ON THE INTERPLAY BETWEEN COMPUTATIONAL MODELS AND STATISTICAL CONCEPTS IN OMICS APPLICATIONS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Emery T. Goossens

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Rebecca W. Doerge, Co-Chair

Department of Statistics and Data Science & Department of Biology, Carnegie Mellon University

Dr. Vinayak Rao, Co-Chair

Department of Statistics, Purdue University

Dr. Lei Sun

Department of Statistics, University of Toronto

Dr. Lingsong Zhang

Department of Statistics & Regenstrief Center for Healthcare Engineering, Purdue University

Dr. Jennifer Neville

Department of Computer Science & Department of Statistics, Purdue University

Approved by:

Dr. Hao Zhang

Head of Department of Statistics, Purdue University

To Ayu.

ACKNOWLEDGMENTS

Throughout my studies, I have had the opportunity to take classes and discuss research with various mentors. I want to thank the amazing professors and rigorous instruction I had at the University of Toronto that brought me from zero to graduate-level understanding of statistics in two years. My instructor turned friend, Dr. Alex Shestapaloff, has given me the encouragement needed to pursue graduate studies. I was inspired by the passionate and enthusiastic teaching of Dr. David Brenner. My experience as a teaching assistant with Dr. Alison Gibbs honed my statistical intuition. The clear explanations and discussions with Dr. Keith Knight inspired me think deeply about computational statistics and pursue research in this area. I gained many useful insights into statistical genetics from Dr. Lei Sun, who has shown incredible support for research aspirations over the years.

My accomplishment would not be possible without the friends, staff, and faculty I have interacted with during my time at the Purdue University Department of Statistics. I want to thank Dave Lefevre and Doug Crabill for setting up the amazing computational resources that enable students to pursue exciting areas of research. The instruction of Dr. Vinayak Rao has been an essential component in my research into sampling algorithms. I greatly appreciate the time and effort put into the scEPC work by Dr. Julea Vlassakis, Dr. Kevin Yamauchi, Anjali Gopal, and Professor Amy Herr. Special thanks is owed to Professor Rebecca W. Doerge for providing many lessons, skills, and opportunities. Her steadfastness and candor helped me overcome many challenges throughout my Ph.D. Her instruction has given me ability to clearly present and communicate complex ideas. Her encouragement to supplement my education with outside courses and conferences instilled a passion for cutting edge research in bioinformatics. Because of her mentorship, I feel confident in my expertise yet driven to learn more.

I would also like to thank my parents for their unwaivering love and support throughout my academic career. Their advice and wisdom were invaluable during the low points and grounding during the highs. Mom, I have always appreciated your pep talks, your ability to approach challenges calmly, and your patience. Dad, I have learned a tremendous amount from you: how to strategize through life, follow opportunities, and work everyday to help people. I would also like to thank my siblings who have put up with me my entire life. Erik, our talks have helped me more than you know. I actively seek to better myself in the hopes of being someone you look up to. Andrea, you have shown me what it means to be determined and steadfast in your career while prioritizing the ones you love. Ehren, your enthusiasm of trying new things and love of exploring the world gave me the courage to up sticks and pursue my own path. I continue to be inspired by the amazingly wonderful Putu Ayu Gatrani Sudyanti. I admire your ability to stay focused, organized, and cheerful. From our marathon study sessions for the Qualifying Exams to the late nights in the office doing research, to traveling around the world, you have been an amazing companion on this journey. I am so excited for our next (non-thesis) chapter together.

TABLE OF CONTENTS

Page

LIST OF TABLES viii

LIST OF FIGURES x

ABSTRACT xv

1 INTRODUCTION 1
1.1 Introduction to Deep Learning Models 2
1.1.1 Activation Functions 4
1.1.2 Hidden Units and Hidden Layers 4
1.1.3 Convolutional Neural Networks 5
1.1.4 Loss Functions 7
1.1.5 Accounting for Class Imbalance in Deep Learning Models 7
1.2 Thematic Overview 8
1.2.1 Accounting for Technical Variation in Data Processing 8
1.2.2 Evaluating a Model in Terms of the End Use Case 9
1.2.3 Integration of Outside Data 9

2 CLASSIFICATION, SEGMENTATION, AND QUANTIFICATION OF ELECTROPHORETIC CYTOMETRY IMAGES 11
2.1 Introduction 11
2.2 Overview of Electrophoretic Protein Cytometry 13
2.2.1 Manual Curation and Quantification of scEPC Images 14
2.3 Predictive Models of scEPC Images 15
2.3.1 Classification and Segmentation of scEPC Images 15
2.3.2 Unsupervised Learning and Denoising of scEPC Images 18
2.4 scEPC Quantification Pipeline 21
2.4.1 Multi-task Model Architecture 21
2.4.2 Quantification of Model Outputs 24
2.5 Training and Evaluation of the scEPC Quantification 25
2.5.1 Comparison of Single- and Multi-task Learning 27
2.5.2 Testing Correlation of Ground Truth Data 30
2.6 Discussion 32

3 CLASSIFYING GENOMIC MUTATIONS IN CANCER DIAGNOSTICS . . 33
3.1 Somatic Mutations and Cancer 34
3.1.1 Datasets 34
3.2 Convolutional Neural Networks and Genomics 35

Page

3.2.1 Modeling Genomic Sequences with Convolutions 36
3.2.2 Interpreting Mutations via Change in Predicted Probability . . 37
3.2.3 Classifying Mutations Directly 39
3.3 Early Detection of Cancer from Raw Sequencing Reads 40
3.3.1 Existing Approaches to Cancer Diagnosis using Liquid Biopsies 42
3.3.2 Simulating Population Liquid Biopsy Results 42
3.3.3 Data Pre-processing the COSMIC Dataset 43
3.3.4 Results 45
3.4 Validating Somatic Mutations 47
3.5 Discussion 48

4 BEST ORDERED SUBSET SELECTION 50
4.1 Introduction 50
4.2 Adaptive Testing Using Exponential Order Statistics 52
4.2.1 Computing the Combined P-Value of Each Subset 52
4.2.2 Best Ordered Subset Selection 54
4.3 Controlling Type I Error of BOSS 55
4.3.1 The Covariance of Exponential Order Statistic Sums 56
4.3.2 Estimating the Effective Number of Tests 57
4.4 Efficient and Accurate Computation of BOSS 59
4.4.1 Rao-Blackwellized Monte Carlo Integration 59
4.4.2 Importance-weighted Monte Carlo Sampling 61
4.5 Applications 64
4.5.1 Combining Evidence of the Joint Location-Scale Test 65
4.5.2 Analysis of the Apical Plasma Membrane Gene-set 69
4.5.3 Pleiotropic Signal in UK Biobank Summary Statistics 70
4.6 Simulation Studies 73
4.7 Discussion 80

5 CONCLUSION 81
5.1 Summary 82
5.1.1 Accounting for Technical Variation in Data Processing 82
5.1.2 Evaluating a Model in Terms of the End Use Case 83
5.1.3 Integration of Outside Data 83
5.2 Future Work 84
5.2.1 Classification, Segmentation and Quantification of Electrophoretic Cytometry Images 84
5.2.2 Classifying Genomic Mutations in Cancer Diagnostics 85
5.2.3 Best Ordered Subset Selection 86
5.3 Novel Contributions 86

REFERENCES 87

VITA 96

LIST OF TABLES

Table Page

2.1 Single- and Multi-task learning performance on the experimental test dataset for classification, segmentation and denoising. The table shows the average and standard error of the loss across five models. The best performance on the classification task is obtained when all three tasks are trained simultaneously. On the segmentation task, the best performance is obtained when only trained on the segmentation task only and other tasks seem to worsen performance drastically. While the performance on the denoising task is best when only on this single task, the three task model performs comparably. 28

2.2 Single- and Multi-task learning performance the ground truth dataset for classification, segmentation and denoising. The table shows the average and standard error of the loss across five models. The relative model performance on the ground truth dataset is similar to that of the experimental test dataset, though the models perform better overall. Models training on both the segmentation and denoising task only have the best performance. The best classification performance is obtained when training on both the segmentation and classification task but is also comparable to the models trained on all three tasks. 29

2.3 Correlation and P-values for Ground Truth Dataset. The leftmost, orange column includes the original number of "pass" images for both the GFP and AB images, the estimated correlation, and the p-value of the test of correlation in the ground truth dataset. The green column includes the same results of the Deep Learning Quantification model, which have similar correlation estimates but more significant p-values. The added significance is a result of the Deep Learning Quantification model including more observations and thus providing more confidence in the significance of the correlation. 31

3.1 Table of Most Mutated Genes in the COSMIC Dataset [40]. The dataset used includes many well-known oncogenes with varying frequencies of substitutions, insertions, and deletions. 35

Table Page

3.2 Table of Ensemble Convolutional Neural Network Modeling Performance. This table displays the precision and recall using the average prediction of five convolutional models for each setting. Within each dataset, using a wider genomic region improves predictive performance. Within the 1000 bp setting, the use of the targeted dataset leads to higher precision whereas the total dataset leads to the highest recall. 46

3.3 Table of Ensemble Recurrent Neural Network Modeling Performance. This table displays the precision and recall using the average prediction of five recurrent models for each setting. Including more training examples with the total dataset as well as using wider genomic windows both improve model performance. 46

3.4 Table of Predicted Somatic Mutations in the Unconfirmed Cosmic Dataset [40]. This table includes the ten most highly predicted cancerous somatic mutations listed as unvalidated in the COSMIC database. This ranking can inform which mutations should be investigated to determine whether they are likely to be associated with cancer. 48

4.1 Joint Location Scale Results. The top ten most significant single nucleotide variants (SNPs) ordered by BOSS p-value, as well as a comparison with the MinPV and Fisher's methods. BOSS p-values are computed using the Rao-Blackwellized Monte Carlo (RBMC) method. The Location and Scale columns indicate p-value order in the most significant subset. For example, the most significant variant, rs11611796, includes only the scale p-value in the most significant subset. Another variant, rs2399880, includes both the location and scale p-values, albeit with the location p-value being ordered as more significant than the scale p-value. 66

LIST OF FIGURES

Figure Page

2.1 Overview of Single Cell Western Blot Workflow. Single cells are isolated from a tissue sample and seeded onto a 30-μm-thick polyacrylamide gel patterned with 30-μm-diameter microwells. Cells in microwells are then lysed and 'sieved' through the polyacrylamide gel matrix via application of an electric field. Protein targets of interest can be detected upon incubation of the polyacrylamide gel with fluorescent antibodies. In order to quantify the protein target of interest, the full polyacrylamide gel is scanned with a laser microarray scanner. With this method, protein targets from hundreds of single cells can be individually detected on a single polyacrylamide gel, thereby enabling interrogation of 'rare cell' protein isoforms. 13

2.2 Positive Class Segmentation of scEPC Images. Three examples of "pass" images with their corresponding segmentation masks predicted by a convolutional neural network. The first, third and fifth image from the left show examples of scEPC images with the segmentation mask of each image immediately to their right. 17

2.3 Positive Class Denoising of scEPC Images. Examples of "pass" images that have been denoised by a convolutional neural network. The first, third and fifth image from the left show examples of scEPC images with the denoised output of each image immediately to their right. 19

2.4 Negative Class Denoising of scEPC Images. Examples of "fail" images that have been denoised by a convolutional neural network. The first, third and fifth image from the left show examples of scEPC images with the denoised output of each image immediately to their right. 21

Figure Page

2.5 Architecture of Classification, Segmentation and Denoising Model. An encoder-decoder architecture forms the basis for this model and enables learning from large amounts of unlabeled data. The encoding learned from unlabeled data can then be used for classification (top), as well as segmentation (bottom). Because of the cost associated with obtaining labels, there typically are fewer labeled observations available for the classification task. Simultaneous training for all outputs is possible by weighting irrelevant gradient updates to zero within the loss function. Alternatively, the model can learn to denoise images by predicting a "smoothed" image. In this application, both segmentation and denoising labels must be generated using preexisting algorithms. 23

2.6 Comparison of Predicted Quantification with Existing Pipeline. Scatterplot comparing the quantification of protein expression using manual gaussian fitting and deep learning on the experimental test dataset. Points in black represent images that were classified as "pass" by both the Manual Gaussian Fitting and Deep Learning Quantification approach for which there is high correlation (r = 0.97). Points in red are images that were classified as "pass" only by the Deep Learning Quantification model and thus have zero values for the protein expression value as they were not originally quantified. Points in green were classifed as "fail" by the Deep Learning Quantification model but not by Manual Gaussian Fitting. The protein expression values of these green points are computed by Deep Learning Model though in practice they would be excluded due to their classification. 26

2.7 Comparison of Predicted Quantification with Existing Pipeline on Specific Experiments. Scatter plots of GFP protein and antibody (AB) quantification from the Deep Learning Quantification model. Overall, the protein expression values are highly correlated as expected. In the lower right plot GFP and AB at 5mug concentration there is a clear outlier where the model has failed to provide an accurate prediction. 31

3.1 The input data includes the whole reference genomic window as well as a subset of the this window with a cancerous or non-cancerous mutation. Depicted here are three examples of reference genomes with their corresponding mutations. 41

Figure Page

4.1 Estimating the Effective Number of Tests. This plot shows various ways of computing the effective number of tests of BOSS using the covariance matrix of the exponential order statistics. Due to the fact that the combined p-values from each of the ordered subsets are highly correlated, the effective number of tests should be quite small relative to the total number of tests considered. The proposed method, which effectively controls the Type I error rate, is based on [94] with additional scaling determined by the two test case. 58

4.2 Comparison of Methods of Combining P-Values. Scatterplots of the negative log10 p-values from BOSS and MinPV (Left), BOSS and Fisher's (Center), and Fisher's and MinPV (Right). The BOSS p-values are computed using the RBMC method, where the legend indicates the number of variables selected in the best ordered subset. While only one variant is statistically significant (when controlling for the number of tests considered) using the MinPV method, there are many variants that are one or two orders magnitude more significant when using BOSS compared to the MinPV. There is less of a difference in p-values when considering Fisher's method compared to BOSS as there are only two ordered subsets being considered. By using BOSS, it is possible to eliminate the choice of whether to use the MinPV method or Fisher's method, which often differ dramatically. 67

4.3 Comparison of Methods of Computing BOSS P-Values. Scatterplots of the BOSS negative log10 p-values using Rao-Blackwellized Monte Carlo (RBMC) and vanilla Monte Carlo (MC) methods (Left), Rao-Blackwellized Monte Carlo and Importance Sampling (IS) (Center), and vanilla Monte Carlo and Importance Sampling (Right) for the joint Location-Scale test data. For these results, the RBMC method used only $1,000$ samples whereas the MC and IS methods used $10,000,000$. The IS methods uses a variance scaling parameter of 1.1. While the estimated BOSS p-values are quite similar for most p-values, the MC method is not able estimate the most extreme BOSS p-value (estimated as $1.99e-8$ using IS and $2.96e-8$ using RBMC), which is set to $1e-7$ in the plots above as no null distribution samples were more extreme. These results show that the use of RBMC allows for estimating extreme BOSS p-values with relative few iterations. Additionally, the use of IS can estimate p-values more extreme (by order of magnitude here) than the vanilla Monte Carlo approach. . . . 68

Figure Page

4.4 Simulation Study of Null Test Statistics. In order to ensure that the BOSS p-values are uniformly distributed, the first step simulates test statistics with zero mean and covariance identical to the UK Biobank data. A total of 10 million importance weighted samples with a variance scaling of 1.05 are used to estimate the BOSS p-values. (Left) The quantile plot of BOSS p-values resulting from the ~ 1.3 null test statistics illustrates that the null p-values are uniformly distributed. (Center) The number of variables contributing to most significant combination of null p-values. (Right) The relationship between the simulated p-values and importance sampling weights suggest the importance sampling distribution does not result in numerically unstable behavior. 72

4.5 BOSS Results of UK Biobank P-Values. The same 10 million importance weighted samples with a variance scaling of 1.05 were used to estimate the BOSS p-values for the actual UK Biobank test statistics. (Left) The quantile plot of BOSS p-values resulting from the ~ 1.3 million UK Biobank test statistics shows that there is inflation of the negative $\log_{10}$ p-values. (Center) The number of variables contributing to the most significant combination of UK Biobank p-values. (Right) The relationship between the simulated p-values and importance sampling weights are shown again to emphasize that the same set of simulated p-values and importance weights were used to compute the BOSS p-values. 73

4.6 Lower Quantile of the Null Distribution. Inspecting the null distribution quantile plot shows that the BOSS method using RBMC is slightly conservative above the 0.05 cutoff. 75

4.7 Comparison of Null Distributions from Combining Independent P-values. The quantile plots of BOSS, Fisher's and the MinPV method all have tail distributions consistent with the uniform distribution. While Fisher's and the Min PV method are uniformly distributed in the histograms, the scaling of the RBMC BOSS p-values result in many p-values close to one. 76

4.8 Comparison of Null Distributions from Combining Dependent P-values. The quantile plots and histograms of the BOSS, Fisher's and the MinPV methods using importance sampling all have null distributions consistent with the uniform distribution. 77

4.9 Correlation Matrix of Thirty Variants in the Apical Gene Set. This figure shows the correlation matrix used for the simulation of dependent p-values and analysis of the null distributions of the BOSS, Fisher's, and Min PV methods using importance sampling. 78

4.10 Simulation Study Assessing Power. By simulating non-null relationships between a response variable and thirty explanatory variables with a fixed amount of explained variation, we can compare the power of the various methods. While the BOSS, Fisher's, and Min PV methods use the marginal p-values resulting from a linear model including one parameter at a time, the F-test uses a linear model with all parameters at once. These results indicate that power of each method depends on the percentage of non-null parameters in the model. While BOSS has slightly less power compared to Fisher's method and the F-test when the non-null signal is spread across many parameters, it outperforms other methods when the signal is spread across relatively few parameters. 79

ABSTRACT

Goossens, Emery T. Ph.D., Purdue University, December 2019. On the Interplay Between Computational Models and Statistical Concepts in Omics Applications. Major Professors: Rebecca W. Doerge, Vinayak Rao.

Technological advancements have lead to the generation of enormous amounts of data. In order to capitalize on this trend, however, both computational and statistical challenges must be tackled. While computational efficiency is important, interpretability of models and algorithms are essential to ensuring the validity of any conclusions drawn. Nowhere is this more clear than in the case of biomedical data, where inferences drawn from large datasets are used to inform future directions of research, diagnose diseases, and generate leads for the development of new pharmaceuticals. This work examines the interplay between statistical concepts and computational models in three applications. Specifically, quantifying protein expression of fluorescent images, classifying somatic mutations in cancer, and combining p-values computed from genomic summary statistics. Across these applications, there are three recurring themes: accounting for technical and biological variation in data processing, evaluating the performance of a model in its end use case, and integrating results with outside data. Within these applications and themes, many statistical concepts are employed including Bayes theorem, and type I error rate control alongside computational models such a convolutional neural networks and Monte Carlo sampling algorithms. The results of these investigations inform much broader application areas such as biomedical imaging, modeling genomic sequences, and hypothesis testing in high-dimensions. Specific contributions in the application of Convolutional Neural Networks include demonstrating their ability to replicate the quantification of protein expression images from various manually-generated or deterministic label sets as well as the creation of a modeling framework for sequencing-based cancer diagnostics and

the prioritization of unvalidated somatic mutations. In the area of hypothesis testing, novel algorithms are proposed that enable the use of a powerful and interpretable technique of combining p-values in the large-scale setting of genome-wide association studies.

1. INTRODUCTION

The amount of data being collected world-wide is staggering. The diversity of settings and manner in which these data are obtained vary greatly; biological data are no exception. Although large datasets are being collected, the challenge of employing these data for the purpose of answering, in this case biological questions, is a continued challenge. One main issue is that obtaining data is only the first step in the process. Because of data complexity and high amount of human involvement, the current bottleneck in scientific research is preparing and processing data for analysis and, most importantly, appropriately interpreting the results.

While entirety of this work is motivated by biology, much of what is developed has general application outside of the field. Advances in high-throughput technologies (e.g., next-generation sequencing, microfluidic cell isolation, immunohistochemistry) are allowing a deeper, more complex level of data to be obtained and analyzed. The hope is that more data, at a level never before seen, will allow scientists to answer questions that so far have remained unobtainable. That said, obtaining useful insights from current, large biology data requires a certain amount of data processing. Data processing here refers to a larger class of techniques and methods that can prepare the data from analysis. The analysis itself depends on the statistical and computational models and algorithms that have to be developed and assessed for accuracy and related performance metrics. Once data are ready and the model established, the model has to be validated using outside resources. For these reasons, the importance of accurate statistical models and robust computational algorithms for scientific inquiry is the main motivation of this research.

