2)$),

$$\begin{aligned}
p(A_{ij} \mid \mu_{ij}) &= \frac{\tau}{\sqrt{2\pi\mu_{ij}^2}} \cdot \exp\Big(-\frac{(A_{ij} - \mu_{ij})^2}{2\mu_{ij}^2/\tau^2}\Big) = \frac{\tau}{\sqrt{2\pi\mu_{ij}^2}} \cdot \exp\Big(-\frac{A_{ij}^2 - 2A_{ij}\mu_{ij} + \mu_{ij}^2}{2\mu_{ij}^2/\tau^2}\Big) \\
&= \frac{\tau}{\sqrt{2\pi\mu_{ij}^2}} \cdot \exp\Big(\frac{-\tau^2 A_{ij}^2}{2\mu_{ij}^2} + \frac{\tau^2 A_{ij}}{\mu_{ij}} - \frac{\tau^2}{2}\Big) \\
&= \Big[\frac{\tau \exp(-\tau^2/2)}{\sqrt{2\pi}}\Big] \cdot \exp\Big(\frac{-\tau^2 A_{ij}^2}{2} \cdot \frac{1}{\mu_{ij}^2} + \tau^2 A_{ij} \cdot \frac{1}{\mu_{ij}} - \log(\mu_{ij})\Big)
\end{aligned}$$

Hence, replacing $\theta_{ij} = -1/\mu_{ij}$, we obtain

$$p(A_{ij} \mid \theta_{ij}) = \Big[\frac{\tau \exp(-\tau^2/2)}{\sqrt{2\pi}}\Big] \cdot \exp\Big(\begin{bmatrix} -\tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{bmatrix}^\top \begin{bmatrix} \theta_{ij} \\ \theta_{ij}^2 \end{bmatrix} - \big(-\log(-\theta_{ij})\big)\Big),$$

where the domain of θ_{ij} is $\mathcal{R} = (-\infty, 0)$. Hence, the objective function becomes

$$\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{(i,j)}\Big[-\log\big(-X_i^\top Y_j\big) - \begin{bmatrix} -\tau^2 A_{ij} \\ -\tau^2 A_{ij}^2/2 \end{bmatrix}^\top \begin{bmatrix} X_i^\top Y_j \\ (X_i^\top Y_j)^2 \end{bmatrix}\Big].$$

Then the gradient with respect to X is a $n \times k$ matrix where the ith row is

$$\nabla_{X_i}\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{j=1}^{p}\Big[\Big(-\frac{1}{X_i^\top Y_j} + \tau^2 A_{ij} + \tau^2 A_{ij}^2 (X_i^\top Y_j)\Big)\cdot Y_j\Big].$$

- **Exponential**: The density is

$$p(A_{ij} \mid \theta_{ij}) = \Big[1\Big]\cdot\exp\Big(\Big[A_{ij}\Big]^\top\Big[\theta_{ij}\Big] - \big(-\log(-\theta_{ij})\big)\Big),$$

where $\theta_{ij} = -\lambda_{ij}$ (meaning $\mathbb{E}[A_{ij}] = -1/\theta_{ij}$, and the domain of θ_{ij} is $\mathcal{R} = (-\infty, 0)$. Hence, the objective function becomes

$$\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{(i,j)}\Big[-\log(-X_i^\top Y_j) - \Big[A_{ij}\Big]^\top\Big[X_i^\top Y_j\Big]\Big].$$

Then the gradient with respect to X is a $n \times k$ matrix where the ith row is

$$\nabla_{X_i}\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{j=1}^{p}\Big[\Big(\frac{-1}{X_i^\top Y_j} - A_{ij}\Big)\cdot Y_j\Big]$$

- **Poisson**: The density is

$$p(A_{ij} \mid \theta_{ij}) = \Big[\frac{1}{A_{ij}!}\Big]\cdot\exp\Big(\Big[A_{ij}\Big]^\top\Big[\theta_{ij}\Big] - \exp\big(\theta_{ij}\big)\Big),$$

where $\theta_{ij} = \log(\lambda_{ij})$ (meaning $\mathbb{E}[A_{ij}] = \exp(\theta_{ij})$), and the domain of θ_{ij} is $\mathcal{R} = (0, \infty)$. Hence, the objective function becomes

$$\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{(i,j)}\Big[\exp(X_i^\top Y_j) - \Big[A_{ij}\Big]^\top\Big[X_i^\top Y_j\Big]\Big].$$

Then the gradient with respect to X is a $n \times k$ matrix where the ith row is

$$\nabla_{X_i}\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{j=1}^{p}\Big[\Big(\exp(X_i^\top Y_j) - A_{ij}\Big)\cdot Y_j\Big]$$

- **Negative Binomial**: The negative binomial represents the number of successes before a specified number of failures r occurs. (This is different from the binomial distribution which represents the number of success among a fixed number of trials.) For a fixed number of failures r,

$$p(A_{ij} \mid \theta_{ij}) = \Big[\binom{A_{ij}+r-1}{A_{ij}}\Big] \cdot \exp\Big(\Big[A_{ij}\Big]^\top \Big[\theta_{ij}\Big] - \big(-r\log(1-\exp(\theta_{ij}))\big)\Big),$$

where $\theta_{ij} = \log(p_{ij})$ (meaning $\mathbb{E}[A_{ij}] = r\exp(\theta_{ij})/(1-\exp(\theta_{ij}))$, and the domain of θ_{ij} is $\mathcal{R} = (-\infty, 0)$. Hence, the objective function becomes

$$\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{(i,j)}\Big[\big(-r\log(1-\exp(X_i^\top Y_j))\big) - \Big[A_{ij}\Big]^\top \Big[X_i^\top Y_j\Big]\Big].$$

Then the gradient with respect to X is a $n \times k$ matrix where the ith row is

$$\nabla_{X_i}\mathcal{L}_n(X,Y) = \frac{1}{np}\sum_{j=1}^{p}\Big[\Big(\frac{r\exp(X_i^\top Y_j)}{1-\exp(X_i^\top Y_j)} - A_{ij}\Big)\cdot Y_j\Big]$$

5.C.4 Additional comparison of eSVD to estimators in the literature

We discuss additional nuances for eSVD that makes it different from other methods in the literature that were not already discussed in §5.3.

Comparison to alternating gradient descent. The majority of theoretical work that investigate nonconvex estimators to perform matrix factorization use alternating projected gradient descent to refine the initial estimate (see Wang et al. (2016), Yu et al. (2020), and Chi et al. (2019)) where each iteration updates the current estimates with a gradient step instead. This is in contrast with our choice of using alternating constrained minimization in eSVD. While we have found alternating projected gradient descent is more amendable for theoretical analysis, in practice, we have found alternating projected gradient descent more numerically unstable due to its sensitivity to the chosen step-sizes. Hence, we found alternating projected gradient descent harder to tune compared to the alternating minimization approach.

Comparison to low-rank covariance matrix estimation for exponential families. As mentioned in §5.3, Liu et al. (2018b) and Zhang et al. (2018) estimate a low-rank covariance matrix where each entry in the observed matrix A is drawn from an exponential-family distribution. While this task is non-trivial due to the possible dependency between the mean and covariance (which prevents naively centering the variables around 0), this is fundamentally a different task than the one posited in this paper in two ways. First, eSVD estimates a low-rank matrix of natural parameters. Second, estimating a low-rank covariance matrix does not immediately suggest a non-linear embedding procedure, unlike eSVD's goal of generalizing the SVD to exponential-family distributions.

Comparison to non-negative matrix factorization. There is an extremely rich literature on non-negative matrix factorization (NMF, see Donoho and Stodden (2004), Arora et al. (2016), Gillis (2017) and references within), and upon first glance, it might seem like eSVD similar to NMF. However, there are two importance distinctions. First, the key difference between eSVD and NMF can be seen in Assumption 1. Specifically, eSVD only assumes that the inner products between X and Y lie in $\mathcal{R}$, such as the positive half-line. This is in contrast to NMF, where either each entry in X or Y (or both) lie on the positive half-line. Intuitively, this means eSVD's task is easier than NMF's task (loosely speaking), since ensuring that each entry in X and Y lying on the positive half-line implies that the inner product between each pair of rows in X and Y lies on the positive half-line. Second, eSVD's identifiability assumptions are much easier than NMF's identifiability assumptions. eSVD's identifiability assumptions are outlined in Proposition 37, which shows that there exists a transformation such that the population second moment matrix for X and Y are equal. NMF's identifiability conditions are much more nuances, and leads to concepts such as simplicial cones, separability and anchor words as discussed in Donoho and Stodden (2004) and Arora et al. (2012).

5.D Application of propositions to the curved Gaussian model

In this section, we detail the assumptions needed to imply Assumptions 2-3 and the modifications to eSVD and Assumption 4 in order to apply Proposition 35 to the curved Gaussian model (5.5). This entails introducing assumptions to help determine the values of μ, L, and S used in Assumptions 2 and 3. We introduce some notation beforehand. First, observe that we can rescale the normalization constants in (5.7) and (5.9). Hence, we redefine the sample loss functions to be

$$\mathcal{L}_n^{(X)}(X,Y) = \frac{1}{p}\sum_{(i,j)}\Big[-\log(-X_i^\top Y_j) - \begin{bmatrix}\tau^2 A_{ij}\\ -\tau^2 A_{ij}^2/2\end{bmatrix}^\top \begin{bmatrix}-X_i^\top Y_j\\ (X_i^\top Y_j)^2\end{bmatrix}\Big], \tag{5.21}$$

$$\mathcal{L}_n^{(Y)}(X,Y) = \frac{1}{n}\sum_{(i,j)}\Big[-\log(-X_i^\top Y_j) - \begin{bmatrix}\tau^2 A_{ij}\\ -\tau^2 A_{ij}^2/2\end{bmatrix}^\top \begin{bmatrix}-X_i^\top Y_j\\ (X_i^\top Y_j)^2\end{bmatrix}\Big], \tag{5.22}$$

Their corresponding population loss functions that we analyze in this section are

$$\mathcal{L}^{(X)}(X,Y) = \frac{1}{p}\sum_{(i,j)}\Big[-\log(-X_i^\top Y_j) - \begin{bmatrix}\tau^2 \mathbb{E}[A_{ij}]\\ -\tau^2 \mathbb{E}[A_{ij}^2]/2\end{bmatrix}^\top \begin{bmatrix}-X_i^\top Y_j\\ (X_i^\top Y_j)^2\end{bmatrix}\Big], \tag{5.23}$$

$$\mathcal{L}^{(Y)}(X,Y) = \frac{1}{n}\sum_{(i,j)}\Big[-\log(-X_i^\top Y_j) - \begin{bmatrix}\tau^2 \mathbb{E}[A_{ij}]\\ -\tau^2 \mathbb{E}[A_{ij}^2]/2\end{bmatrix}^\top \begin{bmatrix}-X_i^\top Y_j\\ (X_i^\top Y_j)^2\end{bmatrix}\Big], \tag{5.24}$$

where A_{ij} follows the distribution (5.5). The change from $1/(np)$ in (5.7) and (5.9) to $1/p$ and $1/n$ is for simplicity and facilitates to control the spectrum of the Hessian appropriately. We define the minimization operators we will use in this section as

$$\begin{aligned}
M_n^X(\overline{Y}) &= \operatorname*{argmin}_{X\in\mathbb{R}^{n\times k}} \mathcal{L}_n(X,\overline{Y}) \ : \ X_i^\top \overline{Y}_j \in \mathcal{R}, \quad \forall (i,j) \\
M_n^Y(\overline{X}) &= \operatorname*{argmin}_{Y\in\mathbb{R}^{p\times k}} \mathcal{L}_n(\overline{X},Y) \ : \ (\overline{X}_i)^\top Y_j \in \mathcal{R}, \quad \forall (i,j) \\
M^X(\overline{Y}) &= \operatorname*{argmin}_{X\in\mathbb{R}^{n\times k}} \mathcal{L}(X,\overline{Y}) \ : \ X_i^\top \overline{Y}_j \in \mathcal{R}, \quad \forall (i,j) \\
M^Y(\overline{X}) &= \operatorname*{argmin}_{Y\in\mathbb{R}^{p\times k}} \mathcal{L}(\overline{X},Y) \ : \ (\overline{X}_i)^\top Y_j \in \mathcal{R}, \quad \forall (i,j).
\end{aligned}$$

While we state results for both $\mathcal{L}^{(X)}(\cdot,\cdot)$ and $\mathcal{L}^{(Y)}(\cdot,\cdot)$ in this section, we prove only statements for $\mathcal{L}^{(X)}(\cdot,\cdot)$ since the proofs for both loss functions are identical.

Assumptions. The following three assumptions are needed to analyze the curved Gaussian setting.

Assumption 8 (Refinement of domain)**.** *Assume for the curved Gaussian distribution* (5.5)*, let* $\mathcal{R} = [1/r, r]$ *defined in Assumption 1, where* $r > 1$*.*

Assumption 9 (Inner product error)**.** *Conditioned on* X *and* Y*, assume there exists* $\{X^*, \overline{Y}^*\} \in \{\mathcal{X}^*, \overline{\mathcal{Y}}^*\}$ *such that the initialization* $\overline{Y}^{(0)}$ *satisfies for all* $(i,j) \in \{1,\dots,n\} \times \{1,\dots,p\}$

$$|(X_i^*)^\top(\overline{Y}_j^{(0)} - \overline{Y}_j^*)| \le c \cdot \frac{1}{\log(\min\{n,p\})} |(X_i^*)^\top \overline{Y}_j^*|. \tag{5.25}$$

In addition, conditioned on X *and* Y*, assume for each iterations* $t = 1,\dots,T$ *throughout the algorithm, there exists* $\{X^*, \overline{Y}^*\} \in \{\mathcal{X}^*, \overline{\mathcal{Y}}^*\}$ *and* $\{\overline{X}^*, Y^*\} \in \{\overline{\mathcal{X}}^*, Y^*\}$ *where* $\Theta = X^*(\overline{Y}^*)^\top = \overline{X}^*(Y^*)^\top$ *such that for all* $(i,j) \in \{1,\dots,n\} \times \{1,\dots,p\}$

$$|(X_i^*)^\top(\overline{Y}_j^{(t)} - \overline{Y}_j^*)| \le c \cdot \frac{1}{\log(\min\{n,p\})} |(X_i^*)^\top \overline{Y}_j^*|, \tag{5.26}$$

$$|(\overline{X}_i^{(t)} - \overline{X}_i^*)^\top Y_j^*| \le c \cdot \frac{1}{\log(\min\{n,p\})} |(\overline{X}_i^*)^\top Y_j^*|. \tag{5.27}$$

Assumption 8 effectively ensures $\mathcal{R}$ is bounded away from 0, a condition to ensure the Hessian is well-controlled. Assumption 9 is an assumption that is similar to the incoherence assumption in Ma et al. (2018) and Chi et al. (2019). There, the authors prove that spectral initialization ensures the requirement analogous to (5.25) is met with high probability, and each iteration retains properties analogous to (5.26) and (5.27).

Assumption 10 (Fixed statistical error)**.** *Conditioned on X and Y, for any matrices $X' \in \mathbb{R}^{n\times k}$ or $Y' \in \mathbb{R}^{p\times k}$, with probability at least $1 - c/\min\{n,p\}$ for some universal constant c that*

$$\max\Big\{\|M^X(Y') - M_n^X(Y')\|_F,\ \|M^Y(X') - M_n^Y(X')\|_F\Big\} \le \varepsilon_{\textit{fixed}}(n,p),$$

where $\varepsilon_{\textit{fixed}}(n,p)$ is some function of n and p (and possibly other quantities). Assume that

$$\varepsilon_{\textit{fixed}}(n,p) \le \frac{\widetilde{d}_k}{4\max\{n,p\}^{1/2}}.$$

As mentioned in §5.5.1, Assumption 10 is different from Assumption 4 but enables a simpler analysis. The reason we impose this new assumption is that minimizing objective functions such as (5.21) does not have a closed-form solution, so uniformly bounding the error is difficult for the curved Gaussian model. By imposing Assumption 10 instead of Assumption 4, eSVD now requires resampling, i.e., a fresh batch of samples every iteration, for all T iterations. While this yields a different algorithm that is not practical to use, we believe the theoretical properties we prove also roughly hold for the curved Gaussian model without resampling.

Controlling μ, L, S and $\varepsilon_{\textit{fixed}}(n,p)$. The following lemma implies that $\mathcal{L}^{(X)}(\cdot,\cdot)$ is $(2+\tau^2)/r^2$-strongly convex and its gradient is $(2+\tau^2)r^2$-Lipschitz.

Lemma 38 (Spectrum of Hessian)**.** *For the loss function* (5.23) *and* (5.24) *where A_{ij} follows the distribution* (5.5)*, under Assumption 8, the eigenvalues of the Hessian $\nabla_X^2\mathcal{L}^{(X)}(X,\overline{Y})$ and $\nabla_Y^2\mathcal{L}^{(Y)}(\overline{X},Y)$ are bounded between $(2+\tau^2)/r^2$ and $(2+\tau^2)r^2$.*

The following lemma analyzes the gradient smoothness of $\mathcal{L}^{(X)}(\cdot,\cdot)$ with respect to the alternating variable.

Lemma 39 (Gradient smoothness with respect to alternating variable)**.** *Conditioned on X and Y, for the loss function* (5.23) *and* (5.24) *where A_{ij} follows the distribution* (5.5)*, under Assumptions 8 and 9, for $\min\{n,p\}$ large enough, for each iterations $t = 0,1,\ldots,T$, any pairs $\{X^*,\overline{Y}^*\} \in \{\mathcal{X}^*,\overline{\mathcal{Y}}^*\}$ and $\{\overline{X}^*,Y^*\} \in \{\overline{\mathcal{X}}^*,\mathcal{Y}^*\}$ satisfy*

$$\|\nabla_X\mathcal{L}^{(X)}(X^*,\overline{Y}^*) - \nabla_X\mathcal{L}^{(X)}(X^*,\overline{Y}^{(t)})\|_F \le c\tau^2r^2\cdot\Big(\frac{(np)^{1/2}}{p\cdot\log(\min\{n,p\})} + \frac{\widetilde{d}_1}{p}\Big)\|\overline{Y}^{(t)} - \overline{Y}^*\|_F,$$

and

$$\|\nabla_Y\mathcal{L}^{(Y)}(\overline{X}^*,Y^*) - \nabla_Y\mathcal{L}^{(Y)}(\overline{X}^{(t)},Y^*)\|_F \le c\tau^2r^2\cdot\Big(\frac{(np)^{1/2}}{p\cdot\log(\min\{n,p\})} + \frac{\widetilde{d}_1}{p}\Big)\|\overline{X}^{(t)} - \overline{X}^*\|_F,$$

for some universal constant c.

We now analyze the difference between the minimizers in (5.23) and (5.21) (or between (5.24) and (5.22)).

Lemma 40. *Conditioned on X and Y, let A_{ij} follow the curved Gaussian distribution* (5.5)*. For a fixed $\overline{Y}$, under the assumptions in Lemma 38, with probability at least $1-6/p$,*

$$\frac{1}{n}\|M^X(\overline{Y}) - M_n^X(\overline{Y})\|_F \le c\Big(\frac{\log^{1/4}(np)}{n^{1/2}p^{1/4}}\Big),$$

where c is a constant that depends only on k, τ and r. Similarly, for a fixed $\overline{X}$, under Assumption 8, with probability at least $1-6/n$,

$$\frac{1}{p}\|M^Y(\overline{X}) - M_n^Y(\overline{X})\|_F \le c\Big(\frac{\log^{1/4}(np)}{n^{1/4}p^{1/2}}\Big),$$

The above corollary means that

$$\varepsilon_{\text{fixed}}(n,p) = c\Big(\max\Big\{\frac{n^{1/2}}{p^{1/4}}, \frac{p^{1/2}}{n^{1/4}}\Big\}\cdot \log^{1/4}(np)\Big),$$

where c is a constant that depends only on k, μ, L, τ and r.

Corollary 41 (Application of Proposition 35 to curved Gaussian model)**.** *Assume the curved Gaussian model* (5.5) *where k, τ, α, β and r are constant, and Assumptions 5 and 9-10 hold conditioned on X and Y, and*

$$\frac{4(np)^{1/2}}{\widetilde{d}_k}\frac{S}{\mu} < 1.$$

In addition, assume that there exists a universal constant c' such that

$$\widetilde{d}_k > (4+c')\cdot(np)^{1/2},$$

for large enough n and p. Conditioned on X and Y, for the number of iterations T large enough and $n = O(p)$ and $p = O(n)$, eSVD described in (5.7)-(5.10) *where each iteration resamples the observation matrix A achieves the rate*

$$\|\Theta - \widehat{\Theta}\|_F = O_P\Big(n^{3/4}\log^{1/4}(n)\Big).$$

The following corollary is proved immediately after combining Corollary 41 with Proposition 36.

Corollary 42 (Application of Proposition 36 to curved Gaussian model)**.** *Assume all the setting and assumptions in Corollary 41. Then, eSVD after reparamterizations* (5.11) *and* (5.12) *achieves the rate*

$$\frac{1}{n}\|X - \widehat{X}\|_F^2 = O_P\Big(\frac{\log^{1/2}(n)}{n^{1/2}}\Big).$$

5.E Additional simulation details/results

Throughout this section, we use the following notational scheme to parameterize different distributions. The Negative Binomial and Bernoulli distributions are parameterized by Negative Binomial(r, p) and Bernoulli(p) respectively where r is the number of failures and p is the probability of success. The Gamma distribution is parameterized by Gamma(a, b) where a and b are the shape and rate parameters respectively. The Poisson distribution is parameterized by Poisson(λ) where the mean is λ.

5.E.1 Simulation setup

Generation of natural parameters. To sample from G (prior to identifiability conditions), we uniformly sample an equal number of points along 4 connected line segments, where the line segments collectively have endpoints at $\{(4, 10), (25, 100), (60, 80), (40, 10), (100, 25)\}$ in the Cartesian coordinate system. We then add Gaussian noise with $\sigma = 0.05$ to each of the points. This generates $X_1, \ldots, X_n$ (prior to identifiability conditions).

To sample from H (prior to identifiability conditions), we similarly uniformly sample an equal number of points along 2 disconnected line segments. One line segment goes from $(1, 4.5)$ to $(1.25, 5)$ while the other goes from $(4.5, 1)$ to $(5, 1.25)$ in the Cartesian coordinate system. We then also add Gaussian noise with $\sigma = 5$ to each of the points. This generates $Y_1, \ldots, Y_p$ (prior to identifiability conditions).

We then compute Θ where $\theta_{ij} = X_i^\top Y_j$. Our chosen noise level ensures (with high probability) that all the entries will be positive. Letting the SVD of this matrix be $\Theta = UDV^\top$, we then output the target embedding we wish to estimate, $X = (n/p)^{1/4} \cdot U \cdot \sqrt{D}$ and $Y = (p/n)^{1/4} \cdot V \cdot \sqrt{D}$, where the square root function is interpreted to be entry-wise.

Other methods. While SVD, UMAP and t-SNE are more common methods in statistical and genomic analyses, ZINB-WaVE and pCMF are methods more specific to single-cell analyses that we briefly overview here.

- **ZINB-WaVE**: ZINB-WaVE relies on the Negative Binomial distribution. Similar to our model, X_i represents the fixed lower-dimensional latent vector for each cell, and Y_j and W_j represent two sets of fixed lower-dimensional latent vectors for each gene. Let π_j denote the a parameter for gene j. The generative model is, for any $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, p\}$,

$$\begin{aligned} Z_{ij} &\sim \text{Negative Binomial}\big(\pi_j, \frac{\exp(X_i^\top Y_j)}{\exp(X_i^\top Y_j) + \pi_j}\big), \\ D_{ij} &\sim \text{Bernoulli}\big(1/(1 + \exp(-X_i^\top W_j))\big) \\ A_{ij} &= Z_{ij} \cdot D_{ij}, \end{aligned} \tag{5.28}$$

where all the latent variables are independent of one another and A_{ij}'s are all conditionally independent[4]. ZINB-WaVE estimates the parameters X, Y and W via an alternating minimization scheme based on ridge-regression. We note that the ZINB-WaVE model is able to handle additional covariate information about each cell or gene.

- **pCMF**: pCMF relies on the Poisson distribution. Similar to our model, X_i represents the lower-dimensional latent vector for each cell, and Y_j represents the lower-dimensional latent vector for each gene, but explicit distributions for G and H are used to facilitate a Bayesian approach. Let π_j denote the unknown gene-specific dropout rate. The generative model is,

$$\begin{aligned}
X_{i\ell} &\sim \text{Gamma}(\alpha_{\ell,1}, \alpha_{\ell,2}), \quad \text{for } (i,\ell) \in \{1,\dots,n\} \times \{1,\dots,k\} \\
Y_{j\ell} &\sim \text{Gamma}(\beta_{\ell,1}, \beta_{\ell,2}), \quad \text{for } (i,\ell) \in \{1,\dots,n\} \times \{1,\dots,k\} \\
Z_{ij} &\sim \text{Poisson}(X_i^\top Y_j), \quad \text{for } (i,j) \in \{1,\dots,n\} \times \{1,\dots,p\} \\
D_{ij} &\sim \text{Bernoulli}(\pi_j), \quad \text{for } (i,j) \in \{1,\dots,n\} \times \{1,\dots,p\} \\
A_{ij} &= Z_{ij} \cdot D_{ij},
\end{aligned}$$

where all the latent variables are independent of one another and A_{ij}'s are all conditionally independent. pCMF estimates the parameters via a variational EM algorithm.

We list the `R` packages used for comparisons with other methods. We use the `zinbwave` package for ZINB-WaVE, and set the `K` parameter to 2, the `maxiter.optimize` parameter to 100 and the `normalizedValues` parameter to `False`. We use the `pCMF` package for pCMF and set the `K` parameter to 2 and the `sparsity` parameter to `False`. For UMAP and t-SNE, we tune the methods' respective parameters in an oracle-fashion in order to maximize the relative embedding correlation described in §5.6. That is, this tuning requires knowing the true embedding, which is unrealistic in practice but demonstrates the performance of these methods under the most favorable conditions. We use the `Rtsne` package for t-SNE, and set the k parameter to 2 and tune the `perplexity` parameter in an oracle fashion among 10 values between 2 and 50. We use the `umap` package for UMAP, and set the `n_components` parameter to 2 as well as `init` to `random`. Additionally, we tune the `n_neighbors` and `min_dist` parameters in an oracle fashion among the values $\{2, 3, 5, 15, 30, 50\}$ and $\{10^{-5}, 10^{-3}, 0.1, 0.3, 0.5, 0.9\}$ respectively (for a total of 36 different parameter settings).

[4]Their paper actually parameterizes the Negative Binomial distribution by its mean and inverse dispersion parameter, but is equivalent to the one we describe.

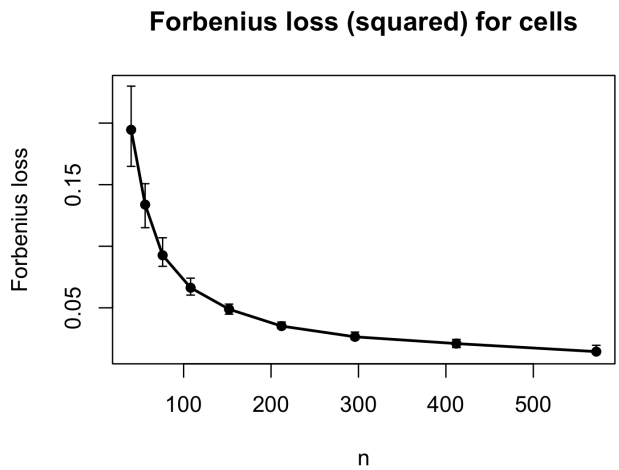


Figure 5.10: $\|X - \widehat{X}\|_F^2/n$ verses n, where the solid points represent thxe median performance over 200 trials and the error bars represent the 25th to 75th quantile.

5.E.2 Verification of convergence to G

Based on the first simulation suite described in §5.6, we plot the empirical performance of eSVD when loss function is set to the likelihood of the curved Gaussian distribution with $\tau = 2$, which is correctly-specified model. We plot $\|X - \widehat{X}\|_F^2/n$ verses n in Figure 5.10.

5.E.3 Simulation under misspecified model

We present the additional simulations alluded to as the third simulation suite in §5.6. In this simulation suite, we build on top of the negative binomial model shown in (5.19), but include additional complexities that make the simulation more realistic according to the generative model for ZINB-WaVE shown in (5.28). Specifically, compared to the negative binomial model shown in (5.19), we vary the dispersion parameter so 25% of the genes have a dispersion parameter of 80, another 25% have a parameter of 120, and the remaining 50% have a parameter of 600. Additionally, a dropout term is now included based on a logistic model. Both of these changes ensure that the negative binomial model that eSVD fits is sufficiently misspecified. As our simulations reassuringly shows though, eSVD still estimates the embedding relatively well when compared to other methods aside from ZINB-WaVE itself.

When we fit eSVD via a negative binomial model, we use the matrix-completion diagnostic mentioned in Subsection 5.4.2 to search for a global dispersion parameter of $\pi \in \{50, 100, 500, 1000\}$ and set $k = 3$. Notice that we are using *one* dispersion parameter

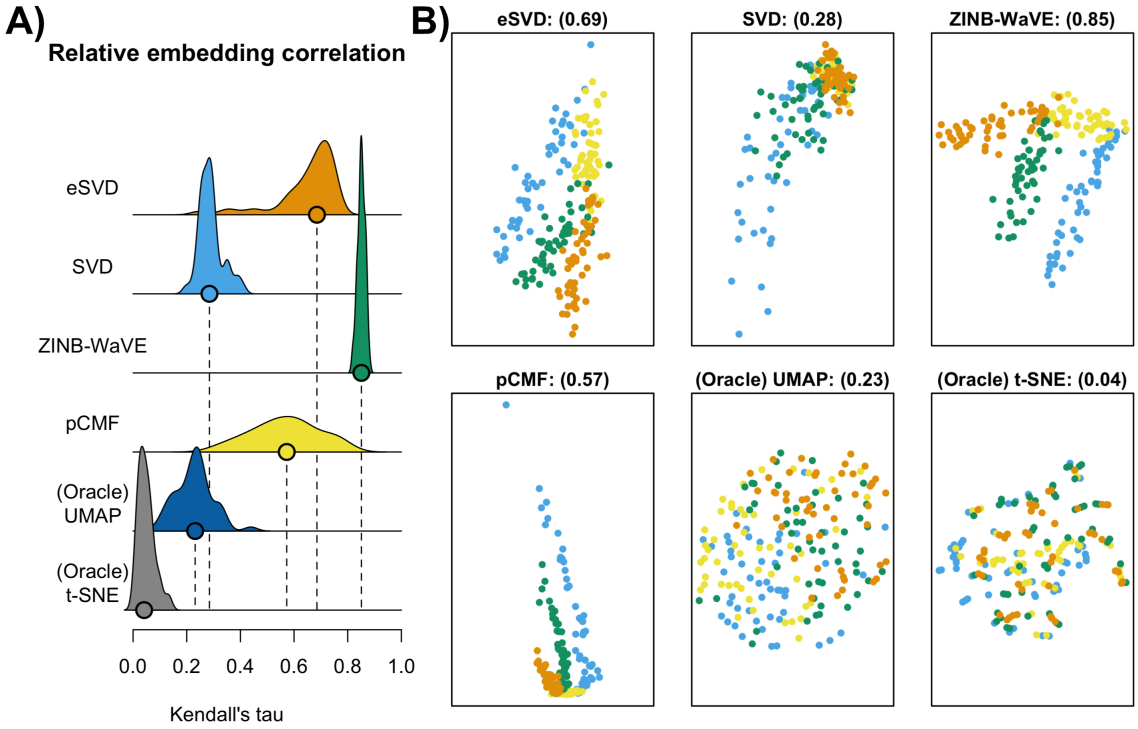


Figure 5.11: Results for the misspecified model simulation suite, where the plots are comparable to those in Figure 5.5.

to model the simulated dataset, although our true generative model uses three different dispersion parameters. Also, we increase the latent space to $k = 3$ since we found empirically, the addition of the dropout factor can be reasonably captured by an extra latent dimension.