Statistics has always played a prominent role in science. The increasing size and complexity of data in many scientific fields (e.g., proteomics, oncology, and genomics), however, demands that statistical concepts and techniques be specifically tailored to

accommodate modern applications. While there has been significant work in developing hierarchical linear models for complex data with relatively few observations, these approaches have often been replaced by neural networks due to their ability to model non-linear relationships in large, high-dimensional data. Over the last 20 years, there has also been theoretical advancements in areas such as multiple hypothesis testing. But these approaches must be accompanied by computationally-scalable software implementations that are robust in estimating p-values at the extremes that millions of statistical tests require. What more, large sets of p-values are often used to summarize and share experimental results due to the shear size of high-throughput experiments and privacy concerns. While the statistical theory of hypothesis testing is obviously fundamental a component of science, modern scientific inquiry increasingly consists of the analysis and interpretation of p-values in aggregate.

Classical statistical models are known to have limited success when applied to the extremely complex and unstructured data such as images and text. More often than not, this is because of both computational constraints and the violation of distributional assumptions, respectively. Statistical models rely on theoretical assumptions that allow one to test specific hypotheses about the association of variables. Computational models can handle more complex data by relying on fewer assumptions but are less amenable to statistical inference, making the rationale behind the results more difficult to interpret. On their own, each model type may be insufficient provide the answer to a challenging question. Together, the interplay between statistical and computational models has great potential to offer robust and efficient inference on large, complex datasets.

1.1 Introduction to Deep Learning Models

A class of computational models known as deep learning has met success in the analysis of complex, unstructured data (e.g., images and text) [1]. While lacking interpretability and requiring significant hyperparameter tuning, computational models

are more efficient, rely on fewer assumptions, and are more flexible in their application. With this as motivation, statistical concepts in the context of deep learning are explored in the design, training, and interpretation of computational models. The overarching goal of this work is to develop computational models from large, complex, unstructured biomedical data. While the details of deep learning models have been extensively covered elsewhere [2], the basic theoretical foundations are included here for completeness.

Neural networks are a general class of models that use a series of non-linear functional compositions of an input to predict an output [2]. First, model some or all of the input variables via a linear function that is later transformed with a non-linear function, known as an activation function. A neural network consists of many separate non-linear transformations of a linear combinations of inputs, referred to as hidden units. Collectively, these hidden units are referred to as hidden layers [3]. The output of hidden units in the first layer can then be combined linearly and subsequently non-linearly transformed in successive hidden layers. Deep learning refers the specification, and more importantly the optimization, of neural network models with many hidden layers. For simplicity, neural networks will henceforth be referred to as deep learning models.

Deep learning models consist of layers of compositional functions. Generally speaking, the number of hidden units in each hidden layer, as well as the number and type of hidden layers in a model is referred to as the architecture of the model. There are three general types of deep learning layers, that can be combined in various ways within one model. The first type is the fully connected layer characterized by a linear combination, and subsequent non-linear transformation, of all inputs (or of all outputs of a previous layer) [2]. Convolutional layers consist of smaller models, known as convolutional filters, of only a subset of the input data at a time [1]. These convolutional filters are applied sequentially to all subsets of the input, producing an output of similar form or shape. Finally, recurrent layers are used to model data with long-term sequential dependencies such as text [2].

1.1.1 Activation Functions

Activation functions are a core component of any deep learning model [3]. Essentially, their purpose is to non-linearly transform the weighted linear combination of output from the previous layer. While early neural networks relied on sigmoid and tanh functions, modern deep learning models use different activation functions due to their performance both in terms of optimization consistency and overall prediction. One of the most commonly used activation functions is the rectified linear unit (ReLU) [4]

$$f(t) = \begin{cases} t, & \text{if } t \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Another example of an activation function is the exponential linear unit (ELU) [5]

$$f(t) = \begin{cases} t, & \text{if } t \geq 0 \\ exp(t) - 1, & \text{otherwise.} \end{cases}$$

1.1.2 Hidden Units and Hidden Layers

Activation functions are used to non-linearly transform the weighted linear combination of outputs from a previous layer. Collectively, a set of hidden units at the same location within the model hierarchy is referred to as a hidden layer. In the first hidden layer, the linear combination of input variables are non-linearly transformed via an activation function. More specifically, let $\mathbf{x_i} = (1, x_{1i}, \ldots, x_{pi})$ denote a column vector of inputs for observations i where p is the dimension of the data. Then the

output of a single hidden unit $u_0 = 1, \ldots, U_0$ in the input layer ($l = 0$) is given by $h^1_{u_1} = f(z^1_{u_1})$ where

$$z^1_{u_1} = \mathbf{x}'\mathbf{w_{u_0}}$$

and $\mathbf{w_{u_0}}$ is a column vector of parameters for the input layer. Note that when $l = 0$, the linear combination is dot product of the input data $\mathbf{x}$ and a set of unique weights for each hidden unit. In subsequent layers, $\mathbf{x}$ is replaced with the output of the hidden units in the previous layer:

$$z^l_{u_l} = \mathbf{z^{l-1\prime}w^{l-1}_{u_{l-1}}}, l > 1$$

where $\mathbf{w^{l-1}_{u_{l-1}}}$ is a unique set of weights for each hidden unit $u_l = 1, \ldots, U_l$ in hidden layer l.

1.1.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are useful for certain types of data that have an inherent spatial or ordered structure such as images or text, respectively [1]. The fundamental component of CNNs is the convolutional operator. In the first layer of a CNN, the hidden unit values consist of non-linear transformations of the convolutional operators at each position of the input space.

One-Dimensional Convolutions

When dealing with one dimensional data, such as genomic sequences, each input observation $i = 1, \ldots, n$ consists of an ordered vector x^t of fixed length $t = 1, \ldots, T$: $(x^1, \ldots, x^T)$. In contrast to standard deep learning models, the output of each one-dimensional convolution is a vector rather than a scalar for each hidden unit $u_l =$

$1, \ldots, U_l$. Applying a convolutional filter of length K with weights $(w_{u_l}^1, \ldots, w_{u_l}^K)$ to this vector results in the matrix product:

$$\begin{aligned}\begin{bmatrix} z_{u_l}^1 & \ldots & z_{u_l}^{T-K+1} \end{bmatrix} &= \begin{bmatrix} x^1 & \ldots & x^T \end{bmatrix} * \begin{bmatrix} w_{u_0}^1 & \ldots & w_{u_0}^K \end{bmatrix} \\ &= \begin{bmatrix} (w_{u_0}^1 x^1 + \ldots + w_{u_0}^K x^K) & \ldots & (w_{u_0}^1 x^{T-K+1} + \ldots + w_{u_0}^K x^T) \end{bmatrix}\end{aligned}$$

where each output z_u^t is non-linearly transformed by an activation function f into the output of a hidden unit $h_{u_l}^{l,t} = f(z_{u_{l-1}}^t)$ at layer l and location t. The output of each hidden layer is then used as an input into the next layer $l+1$ in a similar manner.

Two-Dimensional Convolutions

Two dimensional data, such as black and white images, have input observations $i = 1, \ldots, n$ consisting of an spatially ordered matrix $x^{r,c}$ with rows $r = 1, \ldots, R$ and columns $c = 1, \ldots, C$. The difference between two-dimensional convolutions and one-dimensional convolutions is that the output of each hidden unit is a matrix, rather a vector, corresponding to each spatial location of the input. Applying a convolutional filter of size (K, J) with weights $w_{u_0}^{k,j}$ $k = 1, \ldots, K$ and $j = 1, \ldots, J$ to this matrix results in the matrix product for each hidden unit $u_0 = 1, \ldots, U_0$ at layer $l = 0$:

$$\begin{bmatrix} z_{u_1}^{1,1} & \ldots & z_{u_1}^{1,C-J+1} \\ \vdots & \ddots & \vdots \\ z_{u_1}^{R,1} & \ldots & z_{u_1}^{R-K+1,C-J+1} \end{bmatrix} = \begin{bmatrix} x^{1,1} & \ldots & x^{1,C} \\ \vdots & \ddots & \vdots \\ x^{R,1} & \ldots & x^{R,C} \end{bmatrix} * \begin{bmatrix} w_{u_0}^{1,1} & \ldots & w_{u_0}^{1,J} \\ \vdots & \ddots & \vdots \\ w_{u_0}^{K,1} & \ldots & w_{u_0}^{K,J} \end{bmatrix}$$

$$= \begin{bmatrix} (w_{u_0}^{1,1} x^{1,1} + \ldots + w_{u_0}^{K,J} x^{K,J}) & \ldots & (w_{u_0}^{1,1} x^{R-K+1,C-J+1} + \ldots + w_{u_0}^{K,J} x^{K,C}) \\ \vdots & \ddots & \vdots \\ (w_{u_0}^{1,1} x^{R-K+1,1} + \ldots + w_{u_0}^{K,J} x^{R,J}) & \ldots & (w_{u_0}^{1,1} x^{R-K+1,C-J+1} + \ldots + w_{u_0}^{K,J} x^{R-K+1,C-J+1}) \end{bmatrix}$$

where each output $z_{u_0}^{r,c}$ is non-linearly transformed by a function f into the output of a hidden unit $h_{u_1}^{l,t} = f(z_{u_0}^{r,c})$. This process continues at each subsequent layer l and corresponding location $(r, c) \in (1, \ldots, R_l, 1, \ldots, C_l)$ where R_l, C_l are the spatial dimensions of layer l.

1.1.4 Loss Functions

The process of repeated linear combination and subsequent non-linear transformation is continued until the final layer of a deep learning model that is used to predict an output $\mathbf{y}$. Given a set of predictions $\hat{\mathbf{y}}$, the loss is computed according to the assumed distribution of errors. In the case of an output variable taking continuous values on $\mathbb{R}$, the loss function can be formulated as the mean squared error:

$$L(y, \hat{y}) = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}.$$

Different formulations of the loss are possible by specifying the distribution of the output variables, such as binary or categorical outputs.

1.1.5 Accounting for Class Imbalance in Deep Learning Models

Biological and genomic data often exhibit class imbalance, where a "positive" class occurs much less frequently than the "negative" class. Data class imbalance is a potential source of bias in deep learning model training for such data. If not explicitly accounted for in the optimization procedure, class imbalance can lead poor predictive performance. Practically speaking, when deep learning models fail to learn useful or truly discriminative features there is a failure in the ability to generalize to outside data, an indication of overfitting. Work in this area [6] has shown that one of the best approaches to improve the optimization of deep learning models on class imbalanced data is to oversample the minority class. This oversampling procedure involves including an equal number of observations from each class when updating

model parameters. Alternatively, observations of the minority class can be weighted according to their relative frequency to ensure equal importance in the parameter updates.

Once a model has been optimized with proper accounting of the class imbalance, it is necessary to adjust predictions of a model according to the prior class probability. This adjustment is done via Bayes theorem. Given a prediction of a class for a given observation $\hat{c}$ and a class c, the adjusted probability of a class $p(c|\hat{c})$ is given by:

$$p(c|\hat{c}) = \frac{p(\hat{c}|c) \times p(c)}{\sum_{c \in C} p(\hat{c}|c) \times p(c)}$$

where $p(c)$ is the prior class probability and $p(\hat{c}|c)$ is the unadjusted class prediction.

1.2 Thematic Overview

Here the interplay between statistical and computational concepts in three applications are examined. These applications include quantifying protein expression from images, classifying somatic mutations in cancer, and testing non-traditional hypotheses using genomic summary statistics. Throughout these applications and as mentioned earlier, there are three recurring themes: 1) accounting for technical variation in data processing, 2) evaluating the performance of a model in its end use case, and 3) incorporating results with outside data.

1.2.1 Accounting for Technical Variation in Data Processing

Now that it is possible to examine individual cells via imaging, the importance of separating biological signal from technical noise is essential. Approaches such as immunohistochemistry target specific biological phenomena, such as proteins in a tissue sample or organelles within a cell, by using fluorescent dyes and antibodies [7]. Because this is a complicated, multi-step process, results can vary greatly between experiments. Perhaps the best example is next-generation sequencing (Illumina) [8],

which involves converting raw images of chemical reactions into nucleotide labels prior to alignment and further analysis. Failure to account for or directly model technical variation when processing raw data can lead to inaccurate conclusions in any subsequent analysis. Computational models and approaches are thus an essential aspect of the analysis, statistical or otherwise, of all data that result from high-throughput technology.

1.2.2 Evaluating a Model in Terms of the End Use Case

Because every model makes certain assumptions about data, and unknown sources of variation are inherent to biological datasets, performance must be throughly analyzed. At the most basic level, computational models should generalize to data that have not been used to optimize the model. While there are many metrics (e.g., mean squared error loss, accuracy, precision, or recall) to evaluate model performance [9], it is important to understand that the output of a model may be used within the context of a broader computational pipeline or as a means to test a statistical hypothesis. Good performance of a model in terms of a metric on an intermediate step does not necessarily mean that it is optimal in terms of performance on the end result. Thus, it is essential to evaluate a model in the context of the scientific use case.

1.2.3 Integration of Outside Data

Although properly designed experiments provide data that address the question at hand, additional data are often needed. Toward this end, and to gain more supporting evidence for a particular experiment, researchers often integrate outside data into their analyses. Because of privacy concerns most publicly available biomedical data are only available as summary statistics (e.g., test statistics, p-values, correlation of variants). Genome-wide association studies (GWAS) are an example of data where information available for public use is limited to summary statistics [10]. From a modeling perspective, outside data can be used to evaluate a model's performance

by ensuring that it has not overfit to a particular data type and provides useful predictions in a broader scope.

2. CLASSIFICATION, SEGMENTATION, AND QUANTIFICATION OF ELECTROPHORETIC CYTOMETRY IMAGES

2.1 Introduction

The development of micro-scale tools is unlocking new worlds of inquiry into the drivers of human health via rapid detection and quantification of small quantities of biological samples in a high-throughput manner [11]. With their characteristic small size ($\sim 10^{-6}$ m) and fast reaction times, these microfluidic devices continue to enable the development of a variety of single-cell analysis methodologies, which in turn enable the detection of complex cell signaling events that are responsible for processes such as immunity, senescence, anti-cancer drug resistance, and more [12–14]. Some key examples of these 'microfluidic' single-cell measurements include the detection of nuclear or cytoplasmic proteins (e.g., single-cell western blotting and single-cell isoelectric focusing), genomic and transcriptomic measurements (e.g., microfluidic single-cell RNA sequencing or microfluidic single-cell RT-qPCR), and live single-cell imaging [15–19]. In particular, microfluidics has been especially powerful in improving the detection of protein targets with single cell resolution; since proteins are the major determinants of cell states and phenotypes, improving targeted protein measurements is critical to improving biological enquiry [11]. [1]

A major bottleneck in 'biologically-relevant' microfluidics algorithms that interpret or quantify their output. Owing to the large volumes of data collected in high-throughput microfluidics, especially in image-based measurements (i.e., fluorescence microscopy), downstream data filtering and quantification can be computationally

[1] A previous version of this work appeared in the NIPS Machine Learning in Computational Biology Workshop, Long Beach, CA, 2017.

challenging and time-consuming. For instance, image-based microfluidic systems require complex image segmentation algorithms to differentiate output signal from background noise. Furthermore, although classifying microfluidic devices with "pass" versus "fail" signal is often aided by traditional signal processing methodologies (e.g., signal-to-noise ratio analysis), several of these classification steps often require the user to manually filter low-quality images [14], which lengthens the experimental timeline.

Due to its enormous success in other image processing applications [20], deep learning is becoming an increasingly popular tool in the analysis of biological data from microscopy and other image-based measurements. Recently, Convolutional Neural Networks (CNNs) have been utilized to segment fluorescence images of nuclei of live cells. A combination of CNNs and multiple instance learning has been used to classify and segment microscopy images without the need for segmentation masks [21,22]. Unfortunately, deep learning has made limited advancements in improving microfluidics-based measurements. Although there have been recent efforts [23] in using deep learning to aid fabrication of microfluidic devices (i.e., shaping microfluid flow), these efforts do not address the problem of improving quantification from the outputs of microfluidic devices.

Here, a general quantification framework is developed that analyzes fluorescence-based physical or biological output measurements from microfluidic devices via deep learning. The method offers four main advantages in improving the classification and quantification of microfluidic image analysis. First, classification accuracy improves with the amount of data available. Second, the use of predictive models is computational faster than other methods that require additional human involvement. Third, a semantic segmentation approach is utilized using pixel-level probabilities [24, 25] to determine whether a certain region contains signal from our biological target-of-interest. This improves the quantification accuracy of the measurement. Fourth, a denoising feature to remove technical noise is incorporated. It is anticipated that the

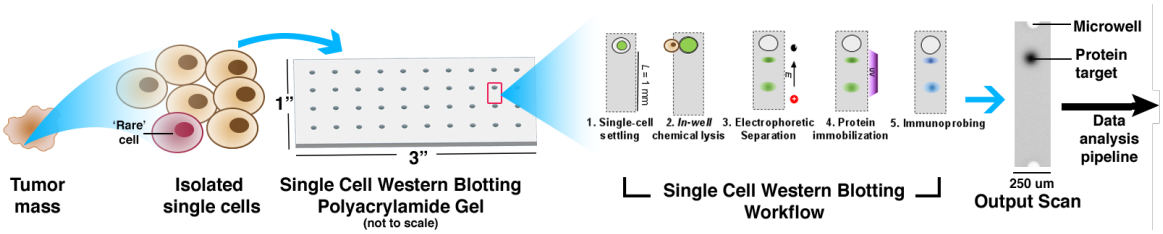

Figure 2.1.: Overview of Single Cell Western Blot Workflow. Single cells are isolated from a tissue sample and seeded onto a 30-μm-thick polyacrylamide gel patterned with 30-μm-diameter microwells. Cells in microwells are then lysed and 'sieved' through the polyacrylamide gel matrix via application of an electric field. Protein targets of interest can be detected upon incubation of the polyacrylamide gel with fluorescent antibodies. In order to quantify the protein target of interest, the full polyacrylamide gel is scanned with a laser microarray scanner. With this method, protein targets from hundreds of single cells can be individually detected on a single polyacrylamide gel, thereby enabling interrogation of 'rare cell' protein isoforms.

results from this research will provide an improved framework to address issues of accuracy, speed, and throughput of microfluidics output quantification.

2.2 Overview of Electrophoretic Protein Cytometry

A key measurement challenge in the biological sciences involves the detection of protein targets with mass or charge differences (isoforms) from 'rare cells' (i.e., cells with rare phenotypes that are present at low fractions compared to the bulk population). Such cellular sub-populations often have their phenotypes masked by the 'averaging' performed in bulk tissue assays. This point is important because small subsets of cells may confer cancer drug resistance and have numerous other significant effects in health and disease. Recently, advances in microfluidics have led to the development of a suite of tools known as single-cell electrophoretic protein cytometry ('scEPC'), which captures the contents of 20 pL cellular lysates, and performs size or charge-based separation of proteins in order to differentiate between isoforms [14–16]. Applications of scEPC have included the identification of cellular heterogeneity in neuronal stem cell differentiation, identifying mechanisms of drug resistance in cancer

(glioblastoma) cells, and identifying subpopulations from circulating tumor cells (i.e., cells present at < 1 cell/mL of blood) ([15], [14], [26]).

Size-based electrophoretic protein cytometry consists of a 30-40 micron-thick polyacrylamide (PA) gel covalently grafted onto a standard microscope slide [15] (see Figure 2.1). The PA gel is patterned with 30-μm-diameter 'microwells', into which a suspension of single cells are seeded via gravity settling. Upon application of lysis buffer and an electric field, protein lysate is extracted and injected into the polyacrylamide gel, which acts as a sieve to separate proteins based on their molecular weights. A 'multistage immunoassay' is used to detect the proteins-of-interest; the polyacrylamide gel is incubated in a solution of fluorescent antibodies, which chemically bind to specific protein targets. These protein targets are typically present as a diffuse Gaussian 'band' in a given 'separation lane', which are imaged via a laser microarray scanner [15]. Any one scEPC gel can have thousands of separation lanes, each of which can contain individual Gaussian bands for 10+ proteins-of-interest [15]. At present, identification and quantification of these Gaussian bands requires the user to manually 'filter' each individual peak, which is both cumbersome and time consuming. Towards this end, machine learning image processing has potential to be well-suited to address the current limitations associated with manual image analysis.