We demonstrate our results in Figure 5.11. We see that ZINB-WaVE performs the best according to the relative embedding correlation metric, but this is unsurprising as our generative model is correctly specified for ZINB-WaVE. However, even though it is misspecified for eSVD using the negative binomial distribution, eSVD's performance is still quite good. The remaining methods all do not perform well in comparison. Hence, we believe that eSVD using the negative binomial distribution, with appropriate tuning based on the matrix-completion diagnostic, is comparable in performance to ZINB-WaVE in practice. As we mentioned in the main text however, the benefits of using eSVD is that eSVD has a solid theoretical foundation, can be easily extended to other one-parameter exponential-family distributions, and can handle missing values to assess model fit. ZINB-WaVE, on the other hand, does not have these three key advantages.

5.F Details on Slingshot and uncertainty tube

5.F.1 Modifications to Slingshot

Our implementation of Slingshot differs from its original implementation in Street et al. (2018) in a few aspects. The first two modifications is related to how the lineages (i.e., what the different branches are and the ordering of the cell sub-types) are estimated, and the last two modifications is related to how the trajectories are estimated given the lineages (i.e., what is the numeric curve that interpolates the points in the lower-dimensional space).

- **Respecting natural order**: Our implementation respects the order among the major cell types (i.e., Pdgfra+ precusor, oligodendrocyte precursor cells, differentiation-committed oligodendrocyte precursors, newly formed oligodendrocytes, myelin-forming oligodendrocytes, and mature oligodendrocytes), but allows the lineage to link any of the cell sub-types to one another within the same cell type.
- **Construction of lineage**: Our implementation determines the lineage via a shortest path tree from the starting cluster as opposed to a minimum spanning tree (used by the original Slingshot), where in either case the distance is determined by the Gaussian distance, which is originally used by Slingshot.
- **Enable upsampling**: The original Slingshot did not weight clusters, so the lineage curves naturally gravitated towards the larger clusters. To compensate for this phenomenon, we upsampled the cells in each cluster via resampling with replacement unclear each cluster has the same number of cells. This effectively adds larger weights to these smaller clusters so each cluster is treated equally regardless of its original size.
- **Smoothing choice**: Our implementation uses a kernel smoother to smooth the data jointly with respect pseudotimes prior to fitting the principal curves. Previously, the original Slingshot fits a smoothing spline on each variable separately with respect to pseudotimes, but this resulted in undesirable behavior in certain cases.

5.F.2 Construction of uncertainty tube

We construct the uncertainty tubes via a bootstrap-based approach to determine if the different cell developmental trajectories are substantially different. This is done via the following procedure.

1. **Bootstrapping**: For a specific trial, sample with replacement the low-dimensional embedding $\widehat{X}_1, \ldots, \widehat{X}_n$ among each cell sub-type. This generates a new dataset with the same proportion of cell sub-types. Run Slingshot on this new dataset to obtain a new set of trajectories.

2. **Computing the quantile of ℓ_2 distance between lineage curves**: Let $\widehat{T}$ be one of possibly many estimated trajectories based on the original estimated embedding $\widehat{X}_1, \ldots, \widehat{X}_n$. For each trajectory estimate T_b from the newly generated dataset that matches $\widehat{T}$ based on the order of cell sub-types, compute the pointwise ℓ_2 distance between the two respective curves. Specifically, letting $\{T_{b,1}, \ldots, T_{b,N}\}$ and $\{\widehat{T}_1, \ldots, \widehat{T}_M\}$ denote the N and M discrete k-dimensional points in order that represent the trajectory T_b and $\widehat{T}$ respectively, this pointwise ℓ_2 distance is computed as the 95% quantile of the following set

$$\Big\{ \min_{j \in \{1,\ldots,M\}} \|T_{b,i} - \widehat{T}_j\|_2, \ \forall i \in \{1, \ldots, N\} \Big\}.$$

   After computing this distance for each of the bootstrapped trajectories, let $\sigma(\widehat{T})$ be the the 95% quantile ℓ_2 distance among all the bootstrapped trajectories, denoting the "margin of error" for a particular original estimated trajectory $\widehat{T}$. Let σ be the maximum value of such values among all the original estimated trajectories, representing the width of the uncertainty tubes.

3. **Computing the lineage tube**: For each lineage in the original dataset, construct a "tube" of radius σ around the lineage curve.

4. **Pruning**: If more than 90% of a particular lineage is "covered" in another lineage curve's tube, then we say that this lineage is "within the margin of error" and concatenate the two lineages together.

5.G Additional plots of results

In Figure 5.12 and Figure 5.13, we show the two-dimensional plots of the estimated embeddings to provide more clarity to three-dimensional plots shown in Figure 5.6 and Figure 5.9 in the main text. In Figure 5.14 and Figure 5.15, we show the same three-dimensional plots as in Figure 5.6 and Figure 5.9 in the main text, but from different viewing perspectives.

5.H Proofs

In Appendix 5.H.1, we prove Proposition 37. In Appendix 5.H.2, we prove Proposition 35 and its related lemmas. In Appendix 5.H.3, we prove Proposition 36 and its related lemmas. In Appendix 5.H.4, we prove the results shown in Appendix 5.D.

Throughout these proofs, for generic matrices A and B, let $\|A\|_{\text{op}}$ denote the spectral norm, i.e., the largest singular value. If A is a square matrix, let $\text{tr}(A)$ denote the trace of A, i.e., the sum of its diagonal elements. We write $A \otimes B$ to denote the Kronecker product of A and B, and if A and B are of the same dimensions, we write $A \preceq B$ if $B - A$ is positive

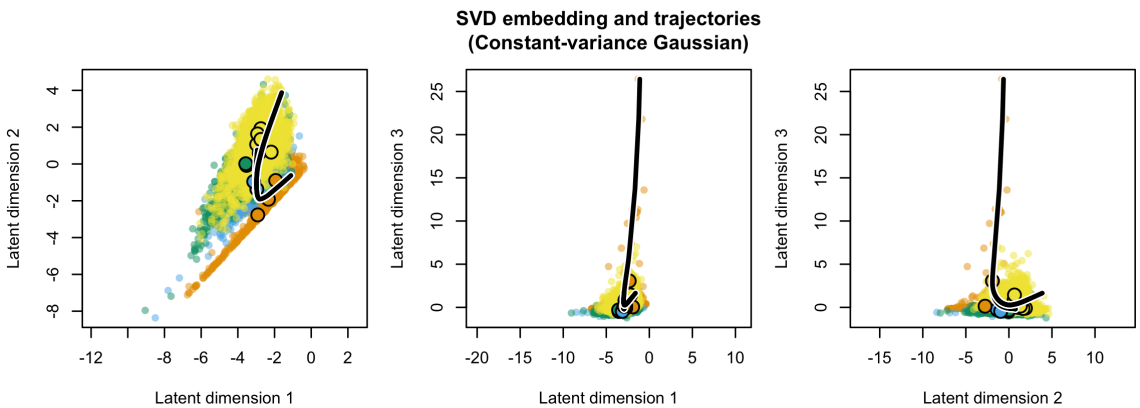


Figure 5.12: Two-dimensional plots of the SVD embedding, which corresponds to the three-dimensional plots shown in Figure 5.6. Each cell is color coded using the same color scheme from the aforementioned figure, the large dots represent the cluster centers for each of the thirteen cell subtypes, and the estimated trajectories are overlaid on top, which correspond to the same trajectories in the aforementioned figure. The three plots correspond to the three pairs of latent dimensions, one of the plots being the same as Figure 5.1 (after rotation).

semidefinite. For two random sequences A_n and B_n, let $A_n = \Theta_P(B_n)$ denote that B_n/A_n is bounded in probability for large enough n. Also, let I_k denote the identity matrix of size $k \times k$ and $\mathbf{1}_{n\times p}$ represent the $n \times p$ matrix of all ones. Throughout these proofs, we implicitly use different equivalent definitions of strong convexity and functions with Lipschitz gradients as stated in Zhou (2018).

5.H.1 Proof for Proposition 37

Proof of Proposition 37. Define the eigendecompositions of the second moment matrices for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$,

$$C_X^* = \mathbb{E}[X_i' X_i'^\top] = \Phi\Lambda\Phi^\top, \quad \text{and} \quad C_Y^* = \mathbb{E}[Y_j' Y_j'^\top] = \Psi\Gamma\Psi^\top, \tag{5.29}$$

where Φ and Ψ are both $k \times k$ unitary matrices. Our construction of the invertible matrix $R \in \mathbb{R}^{k\times k}$ will be done in two steps. In the first step, we first construct an invertible matrix $\widetilde{R} \in \mathbb{R}^{k\times k}$ such that

$$\widetilde{R} C_X^* \widetilde{R}^\top = \widetilde{R}^{-\top} C_Y^* \widetilde{R}^{-1}. \tag{5.30}$$

This would yield a transformation matrix to ensure $\mathbb{E}[X_i X_i^\top] = \mathbb{E}[Y_i Y_i^\top]$. In the second step, we adjust $\widetilde{R}$ into R in order to ensure that both $\mathbb{E}[X_i X_i^\top]$ and $\mathbb{E}[Y_i Y_i^\top]$ are diagonal.

Step 1: Based on the definition (5.29) and the desired goal shown in (5.30), an equivalent goal is to show that

$$\widetilde{R}^\top \widetilde{R} C_X^* \widetilde{R}^\top \widetilde{R} = \Psi\Gamma\Psi^\top.$$

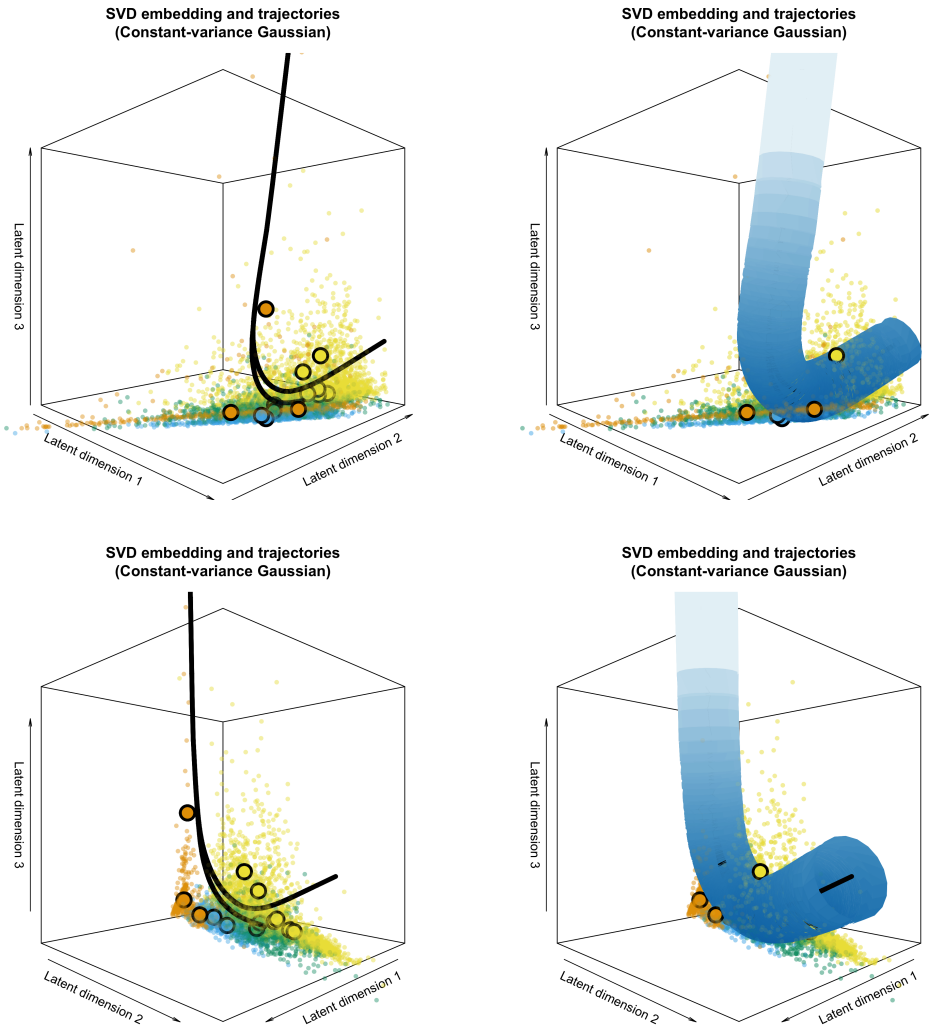


Figure 5.13: Three-dimensional plots of the estimated latent positions via SVD, without and with the uncertainy tube overlaid ontop. This plot is of the same estimated embedding as shown in Figure 5.6 but shown from a different perspective (each perspective represented by a different row).

Let $Q \in \mathbb{R}^{k \times k}$ denote any unitary matrix. Since the matrices on both sides of the above display are symmetric, we have

$$Q\Lambda^{1/2}\Phi^\top \widetilde{R}^\top \widetilde{R} = \Gamma^{1/2}\Psi^\top.$$

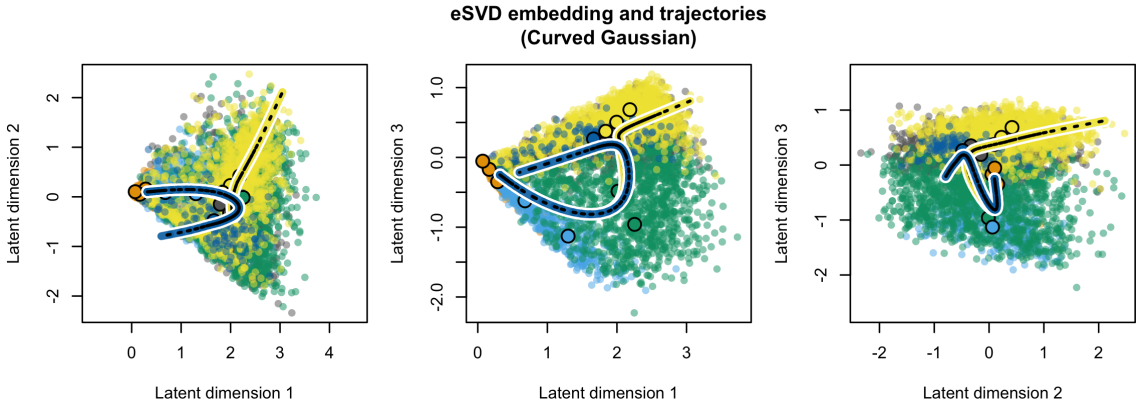


Figure 5.14: Two-dimensional plots of the eSVD embedding using the curved Gaussian distribution with $\tau = 2$, which corresponds to the three-dimensional plots shown in Figure 5.9. Each cell is color coded using the same color scheme from the aforementioned figure, the large dots represent the cluster centers for each of the thirteen cell sub-types, and the estimated trajectories are overlaid on top, which correspond to the same trajectories in the aforementioned figure. The three plots correspond to the three pairs of latent dimensions, one of the plots being the same as Figure 5.8B.

Rearranging, we have

$$\widetilde{R}^\top \widetilde{R} = \underbrace{\Phi \Lambda^{-1/2} Q^\top \Gamma^{1/2} \Psi^\top}_{B}.$$

Hence, we are done once we construct a unitary matrix Q that makes B symmetric. Observe that if the matrix

$$Q^\top \Gamma^{1/2} \Psi^\top \Phi \Lambda^{1/2}$$

were symmetric, then B would be symmetric. (This can be seen by multiplying the above matrix on the left by $E = \Phi \Lambda^{-1/2}$ and on the right by $E^\top$.) Observe that $\Gamma^{1/2} \Psi^\top \Lambda^{-1/2} \Phi$ is guaranteed to be full rank (by assumption of C_X^* and C_Y^* being full rank), so it admits a rank-k SVD of $UDV^\top$. Since the product of two unitary matrices is still unitary, we set $Q = UV^\top$. Hence, we finished our construction of $\widetilde{R}$.

Step 2: Suppose $\widetilde{R} C_X^* \widetilde{R}^\top$ (or equivalently, $\widetilde{R}^{-T} C_Y^* \widetilde{R}^{-1}$ based on (5.30)) has eigenvectors $W_1, \ldots, W_k$. Let W be the unitary matrix formed by concatenating these k eigenvectors column-wise. By diagonalization, we know $W^\top (\widetilde{R} C_X^* \widetilde{R}^\top) W$ is diagonal. This implies that our final construction is $R = W^\top \widetilde{R}$. □

5.H.2 Proofs for Proposition 35

Similar to Balakrishnan et al. (2017), we first analyze the behavior of an iteration of eSVD when using the population loss function $\mathcal{L}(\cdot, \cdot)$. Then, by leveraging Assumption 4, we can analyze the behavior of eSVD when applied on the sample loss function $\mathcal{L}_n(\cdot, \cdot)$.

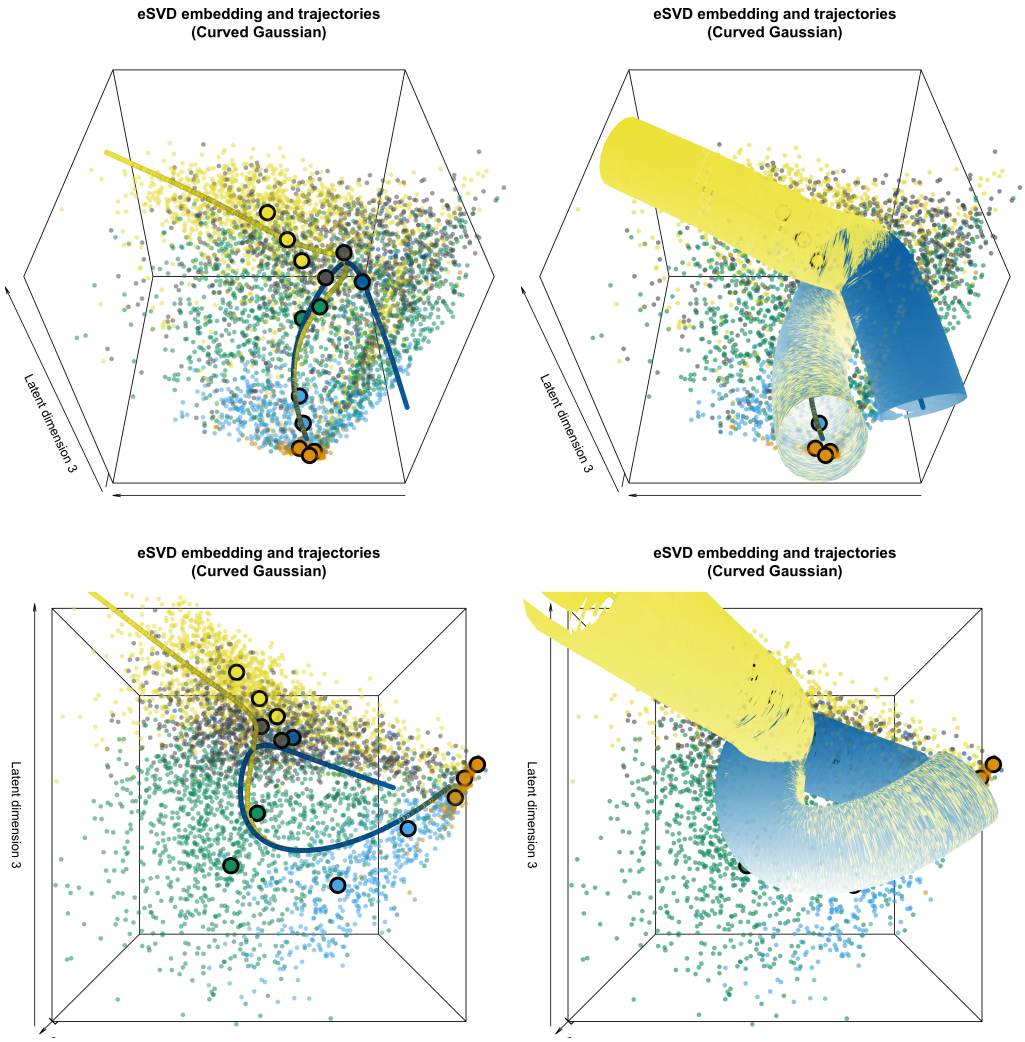


Figure 5.15: Three-dimensional plots of the estimated latent positions via eSVD, without and with the uncertainy tubes overlaid ontop. This plot is of the same estimated embedding as shown in Figure 5.9 but shown from a different perspective (each perspective represented by a different row).

Lemma 43. *Assume the population loss function $\mathcal{L}(\cdot,\cdot)$ satisfies Assumption 2. Conditioned on X and Y, then for any iteration $t = 0, \ldots, T-1$, there exists matrices $\{X^*, \overline{Y}^*\} \in$*

$\{\mathcal{X}^*, \overline{\mathcal{Y}}^*\}$ *and* $\{\overline{X}^*, Y^*\} \in \{\overline{\mathcal{X}}^*, \mathcal{Y}^*\}$ *such that*

$$\|X^* - M^X(\overline{Y}^{(t)})\|_F \leq \frac{S}{\mu}\|\overline{Y}^* - \overline{Y}^{(t)}\|_F,$$
$$\|Y^* - M^Y(\overline{X}^{(t)})\|_F \leq \frac{S}{\mu}\|\overline{X}^* - \overline{X}^{(t)}\|_F.$$

Proof. Observe that by the first-order optimality conditions, for any $\overline{Y}^* \in \overline{\mathcal{Y}}^*$, we have

$$\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^*),\ X' - M^X(\overline{Y}^*) \right\rangle \geq 0, \quad \text{for all } X' \text{ such that } (X_i')^\top \overline{Y}_j^* \in \mathcal{R} \quad \forall (i,j)$$

and similarly,

$$\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^{(t)}), \overline{Y}^{(t)}),\ X' - M^X(\overline{Y}^{(t)}) \right\rangle \geq 0, \quad \text{for all } X' \text{ such that } (X_i')^\top \overline{Y}_j^{(t)} \in \mathcal{R} \quad \forall (i,j)$$

By combining the two inequalities by setting $X' = M^X(\overline{Y})$ in the first display and $X' = M^X(\overline{Y}^*)$ in the second display with some algebra, we get

$$\begin{aligned}&\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^{(t)}), \overline{Y}^{(t)}) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}),\ M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle \\ &\quad \leq \left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^*) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}),\ M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle. \qquad (5.31)\end{aligned}$$

By manipulating the properties of strong convexity assumed in Assumption 2, we can lower bound the left-hand term in (5.31) by

$$\begin{aligned}&\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^{(t)}), \overline{Y}^{(t)}) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}),\ M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle \\ &\quad \geq \mu \|M^X(\overline{Y}^*) - M^X(\overline{Y}^{(t)})\|_F^2 = \mu \|X^* - X^{(t+1)}\|_F^2, \qquad (5.32)\end{aligned}$$

where we plugged in the definition of $X^* \in \mathcal{X}^*$ and $X^{(t+1)}$ in the last display.

Similarly, by the Lipschitz smoothness assumed in Assumption 2 with Cauchy-Schwarz inequality, we can upper bound the right-hand term in (5.31) by

$$\begin{aligned}&\left\langle \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^*) - \nabla_X \mathcal{L}(M^X(\overline{Y}^*), \overline{Y}^{(t)}),\ M^X(\overline{Y}^{(t)}) - M^X(\overline{Y}^*) \right\rangle \\ &\quad \leq S \|X^* - X^{(t+1)}\|_F \|\overline{Y}^* - \overline{Y}^{(t)}\|_F. \qquad (5.33)\end{aligned}$$

Combining (5.32) and (5.33) into the original equation (5.31) yields

$$\mu \|X^* - X^{(t+1)}\|_F^2 \leq S \|X^* - X^{(t+1)}\|_F \|\overline{Y}^* - \overline{Y}^{(t)}\|_F,$$

which completes the proof after simple rearrangements.

□

Similar to Zhao et al. (2015), we now prove Proposition 35 using the above lemma in conjunction with Lemma 48 which describes the effect of the LeftSVD operator.

Proof of Proposition 35 . For simplicity, let $\kappa = S/\mu$. By triangle inequality and Lemma 43, we have

$$\begin{aligned}\|M_n^Y(\overline{Y}^{(t)}) - X^*\|_F &\leq \|M_n^X(\overline{Y}^{(t)}) + M^X(\overline{Y}^{(t)})\|_F - \|M^X(\overline{Y}^{(t)}) - X^*\|_F \\ &\leq \varepsilon_{\text{unif}}(n,p) + \kappa\|\overline{Y}^* - \overline{Y}^{(t)}\|_F.\end{aligned}$$

Note that since $\varepsilon_{\text{unif}}(n,p) < \widetilde{d}_k/(4\max\{n,p\}^{1/2})$ by Assumption 4, if $\|\overline{Y}^* - \overline{Y}^{(t)}\|_F \leq \widetilde{d}_k/(4\max\{n,p\}^{1/2}\kappa)$, then by Lemma 48,

$$\|\overline{X}^* - \overline{X}^{(t+1)}\|_F \leq \frac{4(np)^{1/2}}{\widetilde{d}_k}\Big(\varepsilon_{\text{unif}}(n,p) + \kappa\|\overline{Y}^* - \overline{Y}^{(t)}\|_F\Big).$$

Similarly,

$$\|\overline{Y}^* - \overline{Y}^{(t)}\|_F \leq \frac{4(np)^{1/2}}{\widetilde{d}_k}\Big(\varepsilon_{\text{unif}}(n,p) + \kappa\|\overline{X}^* - \overline{X}^{(t)}\|_F\Big).$$

Therefore, by infinite geometric summation,

$$\begin{aligned}\|\overline{X}^* - \overline{X}^{(T)}\|_F &\leq \frac{4(np)^{1/2}}{\widetilde{d}_k}\Big(\varepsilon_{\text{unif}}(n,p) + \kappa\|\overline{Y}^* - \overline{Y}^{(T-1)}\|_F\Big) \\ &\leq \frac{4(np)^{1/2}}{\widetilde{d}_k}\Big(\varepsilon_{\text{unif}}(n,p) + \kappa\cdot\frac{4(np)^{1/2}}{\widetilde{d}_k}\Big(\varepsilon_{\text{unif}}(n,p) + \kappa\|\overline{X}^* - \overline{X}^{(T-1)}\|_F\Big)\Big) \\ &\leq \ldots \leq \frac{4(np)^{1/2}}{\widetilde{d}_k - 4(np)^{1/2}\kappa}\varepsilon_{\text{unif}}(n,p) + \Big(\frac{4(np)^{1/2}\kappa}{\widetilde{d}_k}\Big)^{(2T-1)}\|\overline{Y}^* - \overline{Y}^{(0)}\|_F,\end{aligned}$$

where the initialization condition Assumption 5 asserts $\|\overline{Y}^* - \overline{Y}^{(0)}\|_F \leq \widetilde{d}_k/(4(np)^{1/2}\kappa)$, which in turn shows $\max\{\|\overline{Y}^* - \overline{Y}^{(t)}\|_F, \|\overline{X}^* - \overline{X}^{(t)}\|_F\} \leq \widetilde{d}_k/(4(np)^{1/2}\kappa)$ for all $t = 1,\ldots,T$.

Let $Y^{(T)} = M_n^X(\overline{X}^{(T)})$. Therefore, for large enough T, we conclude for some universal constant C,

$$\begin{aligned}\|\Theta - \widehat{\Theta}\|_F &= \|\overline{X}^{*(T)}(Y^{*(T)})^\top - \overline{X}^{(T)}(Y^{(T)})^\top\|_F \\ &= \|\overline{X}^{*(T)}(Y^{*(T)})^\top - \overline{X}^{(T)}(Y^{*(T)})^\top + \overline{X}^{(T)}(Y^{*(T)})^\top - \overline{X}^{(T)}(Y^{(T)})^\top\|_F \\ &\leq \underbrace{\|Y^{*(T)}\|_2}_{\widetilde{d}_1/n^{1/2}}\|\overline{X}^{*(T)} - \overline{X}^{(T)}\|_F + \underbrace{\|\overline{X}^{(T)}\|_2}_{n^{1/2}}\|Y^{*(T)} - Y^{(T)}\|_F \\ &= C\cdot\Big(\frac{\max\{n^{1/2}, \widetilde{d}_1/n^{1/2}\}\cdot(np)^{1/2}}{\widetilde{d}_k - 4(np)^{1/2}\kappa}\cdot\varepsilon_{\text{unif}}(n,p)\Big).\end{aligned}$$

□

5.H.3 Proof for Proposition 36

We first introduce some notation. Let the SVD of X and Y be denoted as

$$X = \sqrt{n} U \Lambda^{1/2} \Phi^\top, \quad \text{and} \quad Y = \sqrt{p} V \Gamma^{1/2} \Psi^\top,$$

Using this definition, the empirical second moment matrices are

$$C_X = X^\top X / n = \Phi^\top \Lambda \Phi, \quad \text{and} \quad C_Y = Y^\top Y / p = \Psi^\top \Gamma \Psi.$$

Prior to the proof, we offer a high-level description of the proof strategy. Similar to the proof in Lei (2018), we introduce two levels of approximation. Let us focus on estimating $X_1^{(C)}, \ldots, X_n^{(C)}$, the latent variables that are oriented based on the population covariance matrix C_X^*. Formally,

$$X^{(C)} = X \Phi^*.$$

(Recall by Assumption 7, since $C^* X$ is diagonal, the columns of Φ^* are the standard basis vectors. To approximate this, we consider $X_1^{(S)}, \ldots, X_n^{(S)}$, the latent variables rotated by their own right singular vectors,

$$X^{(S)} = X \Phi = \sqrt{n} U \Lambda^{1/2}.$$

This approximation is driven by C_X being close to $\widehat{C}_X$. This is in turn estimated by $X_1^{(\Theta)}, \ldots, X_n^{(\Theta)}$ based on the SVD of Θ,

$$X^{(\Theta)} = \left(\frac{n}{p}\right)^{1/4} \widetilde{U} \widetilde{D}^{1/2}.$$

This approximation is driven by C_X^* being equal to C_Y^*. Finally, this is approximated by a quantity we can actually compute from data, our estimates $\widehat{X}_1, \ldots, \widehat{X}_n$ as in (5.11),

$$\widehat{X} = \left(\frac{n}{p}\right)^{1/4} \widehat{U} \widehat{D}^{1/2},$$

where $\widehat{U}$ and $\widehat{D}$ are obtained by an SVD of our estimate $\widehat{\Theta}$. This approximation is driven by $\widehat{\Theta}$ being close to Θ.

Proof of Proposition 36. **Step 1:** (Decomposition of error) By the triangle inequality, the rate is dictated by the term,

$$\frac{1}{n} \max \left\{ \|X^{(C)} - X^{(S)}\|_F^2, \|X^{(S)} - X^{(\Theta)}\|_F^2, \|X^{(\Theta)} - \widehat{X}\|_F^2 \right\}.$$

In the following three steps, we bound each term individually, of which the maximum of all three terms concludes the proof.