2.2.1 Manual Curation and Quantification of scEPC Images

Due to the novelty of the scEPC technology, converting images into useful protein expression data requires custom algorithms. The existing approach [14, 15, 27] for quantifying protein expression from the raw output requires a combination of manually selecting "pass" or "fail" images, defining a region within an image based on the assumption of gaussian protein diffusion, computing the area under the fitted gaussian curve, and finally subtracting the background intensity levels. The result is a measure of protein expression in arbitrary fluorescence units (AFUs). Although this approach works well, it is subject to human bias, may not generalize well to non-

gaussian protein diffusion, and is relatively low-throughput. Because these images, classification labels, and AFU values of various experiments are available from various experiment, it is possible to use them to train a computational model to automate these tasks thus resulting in a high-throughput quantification pipeline.

2.3 Predictive Models of scEPC Images

As introduced previously, CNNs have had great success in conventional image processing applications. More recently, neural networks consisting entirely of convolutional layers that maintain ordered spatial information during training and prediction, have improved pixel-level classification of images used in semantic segmentation [1,24]. These models are referred to as Fully Convolutional Networks (FCNs). Furthermore, related work (e.g., Segnet [25]) have successfully combined FCNs and encoder-decoder frameworks to improve segmentation results as well as denoise images [28]. Other approaches include an EM-based segmentation algorithm using bounding boxes of regions of interest rather than pixel-wise labels [29]. This related work has greatly influenced the multi-task framework presented here.

2.3.1 Classification and Segmentation of scEPC Images

In classical statistics, logistic regression is a generalized linear model with the logit link function that transforms a linear combination of the input variables to predict a zero or one label [30]. Because deep learning models have many more parameters than logisitic regression, optimization of such models requires techniques for non-convex loss functions such as Adam [31]. In the machine learning literature, logistic regression is often referred to as binary classification. Classification is a supervised learning algorithm that uses inputs such as images to model outputs a class a quantitative output corresponding to each observation. Regardless of the name, training a

model to predict the class label of an input ultimately involves minimizing the binary crossentropy loss [2]

$$L(\hat{c}, c) = \sum_{i=1}^{n} c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i) \tag{2.1}$$

where $c_i \in 0, 1$ and $\hat{c}_i = P(c_i = 1|w, x_i) = 1 - P(c_i = 0|w, x_i)$ is the probability of an input x having the "positive" class $c = 1$ given the weights of a (deep learning) model w. Note that Equation 2.1 implicitly values the loss associated with one class to be equal to the other class. Weighting each class equally is not ideal in data containing an unequal proportion of observations in the two classes. This issue, known as class imbalance, is pervasive in biological datasets, where the number of observations in the "negative" class far outnumber those in the "positive" class. As in the case of scEPC images, biological data are often only interested in finding, understanding, or quantifying the "positive" class and the "negative" class is viewed either as not valuable or as examples of technical noise.

From an optimization perspective, not addressing the class imbalance can lead to suboptimal results. A model can minimize the loss function simply by predicting the majority class without learning important, generalizable features of the data. In other words, the class imbalance results in the model finding a local optimum within the parameter space. Improved performance can be obtained by formulating the loss function as

$$L_c(\hat{c}, c) = \sum_{i=1}^{N} \frac{c_i \log(\hat{c}_i)\lambda_c + (1 - c_i) \log(1 - \hat{c}_i)}{N}$$

where

$$\lambda_c = \frac{\sum_{i=1}^{n} 1 - c_i}{\sum_{i=1}^{n} c_i}.$$

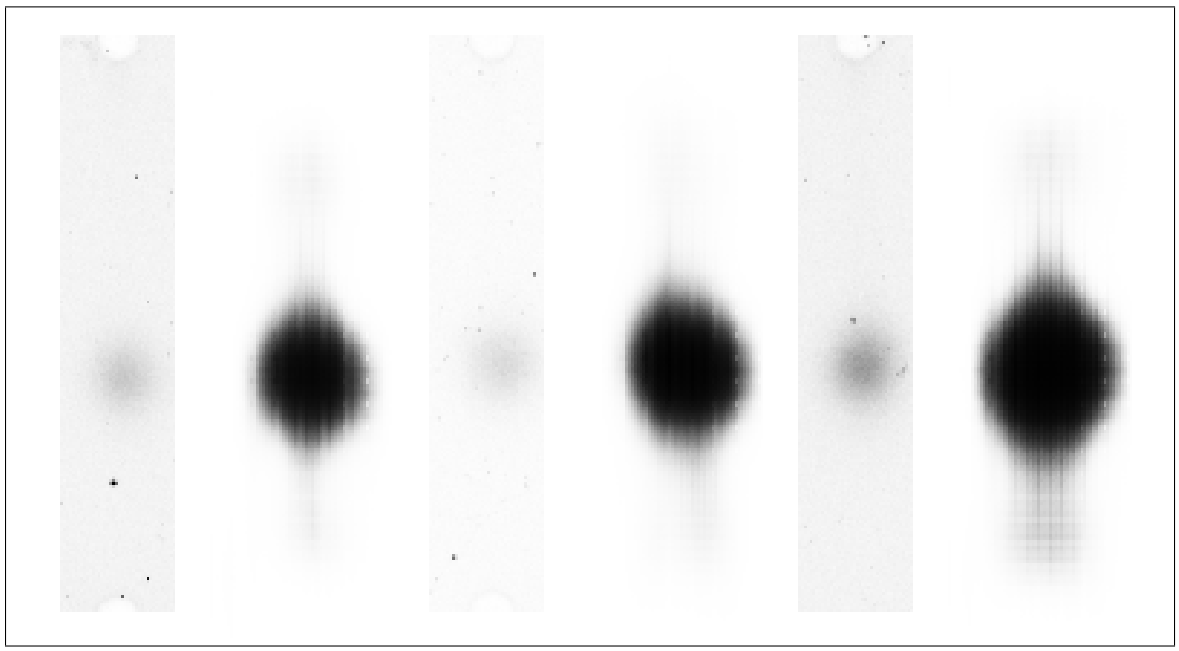

Figure 2.2.: Positive Class Segmentation of scEPC Images. Three examples of "pass" images with their corresponding segmentation masks predicted by a convolutional neural network. The first, third and fifth image from the left show examples of scEPC images with the segmentation mask of each image immediately to their right.

Alternatively, each class can be sampled equally during training. The balanced class sampling approach is less computationally and memory efficient, but is potentially beneficial as a more diverse set of minority class observations are used within each gradient update. With both strategies, however, the model cannot effectively minimize the loss by simply predicting the majority class and thus is encouraged to learn useful distinguishing characteristics between the two class.

In computer vision, segmentation refers to any algorithm used for determining which locations of an image correspond to distinct categories. Segmentation thus can be viewed as classification of each pixel within an image [24]. For scEPC images, segmentation involves classifying whether each pixel corresponds to a "protein" or "background" pixel. As with classification, the class imbalance of pixels must be accounted for as the entire dataset might contain an imbalanced number of protein pixels compared to background pixels. For image $i = 1, \ldots, N$, let J and K refer

to the number of rows and columns, respectively. Using $s_{ijk} \in \{0, 1\}$ to refer to the pixel-level class, the segmentation loss function is given as

$$L_s(\hat{s}, s) = \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{s_{ijk} \log(\hat{s}_{ijk}) \lambda_s + (1 - s_{ijk}) \log(1 - \hat{s}_{ijk})}{N \times J \times K}$$

where

$$\lambda_s = \frac{\sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} 1 - s_{ijk}}{\sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} s_{ijk}}.$$

which is then minimized using an optimization procedure such as Adam [31]. Examples of positive class images and a depiction of the learned segmentation masks are shown in Figure 2.2.

2.3.2 Unsupervised Learning and Denoising of scEPC Images

Unsupervised learning refers to using a model to obtain useful, lower-dimensional features of data without the use of additional information [2]. Autoencoders are a certain type of unsupervised learning algorithm popular within the deep learning literature [32]. Deep autoencoders consist of multi-layer models that use the same data as both inputs as well as outputs. Autoencoders generally consist of five components: the inputs x, an encoding function (encoder) $f(\cdot)$, an encoding layer z, a decoding function $g(\cdot)$, and the predicted outputs $\hat{x}$. While f and g can take any form, here they are used to refer to a CNN. Mathematically, a simplified formulation of autoencoders is given as follows:

$$z = f(x, w_e)$$
$$\hat{x} = g(z, w_d).$$

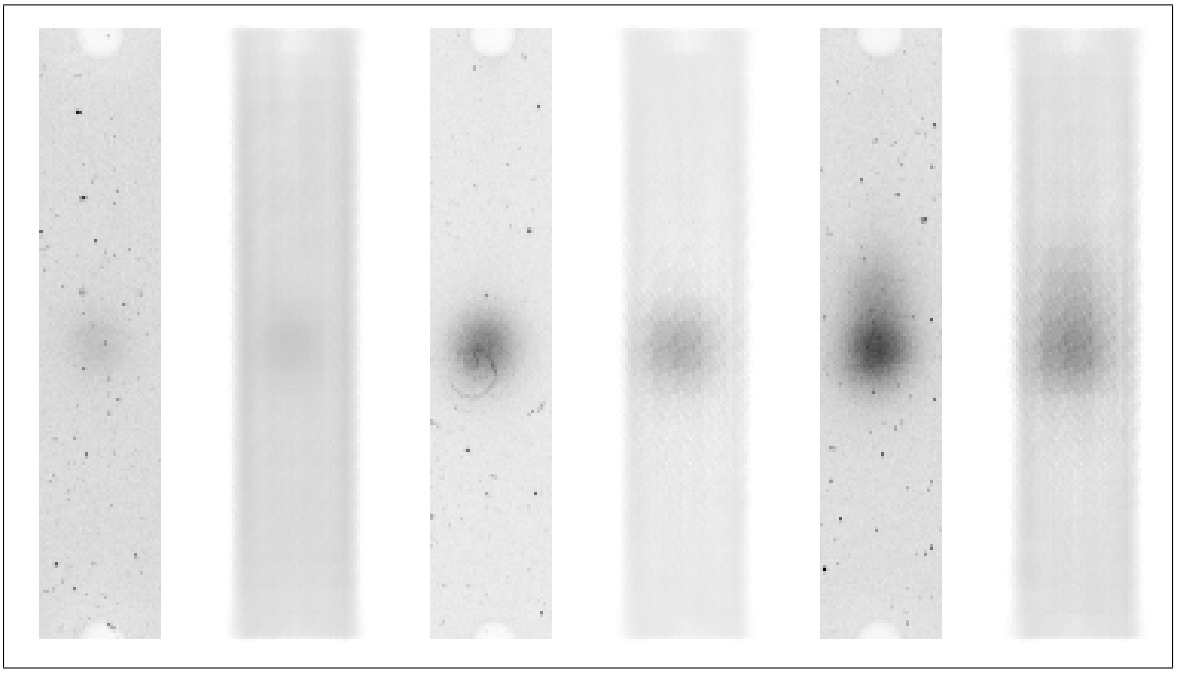

Figure 2.3.: Positive Class Denoising of scEPC Images. Examples of "pass" images that have been denoised by a convolutional neural network. The first, third and fifth image from the left show examples of scEPC images with the denoised output of each image immediately to their right.

where $\hat{x} = h(x, w_e, w_d)$ is a function of the weights (w_e, w_d) and original inputs x, where the subscripts e and d refer to 'encoding' and 'decoding,' respectively. The above model is then optimized to reduce the loss $L(\hat{x}, x)$. The autoencoder loss of scEPC images is the mean squared error between the predicted image (otherwise referred to as the reconstructed image) and the original pixel values of all images:

$$L_{ae}(\hat{x}, x) = \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{(\hat{x}_{ijk} - x_{ijk})^2}{N \times J \times K}$$

where, again, J and K refer to the number of rows and columns, respectively, of each image $i = 1, \ldots, N$. After an autoencoder has been optimized, z can then be used to visualize data in lower-dimensions or as the input into a another model such as a clustering procedure.

As mentioned previously, scEPC images consist of many "negative" class images with fewer "positive" class images. This class imbalance is a result the fact that single cells do not settle in all of the available wells. The percentage of images with protein targets is further reduced due to technical noise in that prevents accurate quantification. This work also describes a modeling framework using autoencoders to salvage images of protein targets corrupted by technical noise. The proposed approach is based on denoising autoencoders [28] and similar work using fully convolutional models for denoising images [33], but differs in a subtle way. Rather than corrupting or adding noise to an input image and training a model to predict the original image, our denoising procedure uses the original image as an input and predicts a "smoothed" version of the image as an output as shown in Figures 2.3 and 2.4. The smoothing function takes advantage of the symmetry of scEPC images, which may not be relevant in other computer vision applications. By taking the average of the original image and the horizontally flipped image, technical noise can be reduced while maintaining the total protein target signal in "pass" images. "Fail" images can also used to train a model to remove technical noise by providing the average of the original image, the horizontally flipped image, the vertically flipped image, and the horizontally as well as vertically flipped image as the predicted output. Using the above formulation of an autoencoder, the only difference is that the model now predicts the "smoothed" image $\tilde{x} = t(x)$ as the output, where t denotes the symmetric smoothing function. Thus, the loss is simply given by

$$L_d(\hat{x}, \tilde{x}) = \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{|\hat{x}_{ijk} - \tilde{x}_{ijk}|}{N \times J \times K}.$$

Taking this slightly different approach is motivated by two practical reasons. First, scEPC images already contain technical noise that we wish to remove. Second, corrupting the original image with a high level of noise alters the "pass/fail" class of the image. When operating within the multi-task framework, as described in the follow-

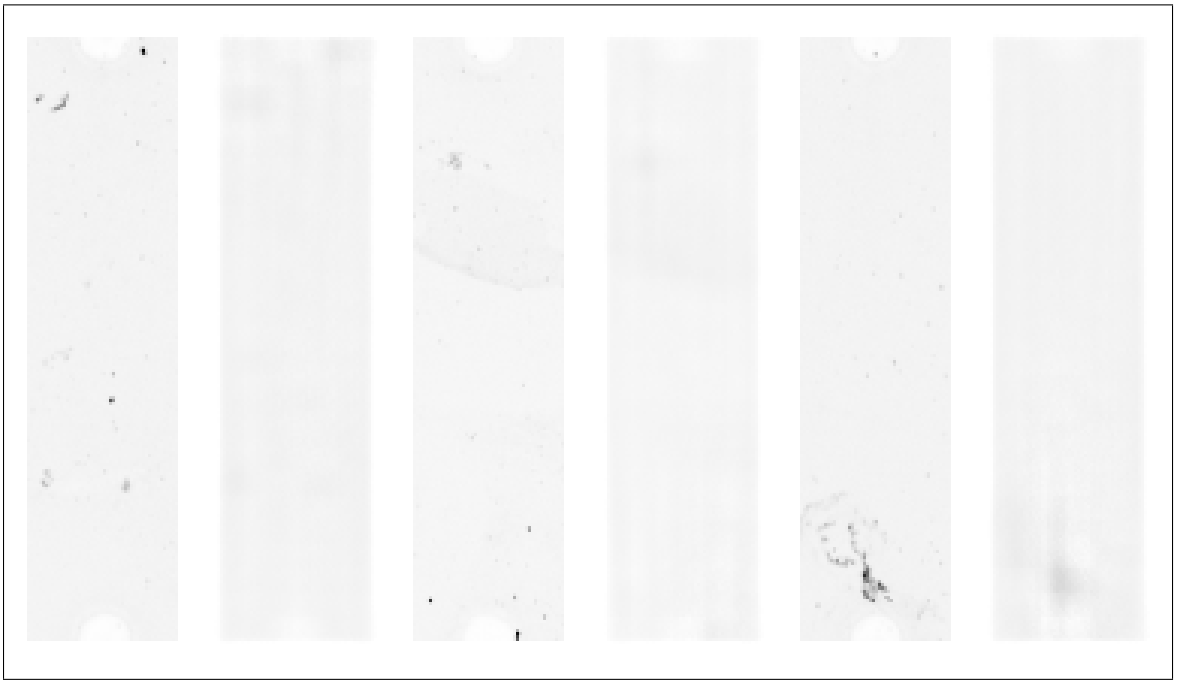

Figure 2.4.: Negative Class Denoising of scEPC Images. Examples of "fail" images that have been denoised by a convolutional neural network. The first, third and fifth image from the left show examples of scEPC images with the denoised output of each image immediately to their right.

ing section, corrupting an input with noise could provide conflicting information to a model that simultaneously predicts the class of an image as well as other outputs. This conflict arises due to the fact that many scEPC images contain protein signals that are deemed as part of the "fail" class due to high levels of technical noise.

2.4 scEPC Quantification Pipeline

2.4.1 Multi-task Model Architecture

While described separately in Section 2.3, the classification, segmentation, and denoising tasks can be combined into a single model. Simultaneously modeling related

outputs with a shared set of parameters is referred to as multi-task learning [34]. A simplified formulation of this model for an input image x is given as

$$\hat{z} = f(x, w_e)$$
$$\hat{x} = g(z, w_d)$$
$$\hat{c} = m(z, w_c)$$
$$\hat{s} = p(z, w_s)$$

where $(f(\cdot), \hat{z})$, $(g(\cdot), \hat{x})$, $(m(\cdot), \hat{c})$, and $(p(\cdot), \hat{s})$ refer to the encoding, decoding, classification, and segmentation models and outputs, respectively. Note that all tasks share the parameters w_e within the encoder component yet have additional task specific parameters (w_d, w_c, w_s). The combined loss to be minimized is thus given by

$$L_{MTL}(\hat{x}, \tilde{x}, \hat{s}, s, \hat{c}, c) = L_d(\hat{x}, \tilde{x}) + L_s(\hat{s}, s) + L_c(\hat{c}, c)$$

which is the sum of the losses of each task. As described in Section 2.3.2, $\tilde{x}$ refers to a smoothed version of the original input image x. In practice, this approach requires careful formulation of each loss function to account for differences in class imbalances as detailed in Section 2.3.1.

While a conceptual visualization of the model framework can be found in Figure 2.5, specific details of the model architecture are decribed here. The cornerstone of this model is a fully convolutional encoder-decoder architecture that gradually down-samples the spatial dimension of an image and then up-samples this lower dimensional representation to reconstruct the original input. The first layer of the model consists of 16 convolutional filters, which are doubled at each successive layer until reaching a maximum of 128 filters. By using an image as both an input and an output, the encoder-decoder model learns a downsampled representation, referred to as the encoding layer, from the data without the need for class labels. The encoding layer can then be used as an input into classification and segmentation models. Specifically,

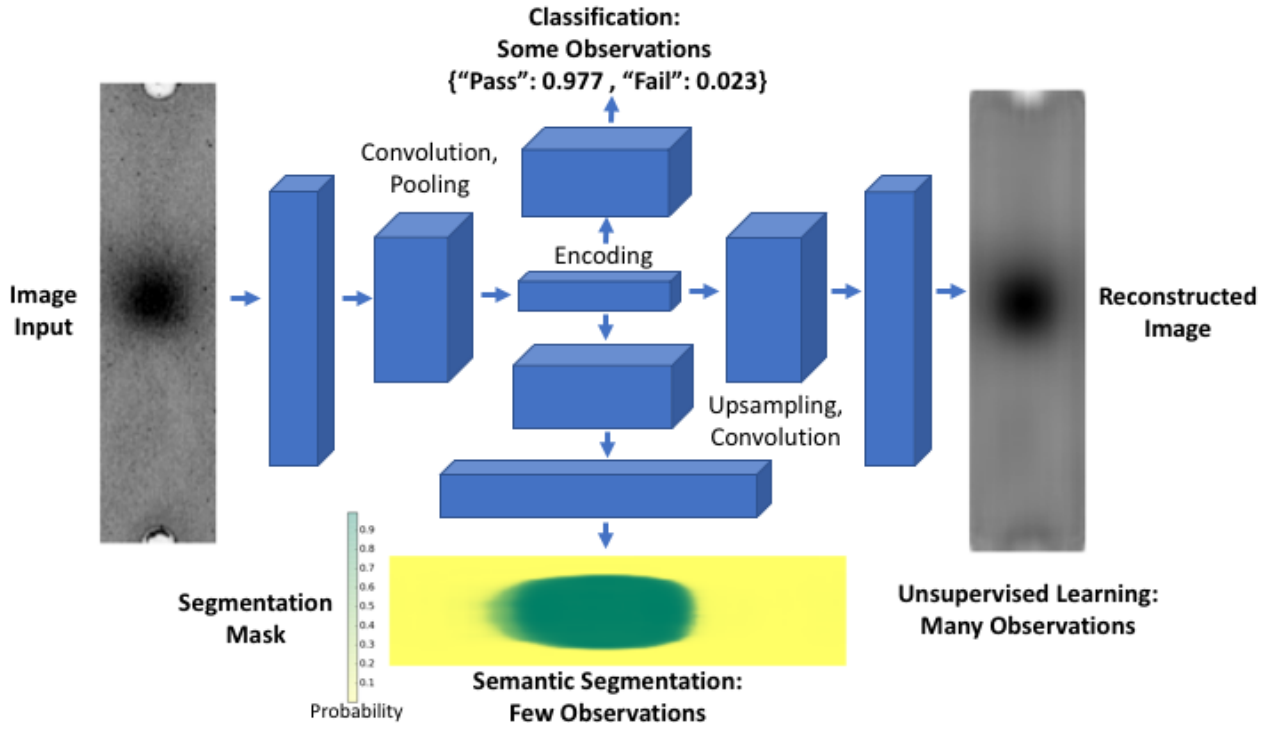

Figure 2.5.: Architecture of Classification, Segmentation and Denoising Model. An encoder-decoder architecture forms the basis for this model and enables learning from large amounts of unlabeled data. The encoding learned from unlabeled data can then be used for classification (top), as well as segmentation (bottom). Because of the cost associated with obtaining labels, there typically are fewer labeled observations available for the classification task. Simultaneous training for all outputs is possible by weighting irrelevant gradient updates to zero within the loss function. Alternatively, the model can learn to denoise images by predicting a "smoothed" image. In this application, both segmentation and denoising labels must be generated using preexisting algorithms.

the 256 by 64 pixel (zero-padded) scEPC images are thrice down-sampled via by means of successive convolutional and max pooling layers resulting in a 32 by 8 dimensional encoding layer. The classification model uses the encoding layer with additional convolutional and downsampling layers to predict a binary class labels of either "pass" or "fail." The segmentation model uses the encoding layer with additional convolutional and upsampling layers to predict pixel-wise class labels of either "protein" or "background."