Step 2: (Approximation from $X^{(C)}$ to $X^{(S)}$) First, we deduce the relations of the eigenvalue λ_j and eigenvectors Φ_j. By applying Lemma 46, we obtain

$$\|C_X^* - C_X\|_{\mathrm{op}} = O_P((k/n)^{1/2}). \tag{5.34}$$

By using Weyl's inequality and the Davis-Kahan theorem, this immediately implies

$$\|\Phi_j^* - \Phi_j\|_2 = O_P(j^\beta (k/n)^{1/2}), \quad \forall j = 1, \ldots, k, \tag{5.35}$$

$$|\lambda_j^* - \lambda_j| = O_P((k/n)^{1/2}), \quad \forall j = 1, \ldots, k. \tag{5.36}$$

Since $k = o(\min\{n, p\})$ by assumption, this implies

$$\lambda_j = O_P(j^{-\alpha}), \quad \forall j = 1, \ldots, k. \tag{5.37}$$

Thus, using the trace function,

$$\begin{aligned}
\frac{1}{n}\|X^{(S)} - X^{(C)}\|_F^2 &= \frac{1}{n}\|X(\Phi^* - \Phi)\|_F^2 \\
&= \mathrm{tr}\left\{(\Phi^* - \Phi)^\top C_X (\Phi^* - \Phi)\right\} \\
&= \mathrm{tr}\left\{(\Phi^* - \Phi)^\top (C_X - C_X^*)(\Phi^* - \Phi)\right\} + \\
&\qquad \mathrm{tr}\left\{(\Phi^* - \Phi)^\top C_X^* (\Phi^* - \Phi)\right\}.
\end{aligned}$$

For the first term,

$$\begin{aligned}
&\left|\mathrm{tr}\left\{(\Phi^* - \Phi)^\top (C_X - C_X^*)(\Phi^* - \Phi)\right\}\right| \\
&\leq \sum_{j=1}^k \left|(\Phi_j^* - \Phi_j)^\top (C_X - C_X^*)(\Phi_j^* - \Phi_j)\right| \\
&\leq \sum_{j=1}^k \|C_X - C_X^*\|_{\mathrm{op}} \|\Phi_j^* - \Phi_j\|_2^2 = O_P(k^{2\beta + 5/2}/n^{3/2}).
\end{aligned}$$

For the second term, recalling that $\alpha \leq \beta$ and Parseval's identity,

$$\begin{aligned}
&\operatorname{tr}\Big\{(\Phi^* - \Phi)^\top C_X (\Phi^* - \Phi)\Big\} \\
&= \sum_{i=1}^k (\Phi_i^* - \Phi_i)^\top \Big[\sum_{j=1}^k \lambda_j^* \Phi_j^* \Phi_j^{*\top}\Big](\Phi_i^* - \Phi_i) \\
&= \sum_{j=1}^k \lambda_j^* \Big\{\sum_{i=1}^k \Big[(\Phi_i^* - \Phi_i)^\top \Phi_j^*\Big]^2\Big\} \\
&= \sum_{j=1}^k \lambda_j^* \Big\{\Big[(\Phi_j^* - \Phi_j)^\top \Phi_j^*\Big]^2 + \sum_{i \neq j, i \leq k} \Big[(\Phi_i^* - \Phi_i)^\top \Phi_j^*\Big]^2\Big\} \\
&= \sum_{j=1}^k \lambda_j^* \Big\{\Big[(\Phi_j^* - \Phi_j)^\top \Phi_j^*\Big]^2 + \sum_{p \neq j, i \leq k} \Big[(\Phi_j^* - \Phi_j)^\top \Phi_i\Big]^2\Big\} \\
&\leq 2 \sum_{j=1}^k \lambda_j^* \|\Phi_j^* - \Phi_j\|_2^2 = O_P(k^{2\beta - \alpha + 2}/n).
\end{aligned}$$

Step 3: (Approximation from $X^{(S)}$ to $X^{(\Theta)}$) First, observe that $\Theta = XY^\top = \sqrt{np} U \Lambda^{1/2} \Phi^\top \Psi \Gamma^{1/2} V^\top$, and we would like show it is close to the matrix $\sqrt{np} U \Lambda^{1/2} \Gamma^{1/2} V^\top$, which has an an SVD of $U(\sqrt{np}\Lambda^{1/2}\Gamma^{1/2})V^\top$. Specifically,

$$\begin{aligned}
&\|\sqrt{np} U \Lambda^{1/2} \Gamma^{1/2} V^\top - \sqrt{np} U \Lambda^{1/2} \Phi^\top \Psi \Gamma^{1/2} V^\top\|_{\mathrm{op}} \leq \\
&\qquad \sqrt{np} \|\Lambda^{1/2}\|_{\mathrm{op}} \|\Gamma^{1/2}\|_{\mathrm{op}} \|I_k - \Phi^\top \Psi\|_{\mathrm{op}},
\end{aligned} \tag{5.38}$$

where we can bound the last term by using the submultiplicative property of the spectral norm and $C_X^* = C_Y^*$,

$$\|I_k - \Phi^\top \Psi\|_{\mathrm{op}} = \|\Phi^\top(\Phi - \Psi)\|_{\mathrm{op}} \leq \|\Phi - \Phi^*\|_{\mathrm{op}} + \|\Psi^* - \Psi\|_{\mathrm{op}} = O_P\Big(\frac{k^{\beta+3/2}}{(\min\{n,p\})^{1/2}}\Big).$$

where the last inequality is due to Davis-Kahan. Hence, plugging the above display into (5.38),

$$\|\sqrt{np} U \Lambda^{1/2} \Gamma^{1/2} V^\top - \sqrt{np} U \Lambda^{1/2} \Phi^\top \Psi \Gamma^{1/2} V^\top\|_{\mathrm{op}} = O_P\Big(\frac{(np)^{1/2} \cdot k^{\beta+3/2}}{(\min\{n,p\})^{1/2}}\Big). \tag{5.39}$$

First, we derive the difference in eigenvalues based on (5.39). We apply Weyl's inequality to the above display to conclude

$$|(np\lambda_j\gamma_j)^{1/2} - \widetilde{d}_j| = O_P\Big(\frac{(np)^{1/2} \cdot k^{\beta+3/2}}{(\min\{n,p\})^{1/2}}\Big), \quad \forall j = 1, \ldots, k. \tag{5.40}$$

Note from (5.36) that we also have for all $j = 1, \ldots, k$,

$$|\lambda_j - \gamma_j| \leq |\lambda_j - \lambda_j^*| + |\gamma_j^* - \gamma_j| = O_P(k^{1/2}/\min\{n,p\}^{1/2}). \tag{5.41}$$

Hence, combining (5.40) and (5.41), noting that $\max\{|\lambda_j - \lambda_j^{1/2}\gamma_j^{1/2}|, |\gamma_j - \lambda_j^{1/2}\gamma_j^{1/2}|\} \leq |\lambda_j - \gamma_j|$, we derive

$$|n\lambda_j - (n/p)^{1/2}\widetilde{d}_j| = O_P\Big(\frac{n \cdot k^{\beta+3/2}}{(\min\{n,p\})^{1/2}}\Big), \quad \forall j = 1, \ldots, k. \tag{5.42}$$

Combining the above display with Assumption 7 and (5.37) and given $k = o(\min\{n,p\})$, this implies

$$\widetilde{d}_j = O_P((np)^{1/2} j^{-\alpha}), \quad \forall j = 1, \ldots, k. \tag{5.43}$$

Second, we derive the difference in eigenvectors based on (5.39). Observe that based on (5.36) and $k = o(\min\{n,p\})$, we can derive that $|(\lambda_j^*)^{1/2}(\gamma_j^*)^{1/2} - \lambda_j^{1/2}\gamma_j^{1/2}|$ is dominated by $\Omega_P(j^{-\beta})$. Hence, we can derive the spacing of the singular values of $\sqrt{np}U\Lambda^{1/2}\Gamma^{1/2}V^\top$,

$$(np)^{1/2}\big(|\lambda^{1/2}\gamma^{1/2} - \lambda_{j+1}^{1/2}\gamma_{j+1}^{1/2}|\big) = \Omega_P((np)^{1/2} j^{-\beta}), \quad \forall j = 1, \ldots, k-1.$$

Hence, using the Davis-Kahan theorem by combining (5.39) with the above display, we conclude

$$\|U_j - \widetilde{U}_j\|_2 = O_P\Big(\frac{j^\beta k^{\beta+3/2}}{(\min\{n,p\})^{1/2}}\Big), \quad \forall k = 1, \ldots, k. \tag{5.44}$$

Using (5.37), (5.42), and (5.44), along with Lemma 49,

$$\begin{aligned}
\frac{1}{n}\|X^{(S)} - X^{(\Theta)}\|_F^2 &= \frac{1}{n}\sum_{j=1}^k \|(n\lambda_j)^{1/2}U_j - (n/p)^{1/4}\widetilde{d}_j^{1/2}\widetilde{U}_j\|_2^2, \\
&\leq \frac{2}{n}\sum_{j=1}^k \|(n\lambda_j)^{1/2}(U_j - \widetilde{U}_j)\|_2^2 + \|(n\lambda_j)^{1/2} - (n/p)^{1/4}\widetilde{d}_j^{1/2})\widetilde{U}_j\|_2^2, \\
&\leq \frac{2}{n}\sum_{j=1}^k n\lambda_j\|U_j - \widetilde{U}_j\|_2^2 + O\Big(\Big(\frac{|n\lambda_j - (n/p)^{1/2}\widetilde{d}_j|}{(n\lambda_j)^{1/2}}\Big)^2\Big) \\
&= O_P(k^{4\beta-\alpha+4}/\min\{n,p\}).
\end{aligned}$$

Step 4: (Approximation from $X^{(\Theta)}$ to $\widehat{X}$) By assumption, we have

$$\|\Theta - \widehat{\Theta}\|_F \leq \epsilon. \tag{5.45}$$

This implies $\|\Theta - \widehat{\Theta}\|_{\text{op}} \leq (2k)^{1/2}\epsilon$. By Weyl's inequality, we conclude

$$|(n/p)^{1/2}\widetilde{d}_j - (n/p)^{1/2}\widehat{d}_j| = O_P((kn)^{1/2}\epsilon/p^{1/2}), \quad \forall j = 1, \ldots, k. \tag{5.46}$$

In addition, following a similar logic as above, using the Davis-Kahan theorem along with (5.42) to control the spacing of the eigenvalues, we can derive

$$\|\widetilde{U}_j - \widehat{U}_j\|_2 = O_P(j^{\beta}k^{1/2}\epsilon/(np)^{1/2}), \quad \forall j = 1, \ldots, k. \tag{5.47}$$

Hence, analogous to the derivation above, using (5.43), (5.46), and (5.47), along with Lemma 49,

$$\begin{aligned}
\frac{1}{n}\|X^{(\Theta)} - \widehat{X}\|_F^2 &= \frac{1}{n}\sum_{j=1}^{k}\|(n/p)^{1/4}\widetilde{d}_j^{1/2}\widetilde{U}_j - (n/p)^{1/4}\widehat{d}_j^{1/2}\widehat{U}_j\|_2^2 \\
&\leq \frac{2}{n}\sum_{j=1}^{k}\|(n/p)^{1/4}\widetilde{d}_j^{1/2}(\widetilde{U}_j - \widehat{U}_j)\|_2^2 + \|(n/p)^{1/4}(\widetilde{d}_j^{1/2} - \widehat{d}_j^{1/2})\widehat{U}_j\|_2^2 \\
&\leq \frac{2}{n}\sum_{j=1}^{k}(n/p)^{1/2}\widetilde{d}_j\|\widetilde{U}_j - \widehat{U}_j\|_2^2 + O\Big(\Big(\frac{|(n/p)^{1/2}\widetilde{d}_j - (n/p)^{1/2}\widehat{d}_j|}{(n/p)^{1/4}\widetilde{d}_j^{1/2}}\Big)^2\Big) \\
&= O_P(k^{2\beta-\alpha+2}\max(\epsilon^2, \epsilon)/(np)).
\end{aligned}$$

□

5.H.4 Proofs for Corollary 41 and Corollary 42

Useful facts. It is useful to have the following forms of the gradients and Heissan written down.

- The gradient $\nabla_X \mathcal{L}^{(X)}(X, \overline{Y}) \in \mathbb{R}^{n\times k}$ has the ith row equal to, for $i = 1, \ldots, n$,

$$\frac{1}{p}\sum_{j=1}^{p}\Big(-\frac{1}{X_i^\top \overline{Y}_j} - \tau^2\mathbb{E}[A_{ij}] + \tau^2\mathbb{E}[A_{ij}^2](X_i^\top \overline{Y}_j)\Big)\overline{Y}_j.$$

- Let $\nabla_Y[\nabla_X \mathcal{L}^{(X)}(X, \overline{Y})] \in \mathbb{R}^{(nk)\times(pk)}$ denote the gradient of above function with respect to $\overline{Y}$. If we focus on a particular block of k rows corresponding to a specific X_i and a particular block of k columns corresponding to a specific $\overline{Y}_j$, we have

$$\frac{1}{p}\Big[\Big(-\frac{1}{X_i^\top \overline{Y}_j} + \tau^2\mathbb{E}[A_{ij}] - \tau^2\mathbb{E}[A_{ij}^2](X_i^\top \overline{Y}_j)\Big)I_k + \Big(\frac{1}{(X_i^\top \overline{Y}_j)^2} + \tau^2\mathbb{E}[A_{ij}^2]\Big)X_i\overline{Y}_j^\top\Big].$$

- The Hessian matrix $\nabla_X^2 \mathcal{L}^{(X)}(X, \overline{Y}) \in \mathbb{R}^{(nk)\times(nk)}$ is all 0, except for n $k \times k$ blocks along the diagonal. The ith block is equal to, for $i = 1, \ldots, n$,

$$\frac{1}{p}\sum_{j=1}^{p}\Big(\frac{1}{(X_i^\top \overline{Y}_j)^2} + \tau^2 \mathbb{E}[A_{ij}^2]\Big)\overline{Y}_j(\overline{Y}_j)^\top.$$

We list properties about the curved Gaussian distribution assumed in (5.3) that will be needed. Recall that $A_{ij} \sim N(\mu_{ij}, \mu_{ij}^2/\tau^2)$, where $\mu_{ij} = -1/\theta_{ij}$.

- (First and second moment):

$$\mathbb{E}[A_{ij}] = 1/\theta_{ij}, \quad \text{and} \quad \mathbb{E}[A_{ij}^2] = \Big(\frac{1}{\tau^2} + 1\Big)\frac{1}{\theta_{ij}^2}. \tag{5.48}$$

- (Squared random variable):

$$A_{ij}^2 \overset{d}{=} \frac{1}{(\tau\theta_{ij})^2} Z_{ij}, \quad \text{where} \quad Z_{ij} \sim \chi^2\Big(k = 1, \lambda = \tau^2\Big).$$

  Hence, the variance of A_{ij}^2 is upper-bounded by

$$\kappa = \frac{2(1 + 2\tau^2)\cdot r^4}{\tau^4}.$$

Proof of Lemma 38. We start with the lower bound. Observe that minimum eigenvalue of one of the n $k \times k$ blocks will be the minimum eigenvalue of the overall Hessian matrix $\nabla_X^2 \mathcal{L}^{(X)}(X, \overline{Y})$. Hence, inspecting one particular block for a specific $i = 1, \ldots, n$, observe that

$$\frac{1}{p}\sum_{j=1}^{p}\Big(\frac{1}{(X_i^\top \overline{Y}_j)^2} + \tau^2 \mathbb{E}[A_{ij}^2]\Big)\overline{Y}_j(\overline{Y}_j)^\top \succeq \frac{2+\tau^2}{r^2}\cdot\frac{1}{p}\sum_{j=1}^{p}\overline{Y}_j(\overline{Y}_j)^\top,$$

where $\sum_j \overline{Y}_j(\overline{Y}_j)^\top = pI_k$ since the columns of $\overline{Y}$ are orthogonal. For the upper bound, we apply the same logic. □

Proof of Lemma 39. Recall by the multivariate Taylor expansion (Feng et al., 2013), we have

$$\nabla_X \mathcal{L}^{(X)}(X^*, \overline{Y}^*) - \nabla_X \mathcal{L}^{(X)}(X^*, \overline{Y}) = \int_0^1 \nabla_Y[\nabla_X \mathcal{L}(X^*, \overline{Y}^* + tu)]du \cdot t,$$

where $t = \overline{Y}^* - \overline{Y}$. This leads to the following inequality,

$$\|\nabla_X \mathcal{L}^{(X)}(X^*, \overline{Y}) - \nabla_X \mathcal{L}^{(X)}(X^*, \overline{Y}^*)\|_F \le \sup_{u\in[0,1]}\Big\|\underbrace{\nabla_Y[\nabla_X \mathcal{L}(X^*, \overline{Y}^* + tu)]}_{B(u)}\Big\|_{\text{op}}\|\overline{Y}^* - \overline{Y}\|_F.$$

Observe we can split the matrix $B(u)$ into two matrices, one sparse matrix and one dense matrix. That is, $B(u) = B_1(u) + B_2(u)$, all matrices of dimension $(nk) \times (pk)$. The (i,j)th $k \times k$ block of $B_1(u)$ has entries equal to

$$\frac{1}{p}\Big(- \frac{1}{(X_i^*)^\top(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))} - \tau^2\mathbb{E}[A_{ij}] + \tau^2\mathbb{E}[A_{ij}^2]\big((X_i^*)^\top(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))\big)\Big) I_k,$$

while the (i,j)th $k \times k$ block of $B_2(u)$ has entries equal to

$$\frac{1}{p}\Big(\frac{1}{((X_i^*)^\top(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))^2} + \tau^2\mathbb{E}[A_{ij}^2]\Big) X_i^*(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))^\top. \tag{5.49}$$

We first analyze $B_1(u)$. Plugging in (5.48), we obtain

$$\frac{1}{p}\Big(\underbrace{- \frac{1}{(X_i^*)^\top(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))} - \frac{\tau^2}{(X_i^*)^\top\overline{Y}_j^*} + (\tau^2+1)\cdot\frac{(X_i^*)^\top(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))}{\big((X_i^*)^\top\overline{Y}_j^*\big)^2}}_{C_1}\Big) I_k,$$

To analyze C_1 in the perspective of Assumption 9, we analyze

$$\max_t \Big| - \frac{r}{(1+t)} - \tau^2\cdot r + (\tau^2+1)(1+t)\cdot r\Big| \quad : \quad t \in \Big[- \frac{1}{\log(\min\{n,p\})}, \frac{1}{\log(\min\{n,p\})}\Big].$$

By analyzing the Lagrangian of the above display, we see that for $\min\{n,p\}$ large enough,

$$C_1 \le c\cdot\frac{\tau^2 r}{\log(\min\{n,p\})}$$

for some universal constant c. Hence, we derive that

$$B_1(u) \preceq c\tau^2 r\cdot\frac{1}{p\cdot\log(\min\{n,p\})}\cdot I_k \otimes \mathbf{1}_{n\times p}.$$

Therefore,

$$\|B_1(u)\|_{\mathrm{op}} \le c\tau^2 r\cdot\frac{(np)^{1/2}}{p\cdot\log(\min\{n,p\})}. \tag{5.50}$$

We next analyze $B_2(u)$. Plugging in (5.48), we obtain

$$\frac{1}{p}\Big(\underbrace{\frac{1}{((X_i^*)^\top(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))^2} + \frac{\tau^2+1}{\big((X_i^*)^\top\overline{Y}_j^*\big)^2}}_{C_2}\Big) X_i^*(\overline{Y}_j^* + u(\overline{Y}_j - \overline{Y}_j^*))^\top.$$

By analyzing C_2 similar to how we analyzed C_1, we derive that for $\min\{n, p\}$ large enough,

$$B_2(u) \preceq (\tau^2 + 2)r^2 \cdot \frac{1}{p} \cdot X^* \otimes \big(\overline{Y}^* + u(\overline{Y} - \overline{Y}^*)\big)^\top.$$

Therefore,

$$\|B_2(u)\|_{\mathrm{op}} \leq (\tau^2 + 2)r^2 \cdot \frac{2\widetilde{d}_1}{p}. \tag{5.51}$$

We conclude by combining (5.50) and (5.51) by a triangle inequality.

□

Proof of Lemma 40. Prior to proving Lemma 40, we need the following set of concentration bounds.

Lemma 44. *Conditioned on Θ, let A_{ij} follow the distribution* (5.5)*. Let $\kappa = (2(1 + 2\tau^2))r^4/\tau^4$. Under Assumption 8, for a fixed $i = 1, \ldots, n$, for some universal constant c, for fixed matrices X and $\overline{Y}$, each of the following four inequalities hold separately.*

$$\mathbb{P}\Big(\Big|\frac{1}{p}\sum_{j=1}^{p} -\tau^2 X_i^\top \overline{Y}_j (A_{ij} - \mathbb{E}[A_{ij}])\Big| \geq t\Big) \leq e \exp\Big[\frac{-ct^2}{\kappa^2\tau^4 r^2} \cdot p\Big], \tag{5.52}$$

$$\mathbb{P}\Big(\Big\|\frac{1}{p}\sum_{j=1}^{p} -\tau^2 \overline{Y}_j (A_{ij} - \mathbb{E}[A_{ij}])\Big\|_2 \geq t\Big) \leq ke \exp\Big[\frac{-ct^2}{\kappa^2\tau^4} \cdot \frac{p}{k^2}\Big], \tag{5.53}$$

$$\mathbb{P}\Big(\Big|\frac{1}{p}\sum_{j=1}^{p} \frac{\tau^2 (X_i^\top \overline{Y}_j)^2}{2} (A_{ij}^2 - \mathbb{E}[A_{ij}^2])\Big| \geq t\Big) \leq 2 \exp\Big[-c \min\Big(\frac{4t^2}{\kappa^2\tau^4 r^4}, \frac{2t}{\kappa\tau^2 r^2}\Big) \cdot p\Big], \tag{5.54}$$

$$\mathbb{P}\Big(\Big\|\frac{1}{p}\sum_{j=1}^{p} \big(\tau^2 (X_i^\top \overline{Y}_j)(A_{ij}^2 - \mathbb{E}[A_{ij}^2])\big)\overline{Y}_j\Big\|_2 \geq t\Big) \leq 2k \exp\Big[-c \min\Big(\frac{t^2}{\kappa^2\tau^4 r^2}, \frac{t}{\kappa\tau^2 r}\Big) \cdot \frac{p}{k^2}\Big]. \tag{5.55}$$

Proof. (5.52) and (5.54) are direct applications of Hoeffding's and Bernstein's inequality (Vershynin (2010), Proposition 5.10 and 5.16) since $X_i^\top \overline{Y}_j \leq r$. To show (5.53), observe that for a particular coordinate $\ell = 1, \ldots, k$, we can apply Hoeffding's inequality to show

$$\mathbb{P}\Big(\Big|\frac{1}{p}\sum_{j=1}^{p} -\tau^2 \overline{Y}_{j\ell} (A_{ij} - \mathbb{E}[A_{ij}])\Big| \geq t\Big) \leq e \exp\Big(\frac{-ct^2}{\kappa^2\tau^4} \cdot p\Big),$$

since the ℓ_2-norm of any column of $\overline{Y}$ is $\sqrt{p}$. Applying a maximal inequality and then summing over all k dimensions, we have

$$\mathbb{P}\Big(\Big\|\frac{1}{p}\sum_{j=1}^{p} -\tau^2\overline{Y}_j(A_{ij}-\mathbb{E}[A_{ij}])\Big\|_1 \geq t\Big) \leq ke\exp\Big(\frac{-ct^2}{\kappa^2\tau^4}\cdot\frac{p}{k^2}\Big),$$

which upper-bounds the RHS of (5.53). The same technique using Bernstein's inequality can be used to show (5.55).

□

With these concentration statements, we are ready to proceed with the analysis of $\varepsilon_{\text{fixed}}(n,p)$ row-wise.

Lemma 45. *Assume the conditions in Lemma 38. Let*

$$\widehat{\Lambda} = \underset{\Lambda\in\mathbb{R}^k}{\operatorname{argmin}}\,\frac{1}{p}\sum_{j=1}^{p}\Big[-\log(\Lambda^\top\overline{Y}_j) - \begin{bmatrix}\tau^2 A_{ij}\\ -\tau^2 A_{ij}^2/2\end{bmatrix}^\top\begin{bmatrix}\Lambda^\top\overline{Y}_j\\ (\Lambda^\top\overline{Y}_j)^2\end{bmatrix}\Big], \tag{5.56}$$

and

$$\Lambda^* = \underset{\Lambda\in\mathbb{R}^k}{\operatorname{argmin}}\,\frac{1}{p}\sum_{j=1}^{p}\Big[-\log(\Lambda^\top\overline{Y}_j) - \begin{bmatrix}\tau^2\mathbb{E}[A_{ij}]\\ -\tau^2\mathbb{E}[A_{ij}^2/2]\end{bmatrix}^\top\begin{bmatrix}\Lambda^\top\overline{Y}_j\\ (\Lambda^\top\overline{Y}_j)^2\end{bmatrix}\Big]. \tag{5.57}$$

Then, conditioned on X and Y, with probability at least $1-6/(np)$,

$$\|\Lambda^*-\widehat{\Lambda}\|_2 \leq c\Big(\frac{\log(np)}{p}\Big)^{1/4},$$

where c is a constant that depends only on k, μ, L, τ and r.

Proof. **Step 1**: (Setup) Our proof is inspired by Lemma 4 of Candès and Sur (2018). The proof is fairly straightforward and relies mainly on the convexity properties of $\mathcal{L}$ and the concentration statements established in Lemma 44.

Throughout this proof, we let $c_1, c_2, \ldots$ denote constants depend on quantities that we will explicitly state, but their explicit form can change from line to line. Let $\mathcal{L}_n(\cdot)$ and $\mathcal{L}(\cdot)$ denote functions being minimized in (5.56) and (5.57) respectively throughout this proof only. Since $\mathcal{L}$ is μ-strongly convex, we have

$$\mathcal{L}(\Lambda) \geq \mathcal{L}(\Lambda^*) + \frac{1}{2\mu}\|\Lambda^*-\Lambda\|_2^2.$$

Define $\|\Lambda^* - \Lambda\|_2 = c_1 v$ and $h = v^2/(6c_1^2\mu)$ for some constant c_1 that depends on quantity defined later in the proof. Using these definitions, the above display equals

$$\mathcal{L}(\Lambda) \geq \mathcal{L}(\Lambda^*) + 3h. \tag{5.58}$$

Next, consider the sphere centered at Λ^* with radius v, $\mathcal{B}_v(\Lambda^*) = \{\Lambda : \|\Lambda^* - \Lambda\|_2 = v\}$. Consider the event E defined as

$$\inf_{\Lambda \in \mathcal{B}(\Lambda^*)} \mathcal{L}_n(\Lambda) > \mathcal{L}(\Lambda^*) + h \quad \text{and} \quad \mathcal{L}_n(\Lambda^*) < \mathcal{L}(\Lambda^*) + h. \tag{5.59}$$

Note that when event E occurs, by convexity of $\mathcal{L}_n$, $\widehat{\Lambda}$ (the minimizer of $\mathcal{L}_n$) must lie within $\mathcal{B}(\Lambda^*)$, so hence, $\|\widehat{\Lambda} - \Lambda^*\|_2 \leq v$.

Step 2: (Decomposition of E) We decompose E in the above display into three separate events that we will control individually. Let the event E_1 be defined as

$$\max_{i=1,\ldots,M} |\mathcal{L}(\Lambda_i) - \mathcal{L}_n(\Lambda_i)| \leq h. \tag{5.60}$$

and the event E_2 be defined as

$$\inf_{\Lambda \in \mathcal{B}_v(\Lambda^*)} \mathcal{L}_n(\Lambda) \geq \mathcal{L}_n(\Lambda_i) - h. \tag{5.61}$$

Observe that from event E_1, we have the following line of implications

$$\begin{aligned}
\max_{i=1,\ldots,M} |\mathcal{L}(\Lambda_i) - \mathcal{L}_n(\Lambda_i)| < h \quad &\Longrightarrow \quad \forall i = 1, \ldots, M, \quad \mathcal{L}_n(\Lambda_i) \geq \mathcal{L}(\Lambda_i) - h \\
&\Longrightarrow \quad \forall i = 1, \ldots, M, \quad \mathcal{L}_n(\Lambda_i) - h > \mathcal{L}(\Lambda^*) + h, \\
&\Longrightarrow \quad \inf_{\Lambda \in \mathcal{B}_v(\Lambda^*)} \mathcal{L}_n(\Lambda) > \mathcal{L}(\Lambda^*) + h,
\end{aligned}$$

where the second implication follows from the definition of h in (5.58), and the last implication follows from E_2. These together would imply the first part of E in (5.59).

Observe that the following event E_3, defined as,

$$|\mathcal{L}(\Lambda^*) - \mathcal{L}_n(\Lambda^*)| \leq h \tag{5.62}$$

would imply the second part of E in (5.59). Hence, since the event $E_1 \cap E_2 \cap E_3$ implies event E, we have by union bound

$$\mathbb{P}(E^c) \leq \mathbb{P}(E_1^c) + \mathbb{P}(E_2^c) + \mathbb{P}(E_3^c).$$

Therefore, the remainder of the proof shows that for a rate of v, the summation of the probability of the complementary events is small.

Step 3: (Analysis of E_1) Using at most $M = (3/\gamma)^k$ matrices, denoted as $\Lambda_1, \ldots, \Lambda_M$, we can construct an γ-net of the sphere $\mathcal{B}(\Lambda^*)$ via Lemma 9.5 of Ledoux and Talagrand (2013), i.e.,

$$\text{for any } \Lambda \in \mathcal{B}(\Lambda^*), \quad \min_{i=1,\ldots,M} \|\Lambda - \Lambda_i\|_2 \leq \gamma.$$

We analyze E_1 in (5.60), the event that $\mathcal{L}_n(\Lambda_i)$ and $\mathcal{L}(\Lambda_i)$ are close along the γ-net. Using a union bound along with concentration bounds shown in Lemma 44, for p large enough, we have with probability at least $1 - 2/(np)$ that event E_1 occurs,

$$\max_{i=1,\ldots,M} |\mathcal{L}(\Lambda_i) - \mathcal{L}_n(\Lambda_i)| \leq c_2 \Big(\frac{\log(np) + k \log(3/\gamma)}{p} \Big)^{1/2},$$

where c_2 depends on κ, τ and r. Treating k as a constant, to set the left-hand side of the above display to be equal to h, we need

$$v = c_3 \big(\log(np)/p \big)^{1/4}, \tag{5.63}$$

where c_3 depends on k, μ, κ, τ, and r, and is poly-logarithmic in γ.