2.4.2 Quantification of Model Outputs

The multi-task model framework provides an efficient means to classify, segment, and denoise scEPC images, but the end use case requires accurate quantification of protein expression therein. Though attempts were made to predict the positive, real-valued protein expression quantities as an additional output, these models failed to generalize to held out data. For this reason, the output of the classification, segmentation, and denoising components were initially combined as a post-processing step as follows. The classification output was used to select images with protein expression absent of technical noise. For each selected image, segmentation probabilities are used to determine which regions contain protein expression profiles. For each row in the segmented region, the average value of the 5 left- and right-most pixels is subtracted from the sum of pixel values. Background subtraction is performed at each row in order to account for differences in brightness across the image. Due to the typical size of a protein target, these pixels can reliably be considered as "background." The final protein expression value is then computed by summing the background corrected expression of all rows.

Given the success of the original quantification algorithm with the outputs of the multi-task learning deep learning model, it was posited combining the outputs within a deep learning model would be a direction to explore. This additional step consists of formulating the protein expression output as a function of the classification, segmentation, and denoising outputs. The model is then optimized with respect to all tasks including mean-squared error loss of the protein expression values. Performing the quantification in such a way has two potential advantages. First, a model can be optimized and evaluated in terms the end use case, potentially improving with the collection of more training data or hyperparameter tuning. If certain artifacts within the classification, segmentation or denoising labels are not useful for the prediction of protein expression, the model can be regularized by increasing the relative weight on the protein expression loss. This is especially important when only approximate labels

are available as in the case of segmentation and denoising. Second, quantification can be performed using the GPU hardware without any post-processing. Using a single model can not only speed up computation, but also reduces the complexity of the pipeline by eliminating additional steps. The performance of this single model quantification is evaluated in Section 2.5.2.

2.5 Training and Evaluation of the scEPC Quantification

The scEPC images used for model training were sourced from Prof. Amy Herr's Lab in the Bioengineering Department at the University of California, Berkeley. Two datasets are used to evaluate different aspects of model performance. The first dataset, referred to as the "experimental" dataset, contains scEPC images from thirty eight different experiment probing for six different proteins (actinin, btuB, GAPDH, GFP, tGFP, PS6) within three cell lines that were generated by two different researchers spanning over three years. The experimental dataset is split into training, validation, and test datasets. The experimental training and validation datasets are used for parameter estimation and hyperparameter tuning, respectively. The experimental test set is used for model evaluation of classification, segmentation, and denoising in the multi-task setting reported in Table 2.1. Comparison of the protein quantification between the Manual Guassian Fitting and Deep Learning Quantification Pipeline for the experimental test set is shown in Figure 2.6. The "ground truth" dataset consists of two channel scEPC images which simultaneously probe a protein (GFP) and an antibody (AB), a protein used in immunofluorescence assays, at various concentrations. Because the antibody is expected to bind to the GFP protein, the protein expression profiles of these data should to be correlated. Model evaluation of classification, segmentation, and denoising in the multi-task setting is reported for the ground truth dataset in Table 2.2. Additionally, the ground truth dataset is used to assess quantification of protein expression in the context of testing correlation in Section 2.5.2.

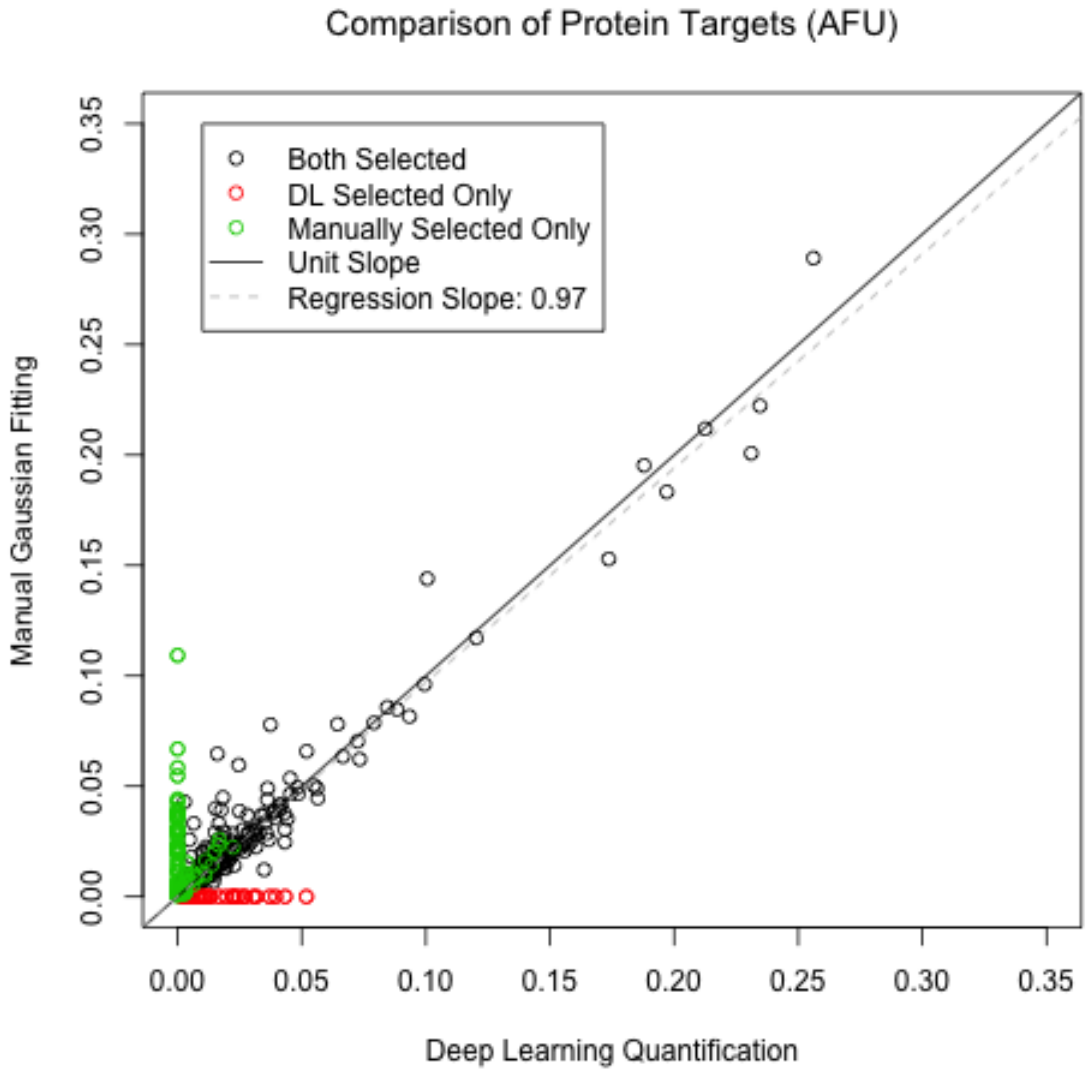

Figure 2.6.: Comparison of Predicted Quantification with Existing Pipeline. Scatterplot comparing the quantification of protein expression using manual gaussian fitting and deep learning on the experimental test dataset. Points in black represent images that were classified as "pass" by both the Manual Gaussian Fitting and Deep Learning Quantification approach for which there is high correlation (r = 0.97). Points in red are images that were classified as "pass" only by the Deep Learning Quantification model and thus have zero values for the protein expression value as they were not originally quantified. Points in green were classifed as "fail" by the Deep Learning Quantification model but not by Manual Gaussian Fitting. The protein expression values of these green points are computed by Deep Learning Model though in practice they would be excluded due to their classification.

Images from different experiments can have drastically different average pixel intensity values, class balance, and size. Images are also imbalanced with respect to these experimental covariates in that there are a different number of experiments across proteins and cell lines. Furthermore, each experiment contains varying proportions of "pass" and "fail" observations as well as the total number of observations. Because the objective the work presented here is to develop a quantification pipeline that generalizes beyond the experimental conditions of the training data (i.e., feasible for application to different experiments, proteins, and cell-lines), these covariates cannot simply be added into the model as inputs.

The models presented here were trained with Keras [35] using Tensorflow [36] for backend computation. The training procedure involves many standard techniques, such as data augmentation and learning rate decay. More recent techniques such as Batch Normalization [37], Exponential Linear Units (ELUs) [5], and the Adam optimization method [31] are also used.

2.5.1 Comparison of Single- and Multi-task Learning

To evaluate the potential benefit of multi-task learning, a systematic evaluation of all potential combinations is performed. While every attempt to control differences between the various training scenarios, it is difficult to make definitive claims about all datasets and model architectures. This is because the optimal value for certain hyperparameters, such as the learning rate, may differ for the individual tasks of classification, segmentation, or denoising. Additionally, there are challenges in interpreting the results of multi-task learning framework due to the inherent technical variation and the fact that the labels for each task can only be considered an approximation of the true labels.

Even with the potential issues raised above, the results in Tables 2.1 and 2.2 provide useful insights into the quantification framework proposed in later Sections. In particular, the classification task tends to improve over the baseline when the model

is also trained on the segmentation task or when the model is trained on all three tasks. In contrast, training a model on the classification and denoising task shows worse performance compared to the model trained on each of these tasks individually. The best performance is obtained on the segmentation task when no other tasks are included. Given that there may be some inaccuracies in the label generation for each task, the perhaps inconclusiveness of these results emphasize the difficulties of evaluating a model in terms of intermediate metrics rather than the end use case.

Table 2.1.: Single- and Multi-task learning performance on the experimental test dataset for classification, segmentation and denoising. The table shows the average and standard error of the loss across five models. The best performance on the classification task is obtained when all three tasks are trained simultaneously. On the segmentation task, the best performance is obtained when only trained on the segmentation task only and other tasks seem to worsen performance drastically. While the performance on the denoising task is best when only on this single task, the three task model performs comparably.

Task	Classification		Segmentation		Denoising	
Metric	Loss	Std Err.	Loss	Std Err.	Loss	Std Err.
Classification	0.1993	0.0058	-	-	-	-
Segmentation	-	-	0.1179	0.0096	-	-
Denoising	-	-	-	-	0.0807	0.0003
Class. & Seg.	0.1877	0.0089	0.1725	0.0025	-	-
Class. & Denois.	0.2394	0.0165	-	-	0.0854	0.0036
Seg. & Denois.	-	-	0.1817	0.0001	0.0844	0.0001
Class., Seg., Denois.	0.1814	0.0021	0.1614	0.0038	0.0829	0.0013

Table 2.2.: Single- and Multi-task learning performance the ground truth dataset for classification, segmentation and denoising. The table shows the average and standard error of the loss across five models. The relative model performance on the ground truth dataset is similar to that of the experimental test dataset, though the models perform better overall. Models training on both the segmentation and denoising task only have the best performance. The best classification performance is obtained when training on both the segmentation and classification task but is also comparable to the models trained on all three tasks.

Task	Classification		Segmentation		Denoising	
Metric	Loss	Std Err.	Loss	Std Err.	Loss	Std Err.
Classification	0.142	0.0048	-	-	-	-
Segmentation	-	-	0.0431	0.0064	-	-
Denoising	-	-	-	-	0.0209	0.0003
Class. & Seg.	0.1232	0.0029	0.0822	0.0013	-	-
Class. & Denois.	0.1991	0.0075	-	-	0.0226	0.0006
Seg. & Denois.	-	-	0.0819	0.0001	0.0224	0.0001
Class., Seg., Denois.	0.1289	0.0044	0.0728	0.0041	0.0216	0.0003

2.5.2 Testing Correlation of Ground Truth Data

While determining the classification and segmentation accuracy of the deep learning pipeline is important, evaluating the utility of this approach should be measured in terms of the end use case. This is because, given a set of labels from manual gaussian fitting approach, it is difficult to determine whether any differences in prediction and quantification are the result of human or algorithmic error. Evaluation of model performance on the end use case is particularly important in the context of denoising, as this feature does not exist in any form within the original approach. To compare each approach, it is thus necessary to compare performance in estimating correlation of a experiment when the two proteins being quantified are known to be correlated. In such a "ground truth" setting, each algorithm can be compared in terms of the number of "pass" protein targets, the correlation estimate, as well as the final p-value computed which tests whether the correlation is in fact zero.

In order to compare manual curation to the proposed quantification pipeline, the pearson correlation coefficient was tested for statistical significance on various experiments. The correlation r is computed as

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(x_i - \bar{x})^2}\sqrt{(x_i - \bar{x})^2}}.$$

Under the null hypothesis of zero correlation for bivariate normal data, the test statistic is t-distributed and is computed as

$$t = r \times \frac{\sqrt{n-2}}{\sqrt{1-r^2}}.$$

It is worth noting that the absolute value of the test statistic t is an increasing function of both the correlation of the data r and the number of observations n. That the p-value tends to be more significant with more observations makes the comparison between the two approaches less straightforward.

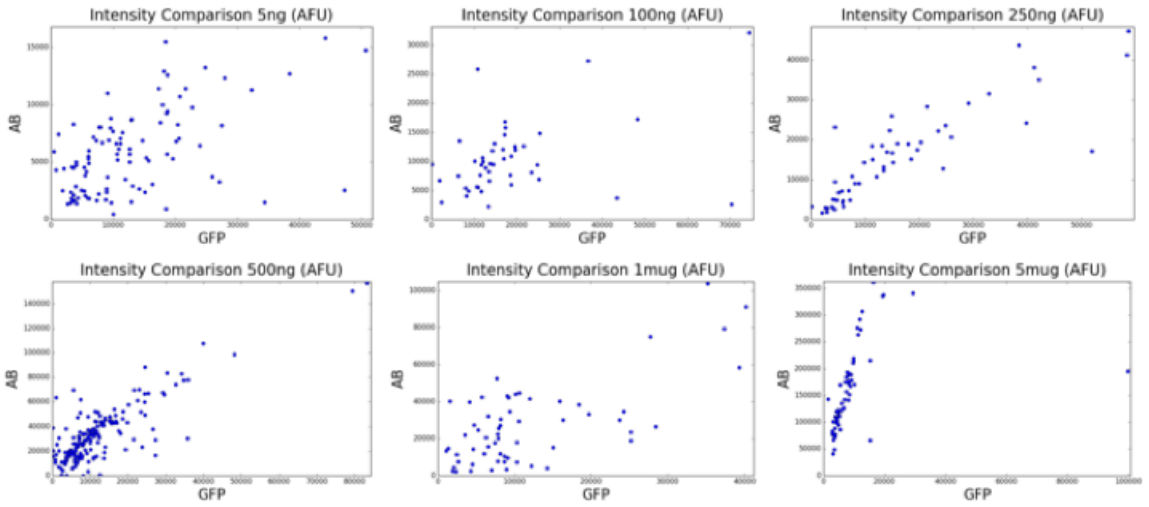

Figure 2.7.: Comparison of Predicted Quantification with Existing Pipeline on Specific Experiments. Scatter plots of GFP protein and antibody (AB) quantification from the Deep Learning Quantification model. Overall, the protein expression values are highly correlated as expected. In the lower right plot GFP and AB at 5mug concentration there is a clear outlier where the model has failed to provide an accurate prediction.

	Manual Gaussian Fitting			DL, Denoised, .5 Cutoff			DL, Original, .5 Cutoff			DL, Denoised, .95 Cutoff			DL , Original, .95 Cutoff		
Name	# Peaks	Corr.	P-Value	# Peaks	Corr.	P-Value	# Peaks	Corr.	P-Value	# Peaks	Corr.	P-Value	# Peaks	Corr.	P-Value
100ng	15	0.71	3E-03	45	0.41	6E-03	56	0.35	8E-03	35	0.80	6E-09	39	0.57	1E-04
1mug	16	0.62	1E-02	60	0.70	7E-10	66	0.34	5E-03	34	0.67	1E-05	37	0.53	8E-04
250ng	19	0.79	7E-05	54	0.87	2E-17	62	0.64	2E-08	42	0.85	6E-13	47	0.55	6E-05
500ng	50	0.89	8E-18	177	0.83	1E-46	196	0.63	5E-23	121	0.86	5E-37	134	0.65	1E-17
5mug	13	0.95	4E-07	62	0.37	3E-03	65	0.75	4E-13	35	0.95	4E-18	41	0.56	1E-04
5ng	14	0.75	2E-03	100	0.54	9E-09	113	0.51	8E-09	56	0.49	1E-04	65	0.25	4E-02

Table 2.3.: Correlation and P-values for Ground Truth Dataset. The leftmost, orange column includes the original number of "pass" images for both the GFP and AB images, the estimated correlation, and the p-value of the test of correlation in the ground truth dataset. The green column includes the same results of the Deep Learning Quantification model, which have similar correlation estimates but more significant p-values. The added significance is a result of the Deep Learning Quantification model including more observations and thus providing more confidence in the significance of the correlation.

The protein expression estimates of the Deep Learning Quantification pipeline for the ground truth dataset are shown in Figure 2.7. These plots indicate that the

estimates of the AB and GFP expression are indeed correlated. The number of positive class observations ("# Peaks"), correlation estimate, and p-values for the test of correlation for both the Manual Gaussian Fitting and Deep Learning Quantification approach for various hyperparameter settings are shown in Table 2.3. For the deep learning approach, two thresholds (0.5 and 0.95) for the predicted probability of the positive class were used to determine whether an image would be used for quantification. Additionally, quantification was performed on either the denoised or original image. These results suggest that quantifying the denoised images using a higher threshold for the positive class gives higher estimated correlation compared to other deep learning model settings. The higher number of observations classified as the positive class as well as the comparable correlation estimates indicate that the Deep Learning Quantification pipeline can replicate the existing approach without the involvement of manual curation.

2.6 Discussion

Computational models, such as convolutional neural networks, have the potential to play an important role in quantification pipelines of biological measurement technologies as presented here. To be used in practice, however, care must be taken to ensure the robustness of these models to technical variation in order to generalize to data from new experiments. Technical variation can, in part, be accounted for by using experimental covariates within the optimization procedure. The work presented in this section thus offers useful techniques to improve the quantification of scEPC and other images used for the quantification of molecular phenotypes.

3. CLASSIFYING GENOMIC MUTATIONS IN CANCER DIAGNOSTICS

Statistical analysis of genomic variants and their relation to disease or other outcomes has played a fundamental role in understanding the genomic basis of biology [38]. The role of the genome in biological function is complex and thus cannot be fully understood solely through genome-wide association studies. Specific genomic sequences often delineate regions of genes with specific functions (enhancer, promoter, etc.) and areas of open chromatin (transcription factor binding sites, histone modifications) [39]. Thus, the role of a specific base-pair (and the mutations thereof) is often determined by surrounding genomic regions. More typically, somatic mutations in cancer alter the exome, which are the regions of genes that are transcribed into RNA and are subsequently translated into proteins. Identifying and classifying how mutations affect biological functions often requires both detailed understanding of a particular gene as well as direct experimentation. Fortunately, public resources containing expert-curated data on somatic mutations in cancer, such as the Catalogue of Somatic Mutations in Cancer (COSMIC) [40], both identify, validate, and provide annotations for all confirmed as well as potential somatic mutations. However, downstream effects of cancerous mutations may be difficult to understand due to other complex mechanisms inside a cell such as protein interactions, microRNA-mediated gene regulatory networks [41], and chromosome folding [42]. Here a description of somatic mutation datasets is provided along with the formulation of various approaches to modeling genomic sequences using convolutional neural networks. Two applications, including validating hypothesized somatic mutations and diagnosing cancer from raw sequencing data, are also discussed.

3.1 Somatic Mutations and Cancer

Cancer is the result of complex changes in cellular function resulting from somatic mutations and their downstream effects on transcription and protein translation [43]. The disruption of specific processes within the cell, most notably cell division (mitosis) and programmed cell death (apoptosis) [44], lead to uncontrolled growth and, depending on the part of the body affected, may result in the formation of tumors. Other mechanisms affect the metabolism of cells leading to increased demand of vital resources which deprive normal cells of nutrients [45].

While mutations of single base pairs (bp) being changed from one nucleotide to another are the prevalent in cancer [40], other alterations such as deletions and insertions also occur quite frequently. Substitutions, deletions and insertions are further subcategorized according to how these mutations affect the transcription of a gene. The identification and classification mutations into specific subcategories are thus the result of broader research efforts of the human genome as well as of cancer specific sequencing experiments.

Indeed, owing to the availability of next-generation sequencing technologies as well as mathematical modeling techniques [43], the common mutational signatures of tissue-specific cancers have been discovered [46, 47]. The growth and development of cancer in the body, known as tumorgenesis, is driven by a progression of mutations that result in increasingly more aggressive cell proliferation, DNA damage, and DNA repair [48]. These mechanisms and signatures are often specific to the tissue involved [49].

3.1.1 Datasets

The Catalogue of Somatic Mutations in Cancer (COSMIC) [40] is a public resource providing detailed annotation of expert-curated somatic mutations from various sources such as research articles, cancer cell lines, and other databases including The Cancer Genome Atlas (TCGA) [50]. These mutations are stored in a text-

based format using standard nomenclature [51]. These data are available for targeted screens, whole genome screens, and non-coding variants. Each mutation data point also includes useful annotation such as the tissue of origin for the biological sample, whether the mutation has been confirmed using other data sources, as well as the genomic location, chromosome and strand.

Table 3.1.: Table of Most Mutated Genes in the COSMIC Dataset [40]. The dataset used includes many well-known oncogenes with varying frequencies of substitutions, insertions, and deletions.

	Mutation Type				Gene Info	
Gene	Sub.	Ins.	Del.	Total	Strand	CHR
APC	1047	273	722	2042	+	5
ARID1A	619	130	305	1054	+	1
FBXW7	920	80	112	1112	-	4
KMT2D	856	57	178	1091	-	12
MLL2	854	58	169	1081	-	12
NF1	787	51	222	1060	+	17
NOTCH1	887	63	107	1057	-	9
PTPRD	1692	12	38	1742	-	9
TP53	1568	272	872	2712	-	17
TTN	1006	4	10	1020	-	2
VHL	411	144	461	1016	+	3

3.2 Convolutional Neural Networks and Genomics

CNNs were inspired by the human visual cortex and perform well in image classification, as well as other tasks [1]. One key feature of CNNs is their ability to detect important characteristics at any location of an spatially ordered data; this is commonly referred to as the spatial invariance property. For this and other reasons, convolutional neural networks are used to predict molecular phenotypes of genomic sequence data in applications to better understand DNA- and RNA-protein binding [52], as well as non-coding variants [53, 54].

3.2.1 Modeling Genomic Sequences with Convolutions

While genomic windows are sequences of letters $b_{ij} \in \{A, T, C, G\}$ for window observation $i = 1, \ldots, N$ and window location $j = 1, \ldots, W$, it is necessary to transform these data into a quantifiable format that can be used by model. One-hot encoding [55] (i.e., transforming categorical variables into a vector of zeros but for single entry of 1) accomplishes this task by transforming genomic letters into a vector as follows:

$$x_{ij} = \begin{cases} (1,0,0,0), & \text{if } b_{ij} = A \\ (0,1,0,0), & \text{if } b_{ij} = T \\ (0,0,1,0), & \text{if } b_{ij} = C \\ (0,0,0,1), & \text{if } b_{ij} = G. \end{cases}$$

The resulting one-hot encoded genomic sequence is thus a $4 \times W$ matrix. Collecting N genomic windows for the purposes of training a model results in a $N \times W \times 4$ matrix of data. In this form, it is possible to model this data with neural networks using convolutional filters. Given a convolution of size $K \times 4$, the output of each hidden unit within a model for a single filter at each location $w = 1, \ldots, W$ is described as:

$$\begin{aligned} \begin{bmatrix} z_u^1 & \ldots & z_u^{1,W-K+1} \end{bmatrix} &= \begin{bmatrix} x^1 & \ldots & x^W \end{bmatrix} * \begin{bmatrix} w_1 & \ldots & w_K \end{bmatrix} \\ &= \begin{bmatrix} (w_1 x^1 + \ldots + w_K x^K) & \ldots & (w_1 x^{W-K+1} + \ldots + w_K x^W) \end{bmatrix} \end{aligned}$$

where $w_k x^w$ represents the product of the vectors $w_k = (w_{k,A}, w_{k,T}, w_{k,C}, w_{k,G})$ and the one-hot vector x^w. In CNNs with many layers, the z_u^w are then non-linearly transformed and subsequently convolved until the final output layer which predicts the class or other quantitative feature of the genomic window being considered.

3.2.2 Interpreting Mutations via Change in Predicted Probability

In many applications, deep learning models are used to predict the class of a genomic window that is obtained either experimentally or via annotation from an outside database using only the reference genome. In the simple case of a "positive" and "negative" class $c \in \{0, 1\}$, a model is trained to minimize the binary cross-entropy loss function

$$L(\hat{c}, c) = \sum_{i=1}^{n} c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i)$$

where the output $\hat{c}_i = d(x_i, w)$ is that of a convolutional neural network. The class assignment of genomic windows is typically determined by outside annotation. In the case of DNA- and RNA-binding proteins [52], positive class genomic windows were determined from outside biological experiments and negative class genomic windows were simulated as completely random genomic sequences. In the case of chromatin accessibility [53, 54], both positive and negative genomic sequences were obtained from outside biological experiments.

One way to interpret a mutation (i.e., a change a base pair b to b^m) is to compute the difference in the predicted probability of a class [52, 53]. For example, the score of a mutation s_m in a genomic window is computed as

$$s_m = \hat{c}^0 - \hat{c}^m = d(x_{ij}, w) - d(x_{ij}^{m_j}, w)$$

where x_{ij} and $x_{ij}^{m_j}$ denote the original genomic window and the genomic window with a base pair mutation m_j at location $j \in \{1, \ldots, W\}$ in window $i \in \{1, \ldots, N\}$. Computing the score of a variant in this way is particularly useful as it is only necessary to change in the input sequence, refered to as an *in silico* mutation [54].

While some work has investigated the interpretation of CNNs in the context of genomics using the above method known as DeepLift [56], it is unclear whether using the difference of the predicted outcome resulting from a single base-pair change of the input is a biologically-valid interpretation of a mutation. For example, changing a single nucleotide of a genomic sequence input could alter the prediction of an outcome due to fact that the model has never seen such data, the 'understood' genomic location has changed, or such a variant is actually casual. Indeed, there is a great deal of research on how and why small perturbations of an input can lead to drastic changes in the predicted outcome of CNNs in the context of images [57]. These issues seem to limit the potential of CNNs to precisely identify which *in silico* mutations are casually linked to a genomic outcome of interest using only reference genomic sequences.

The flexibility of scoring any conceivable variant makes the change in predicted probability approach a valuable tool for prioritizing genomic regions for further investigation. Because such methods are used for exploratory purposes, the false positive rate for mutation scoring at every location in the genome is unknown [58]. The predictive performance of this approach has often been assessed using variants with known effects on a biological outcome of interest. In this setting, an accurate prediction is defined as the change in predicted probability matching the signed effect of a variant. For example, the modeling performance of methods employing the change in predicted probability of *in silico* variants is reported for applications including DNA- and RNA-binding protein binding (up to AUC = 0.76) [52] and chromatin accessibility (up to AUC = .72) [53]. While the the performance metric used (area under the receiver operating characteristic curve) is not well suited for highly class imbalanced data, these metrics suggest that such an approach would not be accurate enough when classifying relatively rare somatic mutations in DNA sequencing data.

3.2.3 Classifying Mutations Directly

There are key differences between classifying somatic mutations and other applications using convolutional neural networks on genomic sequences that inform and enable a different modeling approach. Perhaps the main differentiator is that, for the case of transcription factor binding sites, the reference sequence of a genomic region can be used with a known annotation or label set. The regulatory mechanisms involved in specific transcription factor binding proteins are dependent on the occurrence of specific genomic sequences (known as motifs) that occur throughout the genome; the primary purpose of modeling annotated regions is to learn the genomic sequences in common between many genomic regions and use simulated mutations to predict potential changes in the probability of a particular class. In constrast, cancer is perhaps the result of differences in genomic sequences when compared to the reference genome. It maybe putatively characterized by certain mutational signatures that cause the activation of specific oncogenes and inactivation of tumor suppressor genes [59, 60].

Existing model approaches using the reference genome have potential uses in understanding the genomic basis of cancer. Indeed, cancer is the result of complex and diverse mechanisms that often involve transcription factor binding proteins, as well as other regulatory processes such as microRNAs that are often characterized by specific motifs [59, 61]. While a model that is trained on the reference genome with cancer-specific annotations could be used in cancer diagnosis, it is unclear whether the change in predicted probability approach would offer the precision and recall needed for highly imabalanced clinical applications.

The key challenge in an application such as cancer diagnostics is to accurately discriminate between mutations that are causally linked to tumorgenesis from mutations or sequencing errors that are non-causal. While using a model to predict an outcome that it was not trained to predict is an interesting proof of principle, the most compelling use case for deep learning is in the context of large, labeled datasets.

Because the COSMIC database consists of millions of curated cancerous mutations, it is therefore feasible to train a deep neural network to predict the class of a mutation directly with potentially high accuracy. The performance of this model can be assessed by classifying whether unseen mutations at given locations of the genome are cancerous or not. Using cancer diagnostics as a motivation, such a model has potential in a clinical setting to classify mutations observed in sequencing data for which no annotation exist. From a research perspective, the predicted probability of a mutation being cancerous can be used as a means of prioritizing unvalidated mutations for further investigation. These applications are discussed in Sections 3.3 and 3.4.

Because different cancer types are known to have particular mutational signatures such as changing "GAT" to "GGT" [40], only including the reference genomic window with a mutation as a model input is not ideal. It is therefore essential to design the model input in order to convey how a mutation changes the reference genome, as well as the broader genomic context. For these reasons, the input for the models evaluated in this Sections 3.3 and 3.4 consist of two $4 \times W$ matrices with one matrix being the reference genome and the other matrix being the mutated version of the reference genome. Formatting a single observation as two stacked genomic windows (i.e., one reference sequence and one mutated sequence) has also been used in related work [62]. A visual representation of the data is depicted in Figure 3.1.

3.3 Early Detection of Cancer from Raw Sequencing Reads

There were an estimated 9.6 million deaths in 2018 caused by cancer worldwide [63]. While there are many treatments available, or under development, to prolong life in patients diagnosed with cancer, early detection is potentially the best approach to reducing mortality [64]. The incidence of cancer is $1 - 2\%$ in the United States population and cancers are often tissue specific, therefore population-wide screening of tissue specific cancers is cost prohibitive and liable to result unintended patient

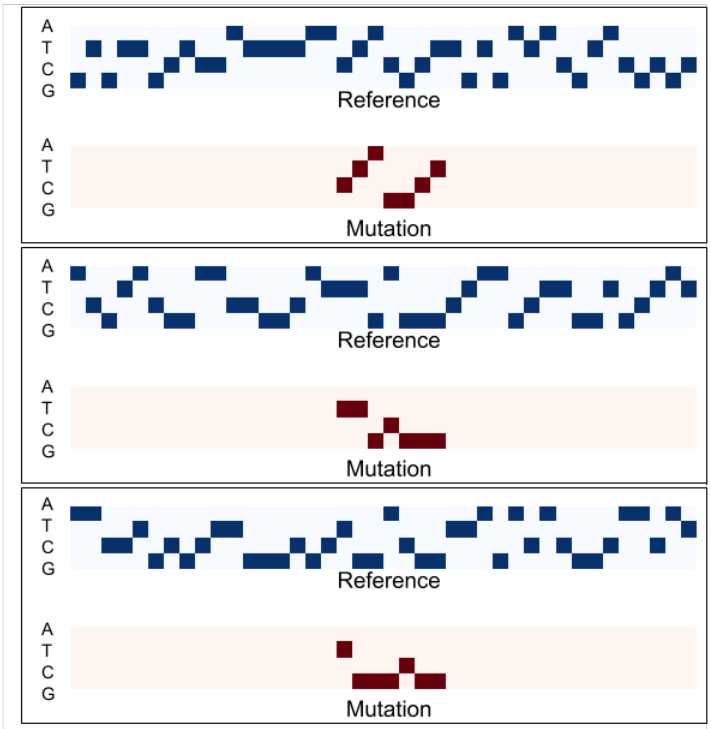

Figure 3.1.: The input data includes the whole reference genomic window as well as a subset of the this window with a cancerous or non-cancerous mutation. Depicted here are three examples of reference genomes with their corresponding mutations.

harm due to false positive diagnoses [65]. In order to obtain a broader view of human health, many researchers and clinicians have turned to an approach known as liquid biopsy, which refers to measuring biomarkers by collecting blood samples [66]. While blood contains many biomarkers of interest, the focus here is on approaches that sequence cell-free DNA (cfDNA) that is present in the blood that is an artifact from cell death (apoptosis). Although the use of liquid biopsies for the early detection of cancer has great potential, the approach faces three interrelated challenges: 1) the percentage of circulating tumor DNA (ctDNA) represents a small fraction of total cfDNA ($< 1\%$) [67]; 2) the current sequencing technology is cost prohibitive; and 3) algorithms used to interpret the genomic output of a liquid biopsy need to be sufficiently accurate [65].

3.3.1 Existing Approaches to Cancer Diagnosis using Liquid Biopsies

Modeling somatic mutations in DNA samples is not the only way to diagnose cancer using next-generation sequencing technologies. Copy number variation (CNV) investigates abnormalities of the number of reads detected in specific areas of the genome [67], though it can be difficult to model CNV for cancer diagnosis due to PCR bias [68]. Other approaches include measuring cell-free RNA levels [69]. Such approaches can not only identify mutations within well studied genes and pathways but also direct a clinical path forward using treatments that target specific cancer signatures [70].

3.3.2 Simulating Population Liquid Biopsy Results

Previous research has shown that CNN models perform well in predicting annotations of genomic sequences [52, 53]. It is essential to understand the performance for the end-use case, namely population-level screening of early-stage cancer. While obtaining real world clinical data is possible, it requires a significant amount of resources [65]. It is instructive, however, to understand the performance of CNNs using certain assumptions on population cancer incidence, the proportion of ctDNA to total cfDNA in the blood, the error rate of the sequencing instruments, as well as the relative frequency of substitutions, insertions, and deletions.

In order to understand the performance of a model in the context of cancer diagnostics, prediction is evaluated on a dataset designed to mimic the sequencing output of a liquid biopsy with a total of one million unique reads. With an assumed sequencing error rate of 0.5% using a 150 base pair read, the estimated frequency of reads being error free is 47.1% using the binomial distribution function. For simplicity, the remaining reads are assumed to contain one error per read, which has an actual estimated frequency of 35.5%. Of the original one million reads, it is assumed that 1% contain mutations resulting from cancer, with only one mutation (either a substitution, insertion or deletion) per read. With the above calculations in mind,

the relative frequency of reads with sequencing errors and cancerous mutations is assumed to be $52.3:1$. While there are ample positive class mutations in the COSMIC database, the relatively high imbalance of the two classes makes the task of modeling these mutations quite challenging.

3.3.3 Data Pre-processing the COSMIC Dataset

As mentioned previously, a single model is two $W \times 4$ genomic window where W is the number of total number of base pairs. The two genomic windows consist of one $W \times 4$ reference genome sequence, as well as one $W \times 4$ window containing the somatic mutation, either real or simulated, at the center (index $W/2$) with three base pairs of the reference genome on both sides of the mutated sequence. Three types of mutations are considered: substitutions, deletions and insertions. While there is a large range of sizes of mutations, only mutations of less than six base pairs in the COSMIC dataset (a large majority of all mutations) are considered.

Deep learning model performance is highly dependent upon the data used for evaluation, especially in the context of unbalanced biological applications. In order to ensure the model is not simply overfitting particular sequences within a gene within which cancerous mutations occur, the validation and test data sets are created only using mutations from chromosomes 2 and 3, respectively. The model is only trained on data outside of these two chromosomes, using chromosome 2 as a means of tuning the architecture and hyperparameters of the model, and completely holding out chromosome 3 for final evaluation. Two different COSMIC datasets are considered: "COSMIC Complete Mutation Data (Targeted Screens)" and "COSMIC Mutation Data" [71]. The first dataset (henceforth referred to as the "targeted" dataset) contains complete curated COSMIC dataset (targeted screens), whereas the second dataset (henceforth referred to as the "total" dataset) includes coding point mutations from targeted and genome wide screens (including whole exome sequenc-

ing). In order to understand the benefit of using additional training data, all models were evaluated on held out test data from the targeted dataset only.

Another important factor to consider for deep learning models is how the data are sampled for each batch of data used for training. For this reason, great care was taken to create a data sampling scheme that does not result in biased predictions that can result from unknown properties of the data. First, to better discriminate between two highly imbalanced COSMIC classes, equal numbers of both positive and negative examples (64 total observations) were included in each training batch. In order to ensure that a model does not overfit to reference genome sequences in regions with high numbers of cancerous mutations during training, the following procedure was employed: Positive class mutations, as well as the surrounding $3,000$ base pairs ("surrounding windows") are sampled uniformly from the COSMIC dataset. The middle subset (of lengths 200 bp, 500 bp, or 1000 bp) of these $3,000$ surrounding windows, referred to as "reference input window" is then used alongside the "mutation input window", which includes the mutated sequence, as well as three base pairs of the original reference sequence on each side (the remaining values within the mutation input window are set to zero). Because, the goal of this work is to classify whether there is a difference from the reference genome when considering a cancerous mutation versus a non-cancerous mutation or error from sequencing, it is necessary to simulate the negative class input data. To create negative class input data, a genomic window is sampled from within a corresponding positive class input window. The negative class reference input window is then accompanied by genome input window containing a simulated mutation at the mid-point. The $3,000$ bp surrounding window ensures that a large number of negative class reference input windows can be sampled during training regardless of the size of the input.

Additional efforts were made to ensure the negative class mutations did not differ systematically from the positive class. The type of negative class mutations were simulated with the same frequency of substitutions (96.47%), deletions(2.52%), and insertions (1.01%) as in the positive class. Furthermore, the frequency of the size

of the mutation within each type were consistent between both classes. The most common mutation size was a single base pair though the frequency of a point mutation differed between substitutions (99.52%), deletions (72.85%), and insertions (82.21%). These frequencies were computed from the total training dataset and used for the negative class simulation during both training and test set evaluation.