Step 4: (Analysis of E_2) Next, we analyze E_2 in (5.61), the event that for any $\Lambda \in \mathcal{B}_v(\Lambda^*)$, we have $\mathcal{L}_n(\Lambda)$ is close to $\mathcal{L}_n(\Lambda_i)$ for some i along the γ-net. By convexity, letting Λ_i be the closest vector to Λ in ℓ_2 distance,

$$\mathcal{L}_n(\Lambda) \geq \mathcal{L}_n(\Lambda_i) + \langle \nabla \mathcal{L}_n(\Lambda_i), \Lambda - \Lambda_i \rangle \geq \mathcal{L}_n(\Lambda_i) - \|\nabla \mathcal{L}_n(\Lambda_i)\|_2 \|\Lambda - \Lambda_i\|_2. \tag{5.64}$$

We know $\|\Lambda - \Lambda_i\|_2 \leq \gamma$ by construction. To bound $\|\nabla \mathcal{L}_n(\Lambda_i)\|_2$, using a union bound and Lemma 44 once again, we have with probability at least $1 - 2/(np)$ for p large enough, event E_2' occurs,

$$\max_i \|\nabla \mathcal{L}(\Lambda_i) - \nabla \mathcal{L}_n(\Lambda_i)\|_2 \leq c_1 k \Big(\frac{\log(np) + k \log(3/\gamma)}{p} \Big)^{1/2}. \tag{5.65}$$

Furthermore, we know that since $\nabla \mathcal{L}(\Lambda^*) = 0$ and the gradient of $\mathcal{L}$ is L-Lipschitz smooth, using the definition of v in (5.63), we have

$$\|\nabla \mathcal{L}(\Lambda_i) - \nabla \mathcal{L}(\Lambda^*)\|_2 \leq L \|\Lambda_i - \Lambda^*\|_2 = Lv = Lc_2 \Big(\frac{\log(np)}{p} \Big)^{1/4}. \tag{5.66}$$

Therefore, combining (5.65) and (5.66), that on event E_2',

$$\max_{i=1,\ldots,M} \|\nabla \mathcal{L}_n(\Lambda_i)\|_F \|\Lambda_i - \widehat{\Lambda}\|_F \leq \Big(c_2 \Big(\frac{\log(np)}{p} \Big)^{1/2} + Lc_2 \Big(\frac{\log(np)}{p} \Big)^{1/4} \Big) \cdot \gamma.$$

If we set $\gamma = (\log(np)/p)^{1/4}$, then plugging the above inequality back into (5.64), we have on event E_2,

$$\mathcal{L}_n(\Lambda) \geq \mathcal{L}_n(\Lambda_i) - c_3 \Big(\frac{\log(np)}{p} \Big)^{1/2} = \mathcal{L}_n(\Lambda_i) - h, \tag{5.67}$$

where c_3 depends on L, k, μ, κ, τ, and r, and is poly-logarithmic in γ. This shows E_2.

Step 5: (Analysis of E_3 and conclusion) From similar calculations above based on Lemma 44, we can show that for p large enough, we have with probability at least $1-2/(np)$ that event E_3 occurs,

$$|\mathcal{L}(\Lambda^*) - \mathcal{L}_n(\Lambda^*)| \leq h.$$

Hence, we conclude the proof, where c_1 initially stated is a constant that depends on L, k, μ, κ, τ, and r, and is poly-logarithmic in γ. By Lemma 38, we know L and μ are constants that depend on only τ and r.

□

We are now ready to prove Lemma 40.

Proof of Lemma 40. Let $\widehat{X} = M_n(\overline{Y})$ and $X^* = M(\overline{Y})$. By Lemma 45, we have with probability $6/(np)$, for any $i = 1, \ldots, n$,

$$\|X_i^* - \widehat{X}_i\|_2^2 \geq c\Big(\frac{\log(np)}{p}\Big)^{1/2}.$$

Hence, applying a union bound over all $i = 1, \ldots, n$, we have with probability $6/p$,

$$\max_i \|X_i^* - \widehat{X}_i\|_2^2 \geq c\Big(\frac{\log(np)}{p}\Big)^{1/2}.$$

This event implies that,

$$\|X^* - \widehat{X}\|_F^2 \geq cn\Big(\frac{\log(np)}{p}\Big)^{1/2}.$$

□

Proof of Corollary 41. The proof mainly follows from the proof of Proposition 35 above, where we replace the term $\varepsilon_{\text{unif}}(n,p)$ with $\varepsilon_{\text{fixed}}(n,p)$, which requires us to take a union bound over all T iterations.

Due to our invocation of Assumptions 6 and 7, we have shown already in (5.40) that $\widetilde{d}_j$ scales with $(np\lambda_j\gamma_j)^{1/2}$ asymptotically, completing the proof. □

5.I Auxiliary results and proofs

The first result follows from Proposition 2.1 of Vershynin (2012).

Lemma 46 (Probability bound of operator norm of a covariance matrix)**.** *Let $X_1, \ldots, X_n$ be i.i.d. k-dimensional random variables that satisfies Assumption 6. Then, with probability at least $1 - 1/k$,*

$$\left\| \mathbb{E}\big[X_i X_i^T\big] - \frac{1}{n}\sum_{i=1}^{n} X_i X_i^T \right\|_2 \leq CD \cdot \sqrt{\frac{k}{n}},$$

for some universal constant C.

The following perturbation result combines the result in Yu et al. (2014) with Hermatian dilation (see, e.g. Tropp (2012)).

Lemma 47 (Davis-Kahan theorem for singular vectors)**.** *For any two $n \times p$ rank k symmetric matrices M and M^*, let $\sigma_1 > \ldots > \sigma_k$ and $\sigma_1^* > \ldots > \sigma_k^*$ denote the singular values of each matrix respectively, and $U_1, \ldots, U_k$ and $U_1^*, \ldots, U_k^*$ denote their corresponding left singular vectors. Fix $1 \leq r \leq s \leq k$ and let $d = s - r + 1$, and let $U = [U_r, \ldots, U_s]$ and $U^* = [U_r^*, \ldots, U_s^*]$ denote the concatenated matrix of left singular vectors. Then,*

$$\|U^* - U\|_F \leq \frac{2^{3/2} d^{1/2} \|M^* - M\|_2}{\min(\sigma_{r-1}^* - \sigma_r^*, \sigma_s^* - \sigma_{s+1}^*)}.$$

The following perturbation bound relates the Forbenius distance between two matrices to the Forbenius distance between their matrix of singular vectors counterparts.

Lemma 48. *Assume two matrices $Z, Z^* \in \mathbb{R}^{n \times p}$ with top rank-k SVDs of $UDV^\top$ and $U^* D^* (V^*)^\top$ respectively satisfy*

$$\|Z - Z^*\|_F \leq \frac{\sigma_{\min}(Z^*)}{2}.$$

Denote

$$\widehat{Q} = \operatorname*{argmin}_{R \in \mathcal{O}^{k \times k}} \|U^* - UR\|_F.$$

Then,

$$n^{1/2} \|U^* - UQ\|_F \leq \frac{4n^{1/2}}{\sigma_{\min}(Z^*)} \|Z^* - Z\|_F.$$

The proof for Lemma 48 can proven using the same techniques as in Lemma 45 of Ma et al. (2018). The next result on random sequences can be easily verified by direct calculation.

In the below lemma, for sequences a_n, b_n, let $a_n = o(b_n)$ denote that $a_n/b_n \to 0$ and for random sequences A_n, B_n, let $A_n = \Omega_p(B_n)$ denote B_n/A_n is bounded in probability for large enough n.

Lemma 49. *For two positive deterministic sequences a_n and b_n, consider a random sequence $X_n = O_P(a_n)$ and a random positive sequence $Y_n = \Omega_P(b_n)$. If $a_n = o(b_n)$, then*

$$(Y_n + X_n)^{1/2} - Y_n^{1/2} = O_P(a_n b_n^{-1/2}).$$

Bibliography

Abbe, E. (2017). Community detection and stochastic block models: Recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

Alamgir, M. and Von Luxburg, U. (2012). Shortest path distance in random k-nearest neighbor graphs. *arXiv preprint arXiv:1206.6381*.

Alzeyadi, M. (2013). *Genetic Characteristics and Prognostic Markers of Lung Cancer*. PhD thesis.

Arora, S., Ge, R., Kannan, R., and Moitra, A. (2016). Computing a nonnegative matrix factorization—provably. *SIAM Journal on Computing*, 45(4):1582–1611.

Arora, S., Ge, R., and Moitra, A. (2012). Learning topic models–going beyond SVD. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE.

Aston, J. A. and Kirch, C. (2012). Evaluating stationarity via change-point alternatives with applications to fmri data. *The Annals of Applied Statistics*, pages 1906–1948.

Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., and Qin, Y. (2017). Statistical inference on random dot product graphs: A survey. *The Journal of Machine Learning Research*, 18(1):8393–8484.

Aue, A. and Horvath, L. (2013). Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16.

Autism and Investigators, D. D. M. N. S. Y. . P. (2014). Prevalence of autism spectrum disorder among children aged 8 years - Autism and developmental disabilities monitoring network, 11 sites, United States, 2010. *Morbidity and Mortality Weekly Report: Surveillance Summaries*, 63(2):1–21.

Balakrishnan, S., Wainwright, M. J., Yu, B., et al. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120.

Bean, J., Brennan, C., Shih, J.-Y., Riely, G., Viale, A., Wang, L., Chitale, D., Motoi, N., Szoke, J., Broderick, S., et al. (2007). MET amplification occurs with or without T790M mutations in EGFR mutant lung tumors with acquired resistance to gefitinib or erlotinib. *Proceedings of the National Academy of Sciences*, 104(52):20932–20937.

Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W., Ng, L. G., Ginhoux, F., and Newell, E. W. (2019). Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38.

Boysen, L., Kempe, A., Liebscher, V., Munk, A., and Wittich, O. (2009). Consistencies and rates of convergence of jump-penalized least squares estimators. *The Annals of Statistics*, 37(1):157–183.

Brodsky, B. and Darkhovski, B. (1993). *Nonparametric Methods in Change-Point Problems*. Springer.

Buxbaum, J. D., Daly, M. J., Devlin, B., Lehner, T., Roeder, K., State, M. W., and The Autism Sequencing Consortium (2012). The Autism Sequencing Consortium: Large-scale, high-throughput sequencing in autism spectrum disorders. *Neuron*, 76(6):1052–1056.

Cai, T., Liu, W., and Xia, Y. (2013). Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings. *Journal of the American Statistical Association*, 108(501):265–277.

Candès, E. J. and Sur, P. (2018). The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *arXiv preprint arXiv:1804.09753*.

Chan, N. H., Yau, C. Y., and Zhang, R.-M. (2014). Group lasso for structural break time series. *Journal of the American Statistical Association*, 109(506):590–599.

Chang, J., Zhou, W., Zhou, W.-X., and Wang, L. (2017). Comparing large covariance matrices under weak conditions on the dependence structure and its application to gene clustering. *Biometrics*, 73(1):31–41.

Chen, H., Jiang, Y., Maxwell, K. N., Nathanson, K. L., and Zhang, N. (2017). Allele-specific copy number estimation by whole exome sequencing. *The annals of applied statistics*, 11(2):1169.

Chen, J. and Gupta, A. (2000). *Parametric Statistical Change Point Analysis*. Birkhauser.

Chen, J. and Saad, Y. (2010). Dense subgraph extraction with application to community detection. *IEEE Transactions on knowledge and data engineering*, 24(7):1216–1230.

Chen, M. and Zhou, X. (2018). VIPER: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. *Genome biology*, 19(1):196.

Chernozhukov, V., Chetverikov, D., Kato, K., et al. (2013). Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors. *The Annals of Statistics*, 41(6):2786–2819.

Chi, Y., Lu, Y. M., and Chen, Y. (2019). Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269.

Cho, H. and Fryzlewicz, P. (2011). Multiscale interpretation of taut string estimation and its connection to unbalanced haar wavelets. *Statistics and computing*, 21(4):671–681.

Collins, M., Dasgupta, S., and Schapire, R. E. (2002). A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems*, pages 617–624.

Cotney, J., Muhle, R. A., Sanders, S. J., Liu, L., Willsey, A. J., Niu, W., Liu, W., Klei, L., Lei, J., and Yin, J. (2015). The autism-associated chromatin modifier CHD8 regulates other autism risk genes during human neurodevelopment. *Nature communications*, 6.

Dalalyan, A. S., Hebiri, M., and Lederer, J. (2017). On the prediction performance of the lasso. *Bernoulli*, 23(1):552–581.

de Jong, S., Boks, M. P., Fuller, T. F., Strengman, E., Janson, E., de Kovel, C. G., Ori, A. P., Vi, N., Mulder, F., Blom, J. D., et al. (2012). A gene co-expression network in whole blood of schizophrenia patients is independent of antipsychotic-use and enriched for brain-expressed genes. *PloS one*, 7(6).

De Rubeis, S., He, X., Goldberg, A. P., Poultney, C. S., Samocha, K., Cicek, A. E., Kou, Y., Liu, L., Fromer, M., Walker, S., et al. (2014). Synaptic, transcriptional and chromatin genes disrupted in autism. *Nature*, 515(7526):209–215.

Dobriban, E. (2018). Flexible multiple testing with the FACT algorithm. *arXiv preprint arXiv:1806.10163*.

Dong, S., Walker, M. F., Carriero, N. J., DiCola, M., Willsey, A. J., Adam, Y. Y., Waqar, Z., Gonzalez, L. E., Overton, J. D., Frahm, S., et al. (2014). *De novo* insertions and deletions of predominantly paternal origin are associated with autism spectrum disorder. *Cell reports*, 9(1):16–23.

Donoho, D. and Stodden, V. (2004). When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, pages 1141–1148.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455.

Duembgen, L. and Walther, G. (2008). Multiscale inference about a density. *The Annals of Statistics*, 36(4):1758–1785.

Durif, G., Modolo, L., Mold, J., Lambert-Lacroix, S., and Picard, F. (2017). Probabilistic count matrix factorization for single cell expression data analysis. In *Research in Computational Molecular Biology*, page 254. Springer.

Eckley, I., Fearnhead, P., and Killick, R. (2011). Analysis of changepoint models. In Barber, D., Cemgil, T., and Chiappa, S., editors, *Bayesian Time Series Models*, chapter 10, pages 205–224. Cambridge University Press, Cambridge.

Fan, J., Liu, H., Wang, W., and Zhu, Z. (2018a). Heterogeneity adjustment with applications to graphical model inference. *Electronic journal of statistics*, 12(2):3908.

Fan, J., Wang, W., and Zhong, Y. (2018b). An ℓ_∞ eigenvector perturbation bound and its application to robust covariance estimation. *Journal of Machine Learning Research*, 18(207):1–42.

Feng, C., Wang, H., Han, Y., Xia, Y., and Tu, X. M. (2013). The mean value theorem and Taylor's expansion in statistics. *The American Statistician*, 67(4):245–248.

Ferdous, T. and Ullah, M. (2017). An overview of rna-seq data analysis. *Journal of Biology and Life Science*, 8:57.

Fithian, W., Sun, D., and Taylor, J. (2014). Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*.

Fithian, W., Taylor, J., Tibshirani, R., and Tibshirani, R. J. (2015). Selective sequential model selection. arXiv: 1512.02565.

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580.

Fryzlewicz, P. (2007). Unbalanced Haar technique for nonparametric function estimation. *Journal of the American Statistical Association*, 102(480):1318–1327.

Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281.

Fryzlewicz, P. et al. (2018). Tail-greedy bottom-up data decompositions and fast multiple change-point detection. *The Annals of Statistics*, 46(6B):3390–3421.

Funke, T. and Becker, T. (2019). Stochastic block models: A comparison of variants and inference methods. *PloS one*, 14(4):e0215296.

Ge, R., Jin, C., and Zheng, Y. (2017). No spurious local minima in nonconvex low rank problems: A unified geometric analysis. pages 1233–1242.

Gillis, N. (2017). Introduction to nonnegative matrix factorization. *arXiv preprint arXiv:1703.00663*.

Greene, C. S., Krishnan, A., Wong, A. K., Ricciotti, E., Zelaya, R. A., Himmelstein, D. S., Zhang, R., Hartmann, B. M., Zaslavsky, E., and Sealfon, S. C. (2015). Understanding multicellular function and disease with human tissue-specific networks. *Nature genetics*.

Gunasekar, S., Ravikumar, P., and Ghosh, J. (2014). Exponential family matrix completion under structural constraints. In *International Conference on Machine Learning*, pages 1917–1925.

Guntuboyina, A., Lieu, D., Chatterjee, S., Sen, B., et al. (2020). Adaptive risk bounds in univariate total variation denoising and trend filtering. *The Annals of Statistics*, 48(1):205–229.