3.3.4 Results

Here the descriptions of the deep learning models and their predictive performance using three input window sizes, two training datasets, and two model architectures as described. In order to provide a straightforward comparison of results, hyperparameters were kept constant for all models unless otherwise specified. Input windows of size $W \in (200, 500, 1000)$ bp were considered to understand whether the cancerous mutations could be identified based on a larger genomic context or whether cancerous mutations relied more on local genomic information. The first dataset ("targeted") is a smaller set of curated mutations from targeted screens from the COSMIC database [71]. The second dataset ("total") is larger as it includes mutations from both the "targeted" dataset, as well as additional mutations from genome-wide screens. The test dataset consists of $8,953$ positive class cancerous mutations located in chromosome 3 from the targeted dataset as well as $465,556$ simulated negative class mutations (to maintain a $52:1$ ratio of the two classes) within the same genomic regions as the positive class mutations.

Because of the atypical shape of the data input windows $(W, 4, 2)$, where $W \in (200, 500, 1000)$, various modifications of traditional convolutional neural networks were employed for both training and comparison purposes. The first layer of all models consists of a 3-dimensional convolution layer with a filter size of $(3, 4, 1)$ ensuring convolutional filter weights are learned and applied to both the reference and mutation input windows. The resulting output of the first layer is of size $(W - 2, 1, 2, c)$, where c is the number of channels used in the first layer. A second 3-dimensional

Table 3.2.: Table of Ensemble Convolutional Neural Network Modeling Performance. This table displays the precision and recall using the average prediction of five convolutional models for each setting. Within each dataset, using a wider genomic region improves predictive performance. Within the 1000 bp setting, the use of the targeted dataset leads to higher precision whereas the total dataset leads to the highest recall.

Dataset	Size	True	Total	Precision	Recall
Targeted	1000	820	1249	0.6565	0.0916
Targeted	500	876	1443	0.6071	0.0978
Targeted	200	655	1112	0.589	0.0732
Total	1000	1217	2034	0.5983	0.1359
Total	500	1105	1905	0.5801	0.1234
Total	200	854	1413	0.6044	0.0954

Table 3.3.: Table of Ensemble Recurrent Neural Network Modeling Performance. This table displays the precision and recall using the average prediction of five recurrent models for each setting. Including more training examples with the total dataset as well as using wider genomic windows both improve model performance.

Dataset	Size	True	Total	Precision	Recall
Targeted	1000	961	1631	0.5892	0.1073
Targeted	500	845	1460	0.5788	0.0944
Targeted	200	377	614	0.614	0.0421
Total	1000	1042	1704	0.6115	0.1164
Total	500	1017	1736	0.5858	0.1136
Total	200	860	1465	0.587	0.0961

convolutional layer is then applied with a filter size of $(3, 1, 2)$ resulting in an output of $(W - 4, 1, 1, 2 * c)$, allowing the model to combine information from the reference and mutation input windows. Exponential Linear Units (ELU) activation functions are used in these two layers and followed by downsampling via a max pooling layer with pool size 2 and a dropout layer. Because there are $4^3 = 64$ combinations of three consecutive base pairs, the number of filters c in the first layer is set to 64.

Model architectures then follow the general format of repeatedly using a series of four layers: a 1-dimensional convolutional layer with filter size 3 with an additional 64 filters compared to the previous layer, an ELU activation layer, a max pooling

layer with a pool size of 2, and a dropout layer. Because max pooling divides the length of the output of each intermediate layer by two, different input sizes result in different model architectures. In order to avoid differences in performance relating to the number of layers and parameters, all convolutional models consist of eight layers regardless of the input size. If the intermediate output layers are max pooled to a length less than three, the filter size is then set to match the output length, mimicking the effect of a dense layer for filter size of 1. Recurrent architectures, consist of the first five layers similar to their convolutional counterparts (though with fewer filters for computational reasons) followed by a bi-directional LSTM layer. All models include a final dense layer prior to predicting the binary class output. The number of model parameters ranges from $\sim 2-4$ million for convolutional architectures and ~ 1 million parameters. Five models were trained for each input window size, dataset, and architecture. The results for both the average of these results as well as the ensemble prediction are shown in the Tables 3.2 and 3.3.

Performance metrics of the ensemble prediction for the two datasets using various input sizes for the convolutional and recurrent models are reported in Tables 3.2 and 3.3, respectively. As expected, using more training data does improve performance, particularly with respect to recall. Within each dataset, there is improvement in performance as larger input sizes are used. Finally, the convolutional model architecture tends to outperform that of the recurrent architecture. While the variability in training outcomes are perhaps, in part, due to the issues raised above, these results offer several useful insights.

3.4 Validating Somatic Mutations

Testing biological hypotheses can be difficult due to the inherent noise within sample preparation and measurement technologies. For these reasons, it is necessary to replicate experimental results, as well as validate any findings with other, and preferably independent, data sources. The COSMIC datasets considered here have

been thoroughly curated to provide high quality resources to the scientific research community. One practical application of the described models is to score unconfirmed somatic mutations with respect to a computational model trained on confirmed somatic mutations. This would not only enhance the utility by existing databases such as COSMIC, but also serve as a means of validating somatic mutations in other datasets, such as a sequencing experiments of an unvalidated cell line where data may be limited. As a proof of concept, the unconfirmed somatic mutations with the highest predicted probability of being cancerous are included in Table 3.4.

Table 3.4.: Table of Predicted Somatic Mutations in the Unconfirmed Cosmic Dataset [40]. This table includes the ten most highly predicted cancerous somatic mutations listed as unvalidated in the COSMIC database. This ranking can inform which mutations should be investigated to determine whether they are likely to be associated with cancer.

Gene	Type	Code	Strand	Chr.	Length	Tissue	Prob.
CD1E	Subst.	c.259G>T	+	1	1023	Lung	0.99913
OR10A3	Subst.	c.16C>T	-	11	945	Lung	0.99913
TRIM68	Subst.	c.1265G>T	-	11	1458	Lung	0.99913
TLR6	Subst.	c.2276C>A	-	4	2391	Endometrium	0.99909
FBXW10	Subst.	c.2645G>T	+	17	3156	Endometrium	0.99907
MMP17	Subst.	c.1388C>G	+	12	1812	Lung	0.99904
ZNF777	Subst.	c.282G>A	-	7	2496	Stomach	0.99904
CSMD1	Subst.	c.4691C>T	-	8	9882	Endometrium	0.99896
KCNQ3	Subst.	c.1666G>A	-	8	2619	Endometrium	0.99894
TRIM55	Subst.	c.1577G>T	+	8	1647	Endometrium	0.9989

3.5 Discussion

As demonstrated in this Chapter, computational models can be used in the context of cancer diagnostics. Classifying a read or base pair as cancerous is interesting, but ultimately must be evaluated in a clinical diagnostics setting. Datasets sets such as COSMIC provide an interesting opportunity to test whether the change of predicted probability approach (Section 3.2.2) correctly identifies a mutation of interest.

Because the data used to train and test any deep learning model are simulated, care must be used when drawing any conclusions about the utility of such algorithm. Indeed, real world sequencing data from patients must be closely approximated by making certain assumptions such as the relative frequency of cancerous and normal tissue reads as well as the inherent error rate within sequencing data itself. Furthermore, technology specific considerations, such as how genomic sequences are collected from patients are also important in determining whether such diagnostic algorithms are useful.

4. BEST ORDERED SUBSET SELECTION

4.1 Introduction

Combining evidence in a statistically robust yet powerful manner is a key challenge in research. When deciding which method to use, researchers often make assumptions about the underlying signal of their data. Meta-analysis, a general class of statistical procedures for combining evidence, assumes a consistently signed (i.e., positive or negative) effect across different experiments testing the same outcome. Additionally, many applications aim to pool evidence across different variables with the assumption that only a small subset are likely to exhibit signal. While some variants of meta-analysis account for the possibility of sparse signals ([72]), these methods are not designed to combine evidence of effects with different signs. For this reason, meta-analysis is not applicable to certain applications in genomics such as the association variant with differently signed effects on multiple traits.

This work builds on the area of research commonly referred to as adaptive testing, which originates from the Adaptive Rank Truncated Product (ARTP) method [73]. The ARTP tests the global null hypothesis by taking the most significant ordered subset using the product of uniform order statistics. The aforementioned work, itself an extension of the Rank Truncated Product (RTP) [74], developed a simulation-based approach to test the global null hypothesis of whether the best ordered subset is non-null. The key advantage of adaptive tests is the explicit assumption that potentially only a subset of the parameters are expected to show a non-null signal; this is often an implicit assumption in many applications. The computational cost of such procedures can be prohibitive in large scale applications using standard estimation techniques such as permutation testing and Monte Carlo integration, especially when estimating extremely small p-values [75].

Improving the understanding and implementation of adaptive testing is important and timely due to the increasing amount of scientific data that are available in the form of summary statistics, such as signed test statistics and p-values. While many existing methods for combining p-values exist, most do not explicitly account for the fact that only a small subset of the tests being considered will likely contribute a truly significant result. The demonstrated ability of adaptive tests to detect sparse signals relative to other methods is particularly useful in genomic applications [76]. A comparison of methods in this context including the Random Effects Meta-analysis [77], the LASSO [78], found that the ARTP method outperformed the others in terms of power [79].

In the field of genomics, summary statistics are often available from genome-wide association studies (GWAS). These data report the the significance of variants tested for association against diseases or other phenotypes [80]. Original approaches aimed to validate or combine evidence for a single variant across multiple studies of a particular disease. Combining evidence at the single variant level is problematic as separate studies encompass various populations with different linkage disequilibrium patterns (dependency between genomic variants) and disease incidence [81]. Because many scientific investigations are conducted in terms of large areas of the genome, such as genes or pathways, summary statistics for a hundreds to thousands of correlated variants can be pooled [82–84]. Summary statistics can also be used to investigate pleiotropy [85], i.e., when a single genomic variant is associated with multiple, often correlated, diseases or phenotypes [86]. These applications require a known or estimated correlation structure between test statistics [87], as the failure to model such depedence results in estremely small tail distributions and the inflation of type I error rates [88].

Due to the large number of tests involved, applying adaptive tests to genome-wide association studies (GWAS) presents additional statistical and computational challenges. The motivation for this work is to explore the statistical properties of adaptive testing, as well as develop computational strategies that allow these methods

to scale to large data applications. Three areas are considered. First, an adaptive testing framework is formulated in terms of the sum of exponential order statistics. This allows us to explore, theoretically as well as via simulation, the correlation between ordered subsets and the control of the type I error rate. Second, two variance reduction techniques are proposed in the context of adaptive testing for the purposes of improving upon standard Monte Carlo integration. Lastly, these methods are applied to three genomics datasets, illustrating their utility and interpretation.

4.2 Adaptive Testing Using Exponential Order Statistics

4.2.1 Computing the Combined P-Value of Each Subset

The theoretical and conceptual foundations of adaptive procedures are rooted in classical tests of the global null hypothesis using a set of p-values. One approach is to take the smallest of r p-values, denoted as $p_{(1)}$ to represent the 'best' individual attempt to find a non-null parameter. Under the null hypothesis, the distribution of the minimum p-value is determined by standard order statistics theory when the corresponding test statistics are independent. Alternatively, the negative log of the minimum p-value can be considered

$$M = -\log(p_{(1)}) = E_{(r)} \tag{4.1}$$

which is equivalently distributed as the maximal order statistic of r exponential random variables. Using the order statistic distribution, the minimum p-value (MinPV) can be used to compute the statistical significance of a set of p-values. The MinPV method is a useful means of summarizing multiple tests under the assumption that most of test statistics are generated from the distribution under the null hypothesis.

A second, and perhaps most well known, method aimed at combining the results of multiple tests is Fisher's method [89]. This method consists of taking the sum of

the negative log of each p-value multiplied by two. Mathematically, this results in the following expression for a total number of r tests with corresponding p-values p_i:

$$C = \sum_{i=1}^{r} -2\log(p_i). \tag{4.2}$$

Under the null hypothesis, each transformed p-value $-2\log(p_i)$ becomes chi-square distributed χ^2_2. Thus, the sum C is known to be distributed as χ^2_{2r}. Equivalently, we can consider the sum of the negative log of the p-values:

$$C^* = \sum_{i=1}^{r} -\log(p_i) = \sum_{i=1}^{r} E_i = \sum_{i=1}^{r} E_{(i)}$$

where a null-distributed p-value $-\log(p_i)$ becomes an exponential random variable $Exp(1)$. This indicates that C^* is distributed as a $Gamma(1, r)$. Note that because all of the r tests are considered, the sum of the r exponential order statistics is equal to the sum of all exponential random variables. Fisher's method is especially useful when multiple non-null effects are used together to reject the global null hypothesis.

After considering the maximum exponential order statistic, as well as the sum of all order statistics, a natural extension is the sum of the top k order statistics which can be written as

$$T_k = \sum_{i=1}^{k} -\log(p_{(i)}) = \sum_{j=r-k}^{r} E_{(j)}.$$

This method [74], was originally proposed in terms of the order statistics of uniform random variables. It is easy to see that the special cases of T_1 and T_r are defined as M (4.1) and C^* (4.2), respectively. In order to combine the k smallest p-values, it is therefore necessary to compute $p_{t_k} = P(T_k > t)$ for some scalar t. Unlike M and C^*, however, the distribution of the final test statistic T_k is less straight forward.

Existing work has thoroughly explored the distribution of the sum of exponential order statistics [90], some of which is included below for completeness. In the following formulation, consider $t = \hat{t}_k$ to be the observed test statistic for the smallest k observed

p-values. For a given k, the combined p-value $\hat{p}_{t_k}$ associated with an observed test statistic $\hat{t}_k$ can be computed as

$$\begin{aligned}\hat{p}_{t_k} &= P(T_k > \hat{t}_k) \\ &= \sum_{j=1}^{k} \frac{w_j \exp(\frac{-c_j \hat{t}_k}{c_{k+1}})}{(n-k-1)!} \int_0^{\hat{t}_k} \exp(d_j y) y^{n-k-1} dy + \sum_{l=0}^{n-k-1} \frac{\hat{t}_k^l}{l!} \exp(-\hat{t}_k)\end{aligned}$$

where $c_j = n - j + 1$, $d_j = \frac{c_j}{c_{k+1}} - 1$, and $w_j = \prod_{i=1,\neq j}^{k} \frac{n-i+1}{j-i}$. Unfortunately, this formulation is numerically unstable as it involves the summation of increasingly large values with alternating sign. Fortunately, the distribution of T_k can also be formulated as the sum of two random variables [90] with known distributions:

$$T_k =^d (n-k) \sum_{j=1}^{k+1} (\frac{1}{c_j}) E_j + W_k, 1 \leq k < n$$

where W_k is the sum of $n - k - 1$ standard exponential random variables and is therefore $Gamma(n-k+1, 1)$. With this later formulation, it is feasible to compute the combined p-value of the top k tests via an efficient Monte Carlo sampling scheme described Section 4.4.

4.2.2 Best Ordered Subset Selection

Assuming its distribution function can be evaluated, a natural choice would be to select the k that with the smallest $\hat{p}_{t_k}$. This insight underpins the original formulation and application of the ARTP test [76], after which the theoretical framework using uniform order statistics was further developed [91]. Henceforth, the minimum of combined, ordered p-value subsets is referred to as

$$p_{BOSS} = \min_k \hat{p}_{t_k} = \min_k P(T_k > \hat{t}_k)$$

the as the the best ordered subset selection (BOSS) test. This alternative formulation enables new directions of investigation in a way that the ARTP does not. In order to emphasize this distinction, BOSS indicates the prosposed modeling framework, and allows for the acknowledgement of existing foundational work.

Indeed, formulating the BOSS test in terms of the sum of exponential, rather than the product of uniform, order statistics has many benefits that are more thoroughly described in later sections. For example, exponential order statistic theory enables analytical expression of the covariance between the sums of ordered negative log p-values. The calculation of this covariance structure is then used as the basis of the deterministic approach used to control the type I error rate of this test. Similarly, the novel Monte Carlo integration algorithm proposed in Section 4.4.1 depends on additive combination of exponential order statistics.

4.3 Controlling Type I Error of BOSS

Selecting the best ordered subset of p-values without any adjustment will adversely affect the statistical properties of testing the global null hypothesis. Because the BOSS test is concerned with the global null hypothesis, interest lies in controlling the type I error rate of the minimum of all ordered combinations. A key question, however, is how should one adjust the end result so as to ensure sound statistical properties? Difficulties arise in determining this correction because the p-value of each ordered subset $\hat{p}_{t_k}$ is correlated with other overlapping subsets. The application of Bonferonni correction, for example, is not ideal as it is known to be overly conservative when applied to either highly dependent or large numbers of tests. Furthermore, this procedure is used to control the family-wise error rate (FWER), i.e., the probability of observing at least one Type I error when considering multiple tests, rather than Type I error rate of a single test. While permutation tests have long been used to provide appropriate significance cutoffs for correlated tests in genetics [92], their computational cost can be prohibitive for large genomic datasets [93]. In what follows,

deterministic approaches to controlling type I error rate for test statistics with known correlation are investigated in the context of BOSS.

4.3.1 The Covariance of Exponential Order Statistic Sums

The additive combination of exponential order statistics allows for straightforward calculation of their correlation structure. For $l < k$, the covariance matrix when there are r total variables under the global null hypothesis is formulated as

$$\begin{aligned}
Var(T_k) &= Var\left(\sum_{j=r-k}^{r} E_{(j)}\right) = \sum_{j=r-k}^{r} Var(E_{(j)}) + 2\sum_{i=r-k}^{r}\sum_{j=r-k,j<i}^{r} Cov(E_{(i)}, E_{(j)}) \\
Cov(T_k, T_l) &= Cov\left(\sum_{j=r-k}^{r} E_{(j)}, \sum_{h=r-l}^{r} E_{(h)}\right) \\
&= Cov\left(\sum_{j=r-k}^{l-1} E_{(j)} + \sum_{h=r-l}^{r} E_{(h)}, \sum_{h=r-l}^{r} E_{(h)}\right) \\
&= Cov\left(\sum_{j=r-k}^{l-1} E_{(j)}, \sum_{h=r-l}^{r} E_{(h)}\right) + Var\left(\sum_{h=r-l}^{r} E_{(h)}\right). \\
&= \sum_{j=r-k}^{l-1}\sum_{h=r-l}^{r} Cov\left(E_{(j)}, E_{(h)}\right) + Var(T_l)
\end{aligned}$$

where

$$\begin{aligned}
Var(E_{(k)}) &= \sum_{j=1}^{k} \frac{1}{n-k+1} \\
Cov(E_{(k)}, E_{(l)}) &= Var(E_{(l)}) = \sum_{j=1}^{k} \frac{1}{(n-k+1)^2}.
\end{aligned}$$

It is worth noting that the covariance between exponential order statistics, as well as their sum, only depends on the total number of tests when the tests are assumed to be independent.

4.3.2 Estimating the Effective Number of Tests

Efforts to understand and control the FWER within the large number of highly correlated tests in GWAS has given rise to the concept of the effective number of tests (ENT). Rather than using the total number of variants being tested to control FWER via Bonferonni correction, for example, researchers desire an estimate of the smaller number of effective tests in order to use less conservative significance cutoffs [94]. These methods consist of aggregating the eigenvalues computed from the correlation matrix of the associated test. Such approaches fit within the BOSS framework because the correlation of ordered subsets is easily computable. Intuitively, a procedure for controlling FWER of correlated tests should also control the Type I error rate of BOSS. Given that these procedures are rough heuristics originally developed for the correlation of normally distributed test statistics, further empirical evaluation is necessary.

For a given correlation matrix, there are many proposed methods for estimating the effective number of tests that must be evaluated. One approach [94] of calculating the effective number of tests, r_e, is given by

$$r_e = r - \sum_{j=1}^{r} I[\lambda_j > 1](\lambda_j - 1).$$

This and another approach [95] were evaluated in the context of BOSS. A third approach [96] was excluded from analysis as it was clearly incorrect (data not shown). Once computed, the effective number of tests can be used to adjust the significance threshold to $\alpha_e = 1 - (1 - \alpha)^{1/r_e}$ [97] or the approximate Bonferonni equivalent $\alpha_e = \alpha/r_e$. In Sections 4.5 and 4.6, BOSS p-values are scaled $p_{BOSS} \times r_e$ in order to facilitate comparisons of existing methods and calculate significance thresholds when computing multiple BOSS tests.

In the specific case of two tests, the type I error rate of the BOSS test without any correction is 0.0821, suggesting that the effective number of tests correction factor should be approximately $0.0821/0.05 = 1.642$ assuming $\alpha = 0.05$. This esti-

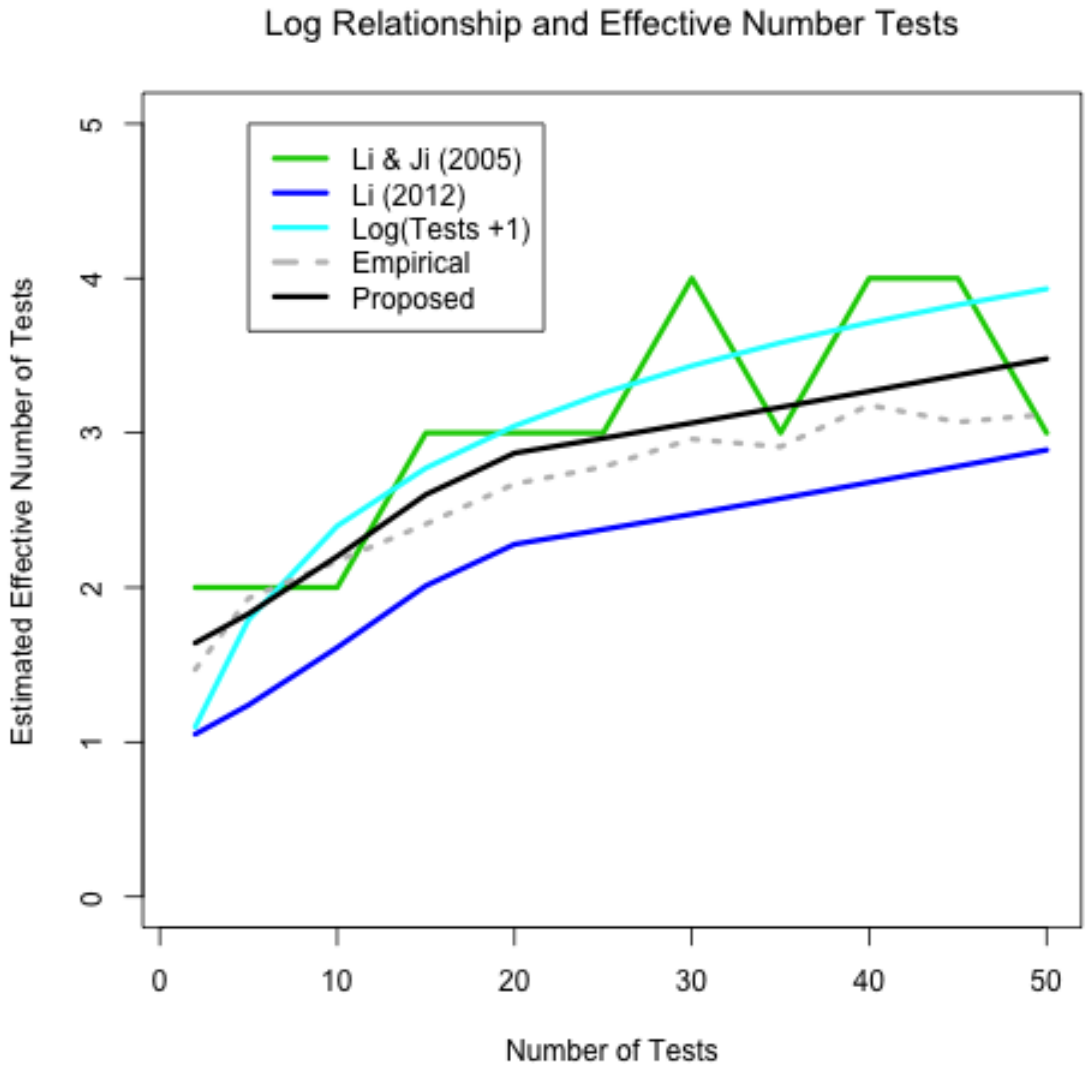

Figure 4.1.: Estimating the Effective Number of Tests. This plot shows various ways of computing the effective number of tests of BOSS using the covariance matrix of the exponential order statistics. Due to the fact that the combined p-values from each of the ordered subsets are highly correlated, the effective number of tests should be quite small relative to the total number of tests considered. The proposed method, which effectively controls the Type I error rate, is based on [94] with additional scaling determined by the two test case.

mate is different than all of the methods: 1.051 (Li (2012)) [94] and 2.000 (Li & Ji (2005)) [95] as shown in Figure 4.1. While the first two methods are too low to control type I error rate, the last method is clearly conservative as it is equivalent

to Bonferonni correction. Notice, however, that the Li's method of estimating the effective number of tests most closely follows the empirical distribution, except for the lack of the correct offset. For this reason, we propose using $BOSS_{ENT} = r_e + \delta$ where delta is the difference between the theoretically determined correction factor and the effective number of tests when only considering the two test case. Thus, for significance threshold $\alpha = 0.05$, we use $\delta = 0.591$ throughout the remainder of this work for the purposes of to calculating the effective number of tests correction.

4.4 Efficient and Accurate Computation of BOSS

4.4.1 Rao-Blackwellized Monte Carlo Integration

A numerical integration approach referred to as Rao-Blackwellized Monte Carlo (RBMC) [98–100] is presented for the purpose of computing the p-value p_{t_k} of each combined ordered subset. Together with the deterministic estimate of the effective number of tests correction in Section 4.3, efficient RBMC estimates of each p_{t_k} result in extremely efficient computation of BOSS p-values. Recall that we are interested computing the value

$$\hat{p}_{t_k} = P(T_k > \hat{t}_k) \tag{4.3}$$

where $\hat{t}_k = \sum_{i=1}^{k} -\log(\hat{p}_{(i)})$ is sum of the negative log of the smallest k observed p-values $\hat{p}_i,\ i \in \{1, \ldots, r\}$. Using an alternative formulation [90], the distribution of the sum of the top k exponential order statistics can be described as

$$T_k \sim (n-k)\sum_{j=1}^{k+1}(\frac{1}{c_j})E_j + W_k, 1 \leq k < n.$$

Substituting in the above formulation into the equation 4.3

$$\hat{p}_{t_k} = P((n-k)\sum_{j=1}^{k+1}(\frac{1}{c_j})E_j + W_k > t_k), 1 \leq k < n,$$

where W_k is the sum of $n-k-1$ standard exponential random variables and is therefore $Gamma(n-k+1,1)$. Although this value is challenging to determine analytically, standard Monte Carlo integration can be used to approximate its value numerically. A simple Monte Carlo approach would be to sample $s = 1 \ldots S$ values $W_{ik} = w_{sk}$ and $E_{sj} = e_{sj}, j = \{1, \ldots, k+1\}$ and compute

$$\hat{p}_{t_k}^{MC} = \frac{\sum_{s=1}^{S} I[\sum_{j=1}^{k+1}(\frac{1}{c_j})e_{sj} + w_{sk} > t_k]}{S}$$

where $I[.]$ is the indicator function taking a value of one when the inequality within the brackets holds and zero otherwise. Because it is often used to compute combined p-values taking extremely small values, the above method would require many samples to obtain even one value where the indicator function takes the value of one.

The RBMC approach consists of sampling all but one of the random variables describing T_k, and then use the distribution function of the remaining variable to compute the probability that this last variable is greater than the resulting number. Here, this simply means for a given sample s, $E_{sj} = e_{sj}$, for $j = \{1, \ldots, k+1\}$ and

$$P(\sum_{j=1}^{k+1}(\frac{1}{c_j})e_{sj} + W_{sk} > t_k) = P(W_{sk} > t_k - \sum_{j=1}^{k+1}(\frac{1}{c_j})e_{sj}) = P(W_{sk} > a_{sk}),$$

is computed where $a_s = t_k - \sum_{j=1}^{k+1}(\frac{1}{c_j})e_{sj}$. Because each $W_{sk} \sim Gamma(n-k+1,1)$, this probability can be computed exactly. We can thus approximate the combined p-value for the top k exponential order statistics as

$$\hat{p}_{t_k}^{RBMC} = \frac{\sum_{s=1}^{S} P(W_{sk} > a_{sk})}{S}.$$

Like other sampling approaches, $\hat{p}_{t_k}^{RBMC}$ is an estimate of the true value. It can, however, be computed with a high degree of accuracy by increasing the total number of samples S. The required number of samples ultimately depends the size of the true value and error tolerance. Empirical results in the two independent tests case $r = 2$ in Section 4.5.1 suggest that only $1,000$ samples, combined with the effective number

of tests correction, are needed to estimate p-values as extreme as $1e-8$. While the two independent tests case does not actually require sampling, the RBMC approach achieves a roughly $1,000,000$-fold reduction in the number of samples if compared to standard Monte Carlo integration (assuming $\sim 10^9$ are required) .

4.4.2 Importance-weighted Monte Carlo Sampling

Monte Carlo methods are widely used in statistical methodology within genomics. One challenge with standard Monte Carlo integration is that for the estimation of small values, the number of iterations required is extremely large. This issue typically arises when estimating the true p-value of a test when the p-value is very small. In order to obtain an estimate of a p-value with a true value of 1×10^{-8}, the standard Monte Carlo procedure would require require on the order of $\sim 10^9$ iterations to provide a somewhat low variance estimation. For these reasons, it is essential to employ more advanced sampling methods, such as the RBMC algorithm described in the previous section. While the RBMC provides an efficient algorithm for estimating extreme p-values, the theoretical basis of this method requires independence between tests. Due to the underlying correlation of tests in many genomics applications, additional methodology is required to efficiently apply BOSS in these scenarios.

Importance sampling [101] is a variance reduction method whereby biased samples are used at each iteration, and subsequently down-weighted based on the ratio of the biased sample density compared to the true sample density and evaluated at the value of the biased sample observation. This approach allows for rare values to be sampled more frequently, which, in turn allows for potential computational gains. For practical reasons, the importance sampling algorithm in the context of computing the minimum p-value of the best ordered subsets is described for the case where test statistics are normally distributed with known covariance. The importance sampling algorithm given below can easily be reformulated and used in other hypothesis testing methods originally developed for the independent testing case.

Under the null hypothesis, we assume that the test statistic is normally distributed with mean zero and known covariance. That is,

$$\beta \sim N((0, \ldots, 0), \Sigma)$$

where β is an r dimensional random variable and Σ is a $r \times r$ positive definite matrix. Given observed test statistics $\hat{\beta}_k, k = 1, \ldots, r$ and marginally computed (i.e., individually computed under the assumption of independent tests) p-values $p_k, k = 1, \ldots, r$, the p-value of the ordered subsets is desired. The first step is to estimate each $P(\tilde{T}_k > \hat{t}_k)$ where $\tilde{T}_k$ represents the distribution of the sum of the negative log p-values resulting from the ordered subsets under the null distribution of β with covariance Σ.

The standard Monte Carlo algorithm [101] for computing an estimate of the true p-value that accounts for the dependence between tests is

1. Determine $\hat{p}_k$ and the corresponding $\hat{t}_k$ for each ordered subset $k = 1, \ldots, r$ as described above.

2. For iterations $s = 1, \ldots, S$

   (a) Sample $\tilde{\beta}_s \sim f = N((0, \ldots, 0), \Sigma)$.

   (b) Compute the 2-sided p-values $\tilde{p_{sk}}$ of $\tilde{\beta_{sk}}$, as well as the corresponding $\tilde{t_{sk}}$ for each ordered subset.

   (c) Determine whether $\tilde{t_{sk}} < \hat{t}_k$, denoting $\tilde{I}_s = I[\tilde{t_{sk}} < \hat{t}_k]$.

3. Compute $\hat{p}_{t_k} = \frac{\sum_{s=1}^{S} \tilde{I}_s}{S}$

4. Determine $\hat{p}_{BOSS} = \min_k \hat{p}_{t_k}$.

At this point, each $\hat{p}_{t_k}$ will be uniformily distributed with respect to the null distribution. As in the RBMC algorithm, it is then necessary to correct $\hat{p}_{BOSS}$ to control the type I error rate due to the fact that each ordered subset is correlated. To obtain

the empirical distribution of the minimum of each ordered subset p-value $\min_k \tilde{p_{t_k}}$ the previously sampled $\tilde{t_s k}$ are used.

5. For each $s = 1, \ldots, S$, determine $\tilde{t_{sk}} < \tilde{t_{qk}}, q = 1, \ldots, S$, denoting $\tilde{I_q}^s = I[\tilde{t_{sk}} \leq \tilde{t_{qk}}]$.
6. Compute $\tilde{p_{t_k}}^s = \frac{\sum_{q=1}^{S} \tilde{I_q}^s}{S}$
7. For each s, determine $p^s_{BOSS} = \min_k \tilde{p_{t_k}}^s$ and $\hat{p}_{BOSS} < p^s_{BOSS}$ denoting $I^s_{BOSS} = I[\hat{p}_{BOSS} < p^s_{BOSS}]$
8. Compute $\hat{p}^c_{BOSS} = \frac{\sum_{s=1}^{S} I^s_{BOSS}}{S}$

The importance-weighted Monte Carlo (ISMC) algorithm is similar, but with the key difference that the samples are drawn from a different distribution. Let $f(x)$ denote the multivariate normal density function evaluated at vector x with mean zero and known covariance under the null hypothesis as in Step 2.(a) above. It becomes necessary to determine a 'better' density g from which rare events (i.e., small p-values) can be sample in order obtain better estimates of the true significance of the observed test statistic. While there are many reasonable choices, choosing g to be a multivariate gaussian with mean 0 with covariance $v\Sigma$ where v is a scalar slightly larger than 1 works well in practice.

The Importance-Weighted version of Monte Carlo algorithm for computing an estimate of the true BOSS p-value that accounts for the dependence between tests is

1. Determine $\hat{p}_k$ and the corresponding $\hat{t}_k$ for each ordered subset $k = 1, \ldots, r$ as described above.
2. For iterations $s = 1, \ldots, S$
   (a) Sample $\tilde{\beta}_s \sim g = N((0, \ldots, 0), v\Sigma)$.
   (b) Compute the importance weight $w_s = f(\tilde{\beta}_s)/g(\tilde{\beta}_s)$

(c) Compute the 2-sided p-values $\tilde{p_{sk}}$ of $\tilde{\beta_{sk}}$, as well as the corresponding $\tilde{t_{sk}}$ for each ordered subset.

(d) Determine whether $\tilde{t_{sk}} < \hat{t}_k$, denoting $\tilde{I}_s = I[\tilde{t_{sk}} < \hat{t}_k]$, which equals when the inequality holds and zero otherwise.

3. Compute $\hat{p}_{t_k} = \frac{\sum_{s=1}^{S} w_s \tilde{I}_s}{S}$

4. Determine $\hat{p}_{BOSS} = \min_k \hat{p}_{t_k}$.

5. For each $s = 1, \ldots, S$, determine $\tilde{t_{sk}} \leq \tilde{t_{qk}}, q = 1, \ldots, S$, denoting $\tilde{I_q}^s = I[\tilde{t_{sk}} \leq \tilde{t_{qk}}]$.

6. Compute $\tilde{p_{t_k}}^s = \frac{\sum_{q=1}^{S} \tilde{I_q}^s}{S}$

7. For each s, determine $p^s_{BOSS} = \min_k \tilde{p_{t_k}}^s$ and $\hat{p}_{BOSS} < p^s_{BOSS}$ denoting $I^s_{BOSS} = I[\hat{p}_{BOSS} < p^s_{BOSS}]$.

8. Compute $\hat{p}^c_{BOSS} = \frac{\sum_{s=1}^{S} w_s I^s_{BOSS}}{S}$.

In practice, the scaling parameter v must be tuned to ensure reasonable estimates are obtained. If v is too small, the algorithm results in similar output to the standard Monte Carlo algorithm but with the additional cost of computing w_s. If v is too large, the importance weights will be extremely small, resulting in a small estimate of $\hat{p}^c_{BOSS}$ regardless of the true value.

4.5 Applications

In order to demonstrate both the utility and versatility of BOSS three applications in genomics are presented. These applications were selected to highlight the ability of BOSS to robustly combine large numbers of p-values, as well as computationally scale to a large number tests. In the first application, Combining Evidence of the Joint Location-Scale Test in Section 4.5.1, focus is on comparing different approaches to BOSS p-value computation and interpreting the results with respect to other methods

in the two parameter setting using real world data [102]. The second application, Analysis of the Apical Plasma Membrane Gene-set in Section 4.5.2, focuses on single gene-test test that requires combining evidence across thousands of correlated p-values [103]. The third application, Pleiotropic Signal in UK Biobank Summary Statistics in Section 4.5.3, combines evidence across a large number of correlated p-values when over a million association tests are conducted [104, 105].

4.5.1 Combining Evidence of the Joint Location-Scale Test

The joint location-scale test [102] aims to combine evidence across two tests: differences of mean (location) and variance (scale). Explicitly, consider a test for association of a phenotype Y with a genomic variant that takes three values $X \in \{0, 1, 2\}$. The linear model can be written as the following

$$Y \sim \beta_0 + \beta_1 X + \epsilon.$$

Additionally, it is possible to test whether there are differences in the variances σ_i^2 for $i = 0, 1, 2$ of Y between each of the categories of X via Levene's test [106]. Thus, the null hypothesis for the joint location-scale test of a single variant is:

$$H_0 : \beta_1 = 0 \text{ and } \sigma_i = \sigma_j \forall i \neq j, i, j = 0, 1, 2 \tag{4.4}$$

$$H_1 : \beta_1 \neq 0 \text{ or } \sigma_i \neq \sigma_j \text{ for some } i \neq j. \tag{4.5}$$

While the null hypothesis in Equation 4.5.1 tests both the differences in mean and variances simultaneously, these tests are independent under the assumption that Y is normally distributed [102]. Because of this independence, the p-value for each test can be combined using Fisher's method [89], the minimum p-value (MinPV), or BOSS.

Because the benefits of BOSS over Fisher's method and MinPV method arise when relatively few p-values exhibit non-null signal, the outcomes of the three methods in the two parameter case do not differ drastically. Nevertheless, the comparison of the results presented here demonstrate how BOSS provides a result that is simultaneously more simplified (i.e., a single p-value output), as well as interpretable (i.e., which variables are included and in what order) when compared to using the results of both Fisher's method and the MinPV method.

Table 4.1.: Joint Location Scale Results. The top ten most significant single nucleotide variants (SNPs) ordered by BOSS p-value, as well as a comparison with the MinPV and Fisher's methods. BOSS p-values are computed using the Rao-Blackwellized Monte Carlo (RBMC) method. The Location and Scale columns indicate p-value order in the most significant subset. For example, the most significant variant, rs11611796, includes only the scale p-value in the most significant subset. Another variant, rs2399880, includes both the location and scale p-values, albeit with the location p-value being ordered as more significant than the scale p-value.