Hahn, G. (2018). Closure properties of classes of multiple testing procedures. *AStA Advances in Statistical Analysis*, 102(2):167–178.

Hao, N., Niu, Y. S., and Heping, Z. (2013). Multiple Change-Point Detection via a Screening and Ranking Algorithm. *Statistical Sinica*, 23(4):1553–1572.

Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406):502–516.

He, X., Sanders, S. J., Liu, L., De Rubeis, S., Lim, E. T., Sutcliffe, J. S., Schellenberg, G. D., Gibbs, R. A., Daly, M. J., Buxbaum, J. D., et al. (2013). Integrated model of *de novo* and inherited genetic variants yields greater power to identify risk genes. *PLoS Genetics*, 9(8):e1003671.

Hicks, S. C., Townes, F. W., Teng, M., and Irizarry, R. A. (2017). Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*.

Hoefling, H. (2010). A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098.

Horvath, L. and Rice, G. (2014). Extensions of some classical methods in change point analysis. *TEST*, 23(2):219–255.

Hyun, S., G'Sell, M., and Tibshirani, R. J. (2018a). Exact post-selection inference for the generalized lasso path. *Electronic Journal of Statistics*, pages 1053–1097.

Hyun, S., Lin, K. Z., G'Sell, M., and Tibshirani, R. J. (2018b). Post-selection inference for changepoint detection algorithms with application to copy number variation data. *arXiv preprint arXiv:1812.03644*.

Ieva, F., Paganoni, A. M., and Tarabelloni, N. (2016). Covariance-based clustering in multivariate and functional data analysis. *The Journal of Machine Learning Research*, 17(1):4985–5005.

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435.

Jain, P., Netrapalli, P., and Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM.

Jandhyala, V., Fotopoulos, S., Macneill, I., and Liu, P. (2013). Inference for single and multiple change-points in time series. *Journal of Time Series Analysis*, 34(4):423–446.

Johnstone, I. M. (2015). *Gaussian Estimation: Sequence and Wavelet Models*. Cambridge University Press. Draft version.

Kang, H. J., Kawasawa, Y. I., Cheng, F., Zhu, Y., Xu, X., Li, M., Sousa, A. M., Pletikos, M., Meyer, K. A., Sedmak, G., et al. (2011). Spatio-temporal transcriptome of the human brain. *Nature*, 478(7370):483–489.

Kanner, L. et al. (1943). Autistic disturbances of affective contact. *Nervous child*, 2(3):217–250.

Kao, J., Salari, K., Bocanegra, M., Choi, Y.-L., Girard, L., Gandhi, J., Kwei, K. A., Hernandez-Boussard, T., Wang, P., Gazdar, A. F., et al. (2009). Molecular profiling of breast cancer cell lines defines relevant tumor models and provides a resource for cancer gene discovery. *PloS one*, 4(7):e6146.

Kessaris, N., Fogarty, M., Iannarelli, P., Grist, M., Wegner, M., and Richardson, W. D. (2006). Competing waves of oligodendrocytes in the forebrain and postnatal elimination of an embryonic lineage. *Nature neuroscience*, 9(2):173.

Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. (2009). ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360.

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A., and Kirschner, M. W. (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201.

Kundu, A., Bach, F., and Bhattacharyya, C. (2017). Convex optimization over intersection of simple sets: improved convergence rate guarantees via an exact penalty approach. *arXiv preprint arXiv:1710.06465*.

Lafond, J. (2015). Low rank matrix completion with exponential family noise. In *Conference on Learning Theory*, pages 1224–1243.

Lai, T. L., Xing, H., and Zhang, N. (2008). Stochastic segmentation models for array-based comparative genomic hybridization data analysis. *Biostatistics*, 9(2):290–307.

Lai, W. R., Johnson, M. D., Kucherlapati, R., and Park, P. J. (2005). Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, 21(19):3763–3770.

Lamas, J. R., Tornero-Esteban, P., and Fernández-Gutiérrez, B. (2012). Therapeutic potential of mscs in musculoskeletal diseases (osteoarthritis). *Tissue reiteration–From basic biology to clinical application*, page 261.

Ledoux, M. and Talagrand, M. (2013). *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media.

Lee, J. D., Sun, D. L., Sun, Y., Taylor, J. E., et al. (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet*, 3(9):e161.

Lei, J. (2018). Network representation using graph root distributions. *arXiv preprint arXiv:1802.09684*.

Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237.

Li, T., Levina, E., and Zhu, J. (2016). Network cross-validation by edge sampling. *arXiv preprint arXiv:1612.04717*.

Lin, K. Z., Lei, J., and Roeder, K. (2020a). Exponential-family embedding with application to cell developmental trajectories for single-cell RNA-seq data.

Lin, K. Z., Liu, H., and Roeder, K. (2020b). Covariance-based sample selection for heterogeneous data: Applications to gene expression and autism risk gene detection. *Journal of the American Statistical Association*, (To appear):1–22.

Lin, K. Z., Sharpnack, J., Rinaldo, A., and Tibshirani, R. J. (2017). A sharp error analysis for the fused lasso, with application to approximate changepoint screening. In *Advances in Neural Information Processing Systems*, pages 6884–6893.

Liu, F., Choi, D., Xie, L., and Roeder, K. (2018a). Global spectral clustering in dynamic networks. *Proceedings of the National Academy of Sciences*, 115(5):927–932.

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. (2012). High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, H., Lafferty, J., and Wasserman, L. (2009). The Nonparanormal: Semiparametric estimation of high-dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328.

Liu, L., Lei, J., and Roeder, K. (2015). Network assisted analysis to reveal the genetic basis of autism. *The Annals of Applied Statistics*, 9(3):1571–1600.

Liu, L., Lei, J., Sanders, S. J., Willsey, A. J., Kou, Y., Cicek, A. E., Klei, L., Lu, C., He, X., and Li, M. (2014). DAWN: A framework to identify autism genes and subnetworks using gene expression and genetics. *Mol Autism*, 5:22.

Liu, L. T., Dobriban, E., Singer, A., et al. (2018b). *e*PCA: High dimensional exponential family PCA. *The Annals of Applied Statistics*, 12(4):2121–2150.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550.

Ma, C., Wang, K., Chi, Y., and Chen, Y. (2018). Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion, and blind deconvolution. *Foundations of Computational Mathematics*, pages 1–182.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

Maezika, M. (2016). The singular value decomposition and low rank approximation.

Mammen, E. and van de Geer, S. (1997). Locally adaptive regression splines. *The Annals of Statistics*, 25(1):387–413.

Marques, S., van Bruggen, D., Vanichkina, D. P., Floriddia, E. M., Munguba, H., Väremo, L., Giacomello, S., Falcão, A. M., Meijer, M., Björklund, Å. K., et al. (2018). Transcriptional convergence of oligodendrocyte lineage progenitors during development. *Developmental cell*, 46(4):504–517.

Marques, S., Zeisel, A., Codeluppi, S., van Bruggen, D., Falcão, A. M., Xiao, L., Li, H., Häring, M., Hochgerner, H., Romanov, R. A., et al. (2016). Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. *Science*, 352(6291):1326–1329.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462.

Menn, B., Garcia-Verdugo, J. M., Yaschine, C., Gonzalez-Perez, O., Rowitch, D., and Alvarez-Buylla, A. (2006). Origin of oligodendrocytes in the subventricular zone of the adult brain. *Journal of Neuroscience*, 26(30):7907–7918.

Muller, P., Parmigiani, G., and Rice, K. (2006). FDR and Bayesian multiple comparisons rules. In *Bayesian Statistics*, volume 8. Oxford University Press.

Mullighan, C. G., Goorha, S., Radtke, I., Miller, C. B., Coustan-Smith, E., Dalton, J. D., Girtman, K., Mathew, S., Ma, J., Pounds, S. B., et al. (2007). Genome-wide analysis of genetic alterations in acute lymphoblastic leukaemia. *Nature*, 446(7137):758.

Newville, J., Jantzie, L. L., and Cunningham, L. A. (2017). Embracing oligodendrocyte diversity in the context of perinatal injury. *Neural regeneration research*, 12(10):1575.

Nielsen, A. M. and Witten, D. (2018). The multiple random dot product graph model. *arXiv preprint arXiv:1811.12172*.

Olshen, A., Seshan, V. E., Lucito, R., and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572.

Padilla, O. H. M., Sharpnack, J., Scott, J., , and Tibshirani, R. J. (2016). The DFS fused lasso: Linear-time denoising over general graphs. *arXiv preprint arXiv:1608.03384*.

Parikshak, N. N., Luo, R., Zhang, A., Won, H., Lowe, J. K., Chandran, V., Horvath, S., and Geschwind, D. H. (2013). Integrative functional genomic analyses implicate specific molecular pathways and circuits in autism. *Cell*, 155(5):1008–1021.

Pierson, E. and Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology*, 16(1):241.

Pollard, D. (2015). A few good inequalities.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature communications*, 9(1):284.

Romano, J. P. and Wolf, M. (2005). Exact and approximate stepdown methods for multiple hypothesis testing. *Journal of the American Statistical Association*, 100(469):94–108.

Rudin, L., Osher, S., and Faterni, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268.

Rutter, M. (1978). Diagnosis and definition of childhood autism. *Journal of autism and childhood schizophrenia*, 8(2):139–161.

Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. *Nature biotechnology*, 37(5):547.

Sanders, S. J., He, X., Willsey, A. J., Ercan-Sencicek, A. G., Samocha, K. E., Cicek, A. E., Murtha, M. T., Bal, V. H., Bishop, S. L., Dong, S., et al. (2015). Insights into autism spectrum disorder genomic architecture and biology from 71 risk loci. *Neuron*, 87(6):1215–1233.

Satterstrom, F. K., Kosmicki, J. A., Wang, J., Breen, M. S., De Rubeis, S., An, J.-Y., Peng, M., Collins, R., Grove, J., Klei, L., et al. (2020). Large-scale exome sequencing study implicates both developmental and functional changes in the neurobiology of autism. *Cell*, 180(3):568–584.

Šestan, N. et al. (2012). The emerging biology of autism spectrum disorders. *Science*, 337(6100):1301–1303.

Sharpnack, J., Rinaldo, A., and Singh, A. (2012). Sparsistency of the edge lasso over graphs. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 1028–1036.

Snijders, a. M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., Hamilton, G., Hindle, a. K., Huey, B., Kimura, K., Law, S., Myambo, K., Palmer, J., Ylstra, B., Yue, J. P., Gray, J. W., Jain, a. N., Pinkel, D., and Albertson, D. G. (2001). Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature genetics*, 29(3):263–264.

Steidl, G., Didas, S., and Neumann, J. (2006). Splines in higher order TV regularization. *International Journal of Computer Vision*, 70(3):214–255.

Street, K., Risso, D., Fletcher, R. B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: Cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 19(1):477.

Sun, Y., Zhang, N. R., Owen, A. B., et al. (2012). Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *The Annals of Applied Statistics*, 6(4):1664–1688.

Talevich, E. and Shain, A. H. (2018). Cnvkit-rna: Copy number inference from rna-sequencing data. *bioRxiv*, page 408534.

Tian, X. and Taylor, J. (2018). Selective inference with a randomized response. *Annals of Statistics*, 46(2):619–710.

Tian, X. and Taylor, J. E. (2015). Selective inference with a randomized response.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. and Wang, P. (2008). Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18–29.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323.

Tibshirani, R. J. (2017). Dykstra's algorithm, admm, and coordinate descent: Connections, insights, and extensions. In *Advances in Neural Information Processing Systems*, pages 517–528.

Tibshirani, R. J., Rinaldo, A., Tibshirani, R., and Wasserman, L. (2018). Uniform asymptotic inference and the bootstrap after model selection. *Ann. Statist.*, 46(3):1255–1287.

Tibshirani, R. J., Taylor, J., Lockhart, R., , and Tibshirani, R. (2016). Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620.

Townes, F. W., Hicks, S. C., Aryee, M. J., and Irizarry, R. A. (2017). Varying-censoring aware matrix factorization for single cell RNA-sequencing. *bioRxiv*, page 166736.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381.

Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434.

Tsourakakis, C., Bonchi, F., Gionis, A., Gullo, F., and Tsiarli, M. (2013). Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 104–112. ACM.

Tsourakakis, C. E. (2014). A novel approach to finding near-cliques: The triangle-densest subgraph problem. *arXiv preprint arXiv:1405.1477*.

Udell, M., Horn, C., Zadeh, R., Boyd, S., et al. (2016). Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118.

Umezu, Y. and Takeuchi, I. (2017). Selective inference for change point detection in multi-dimensional sequences. arXiv: 1706.00514.

van Bruggen, D., Agirre, E., and Castelo-Branco, G. (2017). Single-cell transcriptomic analysis of oligodendrocyte lineage cells. *Current opinion in neurobiology*, 47:168–175.

van de Geer, S. (1990). Estimating a regression function. *Annals of Statistics*, 18(2):907–924.

Verhaak, R. G., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., Miller, C. R., Ding, L., Golub, T., Mesirov, J. P., et al. (2010). Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer cell*, 17(1):98–110.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Vershynin, R. (2012). How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686.

Von Luxburg, U., Radl, A., and Hein, M. (2014). Hitting and commute times in large random neighborhood graphs. *The Journal of Machine Learning Research*, 15(1):1751–1798.

Vostrikova, L. (1981). Detecting 'disorder' in multidimensional random processes. *Soviet Mathematics Doklady*, 24:55–59.

Wang, J., Huang, M., Torre, E., Dueck, H., Shaffer, S., Murray, J., Raj, A., Li, M., and Zhang, N. R. (2018). Gene expression distribution deconvolution in single-cell RNA sequencing. *Proceedings of the National Academy of Sciences*, 115(28):E6437–E6446.

Wang, L., Zhang, X., and Gu, Q. (2016). A unified computational and statistical framework for nonconvex low-rank matrix estimation. *arXiv preprint arXiv:1610.05275*.

Wang, Y.-X., Sharpnack, J., Smola, A., and Tibshirani, R. J. (2017). Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105):1–41.

Wang, Z., Gu, Q., Ning, Y., and Liu, H. (2015). High dimensional em algorithm: Statistical optimization and asymptotic normality. pages 2521–2529.

Willsey, A. J., Sanders, S. J., Li, M., Dong, S., Tebbenkamp, A. T., Muhle, R. A., Reilly, S. K., Lin, L., Fertuzinhos, S., Miller, J. A., et al. (2013). Coexpression networks implicate human midfetal deep cortical projection neurons in the pathogenesis of autism. *Cell*, 155(5):997–1007.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

Xue, L. and Zou, H. (2012). Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics*, 40(5):2541–2571.

Yu, M., Gupta, V., Kolar, M., et al. (2020). Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach. *Electronic Journal of Statistics*, 14(1):413–457.

Yu, Y., Wang, T., and Samworth, R. J. (2014). A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome biology*, 18(1):174.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142.

Zhang, A., Cai, T. T., and Wu, Y. (2018). Heteroskedastic PCA: Algorithm, optimality, and applications. *arXiv preprint arXiv:1810.08316*.

Zhang, Y., Chen, K., Sloan, S. A., Bennett, M. L., Scholze, A. R., O'Keeffe, S., Phatnani, H. P., Guarnieri, P., Caneda, C., Ruderisch, N., et al. (2014). An RNA-sequencing transcriptome and splicing database of glia, neurons, and vascular cells of the cerebral cortex. *Journal of Neuroscience*, 34(36):11929–11947.

Zhao, T., Wang, Z., and Liu, H. (2015). Nonconvex low rank matrix factorization via inexact first order oracle. *Advances in Neural Information Processing Systems*.

Zhou, X. (2018). On the fenchel duality between strong convexity and Lipschitz continuous gradient. *arXiv preprint arXiv:1803.06573*.

Zhu, L., Lei, J., Klei, L., Devlin, B., and Roeder, K. (2019). Semisoft clustering of single-cell data. *Proceedings of the National Academy of Sciences*, 116(2):466–471.