SNP Name	BOSS	MinPV	Fisher's	Location	Scale
rs11611796	2.8e-08	1.8e-08	1.4e-07	0	1
rs1995604	1.2e-06	7.1e-07	1.1e-06	0	1
rs7127354	3.0e-06	1.8e-06	2.2e-06	0	1
rs12067773	7.8e-06	4.8e-06	1.6e-05	0	1
rs4561812	1.3e-05	3.5e-05	7.7e-06	1	2
rs6824903	1.3e-05	1.1e-05	7.9e-06	2	1
rs12847085	1.5e-05	9.2e-06	1.5e-05	1	0
rs2192379	1.5e-05	1.2e-05	9.4e-06	2	1
rs2399880	1.6e-05	6.1e-05	9.8e-06	1	2
rs10138671	1.7e-05	1.0e-05	2.6e-05	0	1

The data used for the application of BOSS to the joint location-scale test are permutation-based p-values of tests of differences in mean and variance within the categories of a single-variant with cystic fibrosis severity [102]. In total, there are $565,884$ variants across the genome for which the location and scale p-values are available. Comparison of the outcomes for all BOSS, MinPV, and Fisher's are shown in Figure 4.2. The most significant BOSS p-values are shown in Table 4.1. As mentioned previously, the combined p-values for each method do not change much

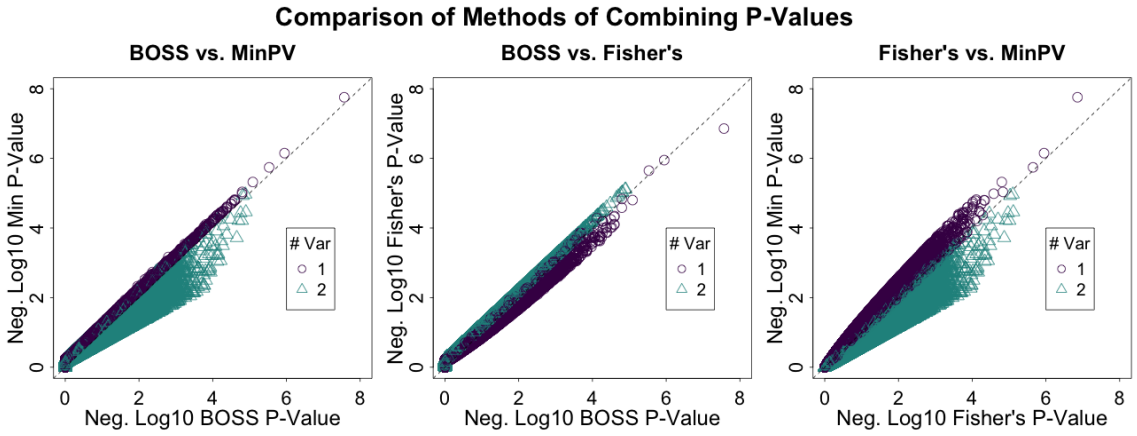

Figure 4.2.: Comparison of Methods of Combining P-Values. Scatterplots of the negative log10 p-values from BOSS and MinPV (Left), BOSS and Fisher's (Center), and Fisher's and MinPV (Right). The BOSS p-values are computed using the RBMC method, where the legend indicates the number of variables selected in the best ordered subset. While only one variant is statistically significant (when controlling for the number of tests considered) using the MinPV method, there are many variants that are one or two orders magnitude more significant when using BOSS compared to the MinPV. There is less of a difference in p-values when considering Fisher's method compared to BOSS as there are only two ordered subsets being considered. By using BOSS, it is possible to eliminate the choice of whether to use the MinPV method or Fisher's method, which often differ dramatically.

when the results of the three methods are compared. It is worth noting, however, that the most significant variant (rs11611796) is statistically significant after a Bonferonni correction ($\alpha_c = 0.05/565,884 = 8.8e-08$) only using the MinPV ($p = 1.8e-08$) and not Fisher's method ($p = 1.4e-07$). Rather than choosing the most significant combined p-value of these two methods, BOSS considers both methods simultaneously without inflating the type I error rate. The result of BOSS for rs11611796 ($p = 2.8e-08$) is significant at the aforementioned threshold, and indicates that the best ordered subset consists of only the scale test p-value. The various methods of computing BOSS p-values are compared in Figure 4.3.

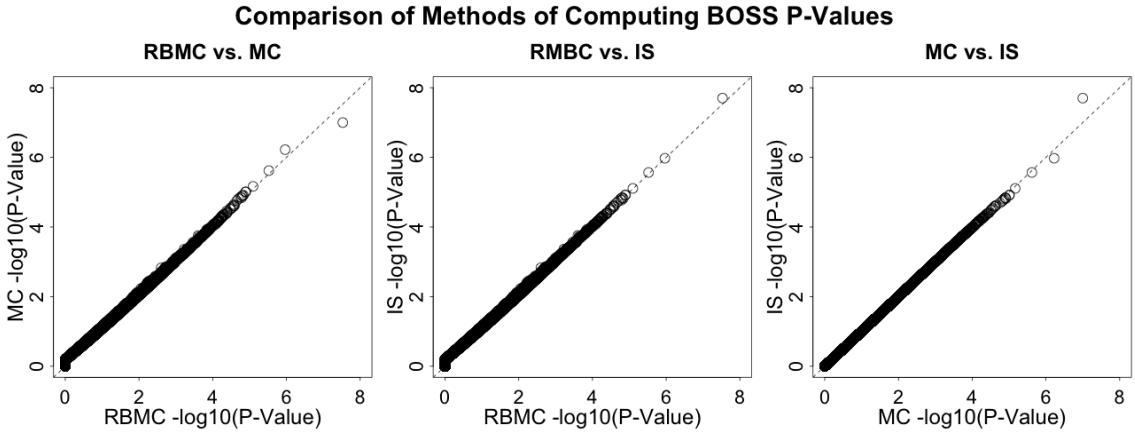

Figure 4.3.: Comparison of Methods of Computing BOSS P-Values. Scatterplots of the BOSS negative log10 p-values using Rao-Blackwellized Monte Carlo (RBMC) and vanilla Monte Carlo (MC) methods (Left), Rao-Blackwellized Monte Carlo and Importance Sampling (IS) (Center), and vanilla Monte Carlo and Importance Sampling (Right) for the joint Location-Scale test data. For these results, the RBMC method used only $1,000$ samples whereas the MC and IS methods used $10,000,000$. The IS methods uses a variance scaling parameter of 1.1. While the estimated BOSS p-values are quite similar for most p-values, the MC method is not able estimate the most extreme BOSS p-value (estimated as $1.99e-8$ using IS and $2.96e-8$ using RBMC), which is set to $1e-7$ in the plots above as no null distribution samples were more extreme. These results show that the use of RBMC allows for estimating extreme BOSS p-values with relative few iterations. Additionally, the use of IS can estimate p-values more extreme (by order of magnitude here) than the vanilla Monte Carlo approach.

4.5.2 Analysis of the Apical Plasma Membrane Gene-set

While single variant genome-wide association studies are useful as an initial analysis, scientists are often interested in testing more specific hypotheses about a set of genomic variants located within a gene or a group of genes known to be related to some biological function, i.e., a gene pathway. In a cystic fibrosis study [103], investigators tested the association of variants across the human genome with a severity measure of cystic fibrosis. In multiple follow-up investigations within the same study, researchers sought to understand whether there is a link between variants in the cystic fibrosis transmembrane conductance regulator (CTFR) gene, as well as other genes annotated for specific biological functions, and infant bowel obstruction, which is common among patients with cystic fibrosis. Using only summary data of test statistics and the correlation matrix, it is possible to re-analyze these data using the BOSS framework and then compare to the original published results [103]. It is worth noting that the main obstacle to successfully employing any summary statistics-based test on genomic variants is the inherent correlation found in the data. Large groups of variants that are located close together in the genome are often correlated, thus violating the assumption that the p-values are independently distributed, uniform random variables under the null hypothesis. This is a prime application for the BOSS method to be employed since only a small number of variants are expected to contribute to any signal under the alternative hypothesis. In what follows, the advantages of BOSS in terms of inference and interpretation are explored. Further, the numerical and algorithmic features of BOSS that are required for efficiently estimating extremely small p-values in the high-dimensional setting are highlighted.

Data Pre-processing

The work of Sun et. al [103] is focused on the associating meconium ileus (severe bowel obstruction in infants occurring in 15% of patients with cystic fibrosis) with $3,814$ variants within 155 genes encoding constituents of the apical plasma membrane.

The protein CFTR resides in the apical plasma membrane. While the p-values and correlation matrix are the only data required to apply BOSS, there were two modifications made to the original data for this new analysis. First, the variant rs4077468 was removed as it is perfectly correlated with, and has the same p-value as, a neighboring variant rs4077469 implying it is effectively offering redundant information. Second, because the correlation matrix of variants was computed in a pairwise fashion, the nearest positive definite [1] correlation matrix was used in the simulation of the null hypothesis.

Results

The 3,814 univariate p-values resulting from testing association of meconium ileus as well as the corresponding $3,814 \times 3,814$ correlation matrix between variants are used as data. Of the variants selected using BOSS, the marginal p-values ranged from 9.88×10^{-9} to 0.053, with only two surpassing the genome-wide significance threshold of 5×10^{-8} [107]. Using the full dataset with permutations to account for the correlation of variants, researchers in [103] reported a p-value of 2×10^{-4} when considering the combined evidence of all variants. By comparison, the BOSS p-value, accounting for correlation, resulted in a p-value of 6.26×10^{-5} with 307 variants contributing to the best ordered subset. This result suggests that BOSS detects slightly more signal than the existing method when considering all ordered subsets of the p-values.

4.5.3 Pleiotropic Signal in UK Biobank Summary Statistics

The UK Biobank [104] is an open-access resource that includes genotyping, electronic health record, health survey, and medical imaging data. Because the UK Biobank contains a large amount of genomic variants ($\sim$ 11 million variants for $500,000$ participants), as well as phenotypes ($\sim 2,000$ diseases and other traits),

[1] Using the nearPD function in the Matrix R package `http://Matrix.R-forge.R-project.org/`

it is an ideal dataset on which to employ BOSS. In order to analyze these data, it is necessary for any approach to accurately account for the correlation of phenotypes, as well as be computationally efficient.

Data Pre-Processing

A pre-processed version of UK Biobank data was obtained through the Benjamin Neale's lab web page [105]. Of the original $500,000$ participants, summary statistics from $337,000$ unrelated participants were downloaded across 191 well-characterized phenotypes and nearly 11 million variants. In order to obtain a reasonable estimate of the correlation of phenotypes, a number of pre-processing steps were applied to mitigate the influence of extreme test statistics, as well as to limit the effect of the natural correlation across variants in close proximity in the genome. The latter effect being referred to as linkage disequilibrium (LD) [108]. The first of these steps clips the test statistics to a range of $(-5, 5)$, which translates to the normal distribution quantile of $1.5e-06$. The second step sub-samples every 100^{th} test statistic, which are ordered according to their location in the genome. To limit our analysis to well-characterized regions of the genome, 1.3 million variants of subjects from European ancestry in 1000 Genomes Project were used for this analysis [109]. While the subset of variants is substantially smaller than the original, $\sim$ 11 million variants, the importance of accurate estimation of the correlation of phenotypes justifies limiting the number of variants.

Results

The primary motivation for applying BOSS to the UK Biobank-based data is to determine whether combining evidence across phenotypes is useful in finding novel genomic loci associated with disease. Furthermore, this application not only showcases the scalability of the BOSS software by making use of parallel processing for simulating the null distribution, but also demonstrates the algorithmic optimization

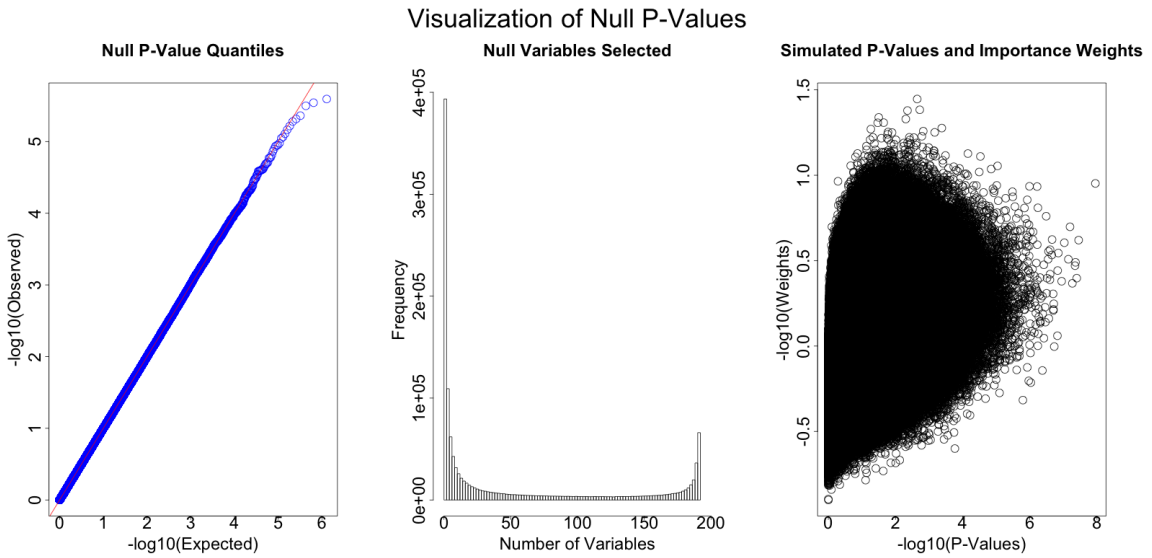

Figure 4.4.: Simulation Study of Null Test Statistics. In order to ensure that the BOSS p-values are uniformly distributed, the first step simulates test statistics with zero mean and covariance identical to the UK Biobank data. A total of 10 million importance weighted samples with a variance scaling of 1.05 are used to estimate the BOSS p-values. (Left) The quantile plot of BOSS p-values resulting from the ~ 1.3 null test statistics illustrates that the null p-values are uniformly distributed. (Center) The number of variables contributing to most significant combination of null p-values. (Right) The relationship between the simulated p-values and importance sampling weights suggest the importance sampling distribution does not result in numerically unstable behavior.

that estimates large numbers of p-values [110]. Given that there are $1,285,100$ variants, a tremendous number of iterations are required to estimate extreme p-values with respect to the null distribution. Here, the results of applying BOSS to the subset of the UK Biobank data described previously using 10 million null iterations, and an importance sampling scheme using the scaling parameter of 1.05 are presented.

Before applying BOSS to the UK Biobank summary statistics, it is useful simulate null-distributed test statistics to assess the importance sampling algorithm. It is necesary to verify that if we were to generate test statistics from the null distribution with a zero mean and correlation estimated from the UK Biobank data, that BOSS would result in uniform p-values. This is indeed the case as illustrated in Figure 4.4. Using the same importance weighted p-values as in the null simulation study,

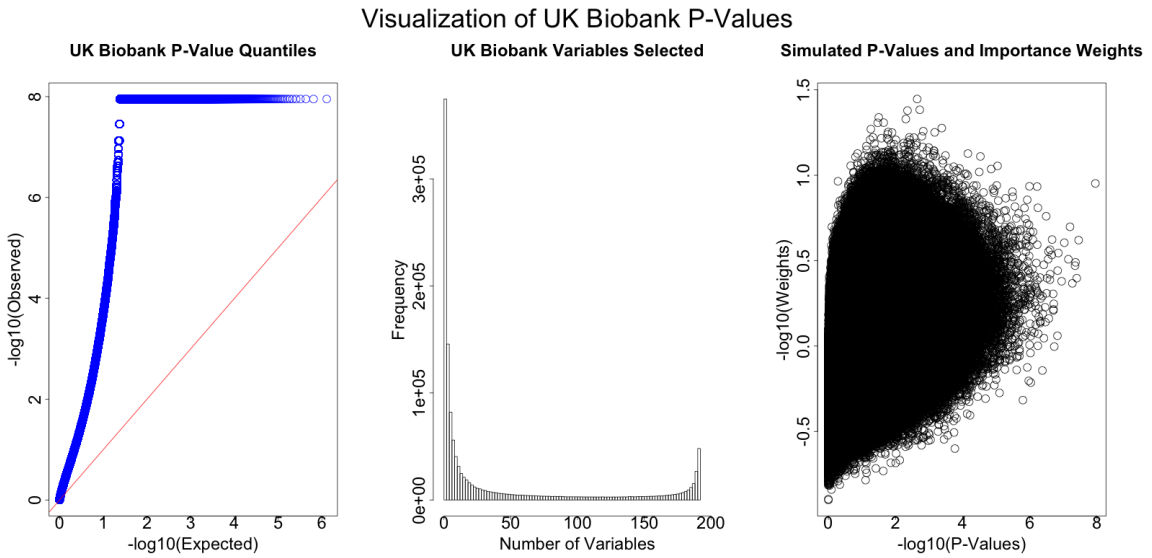

Figure 4.5.: BOSS Results of UK Biobank P-Values. The same 10 million importance weighted samples with a variance scaling of 1.05 were used to estimate the BOSS p-values for the actual UK Biobank test statistics. (Left) The quantile plot of BOSS p-values resulting from the $\sim$ 1.3 million UK Biobank test statistics shows that there is inflation of the negative $\log_{10}$ p-values. (Center) The number of variables contributing to the most significant combination of UK Biobank p-values. (Right) The relationship between the simulated p-values and importance sampling weights are shown again to emphasize that the same set of simulated p-values and importance weights were used to compute the BOSS p-values.

the BOSS results of the UK Biobank data is shown in in Figure 4.5. While the UK Biobank p-values are highly non-uniform in distribution (even when considering that some p-values are non-uniformly distributed due to real signal), the inflation of significance in genome-wide association studies is to be expected due to population stratification, cryptic relatedness, and polygenic inheritance [111].

4.6 Simulation Studies

A number of simulation studies were performed to assess properties of BOSS under the null and alternative hypotheses. The first purpose of these studies is to investigate the the type I error rate control of BOSS. Additionally, the null distributions of the BOSS test in the case of the independent and dependent p-values under the null hy-

pothesis are compared to Fisher's and MinPV methods. Finally, a comparison of the power under different numbers of significant parameters are considered. Additional details of these simulations if found below.

To better understand the dynamics of type I error rate control, BOSS using the RBMC and the proposed correction for the effective number of tests was applied to simulated uniform p-values using 30 parameters. In Figure 4.7, the observed versus the expected quantiles are plotted showing that this approach is slightly conservative above the $a = 0.05$ threshold and effectively controls type I error rate. Figures 4.7 and 4.8 show the null distribution quantile plots and histograms of the BOSS, Fisher's, and MinPV methods for the RBMC (for independent p-values) and ISMC (for dependent p-values) methods, respectively. For the dependent p-value simulation, the first 30 rows and columns of the correlation matrix of the apical gene-set (described in Section 4.5.2) were used to simulate test statistics with zero mean. These quantile plots illustrate that quantiles of the null-distributed p-values are indeed uniform for all the methods considered in both the independent and dependent scenario. With the exception of BOSS using RBMC, for which the p-values are scaled to control the type I error rate, all distributions are indeed uniformly as judging by the histograms. Figure 4.10 demonstrates the power of different methods when simulating the alternative hypothesis with a fixed amount of explained variation is spread across different numbers of parameters. Taken together, these simulations show the implementation of BOSS performs as expected under the null and alternative hypotheses.

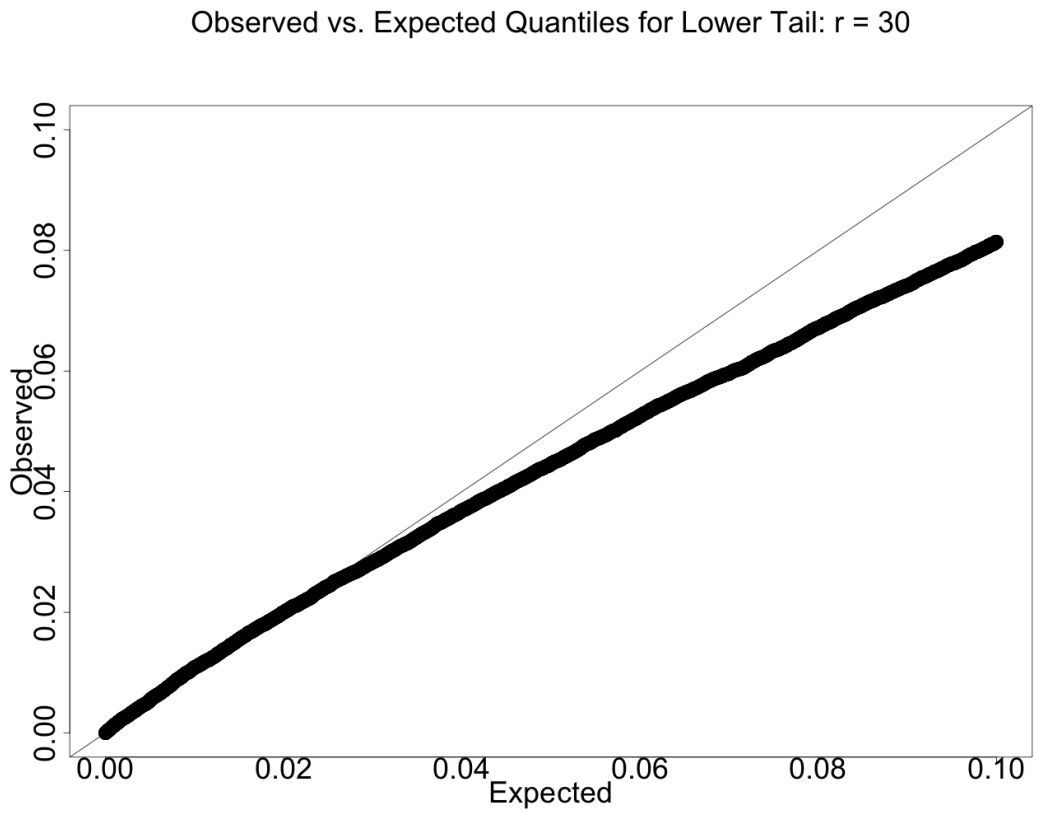

Figure 4.6.: Lower Quantile of the Null Distribution. Inspecting the null distribution quantile plot shows that the BOSS method using RBMC is slightly conservative above the 0.05 cutoff.

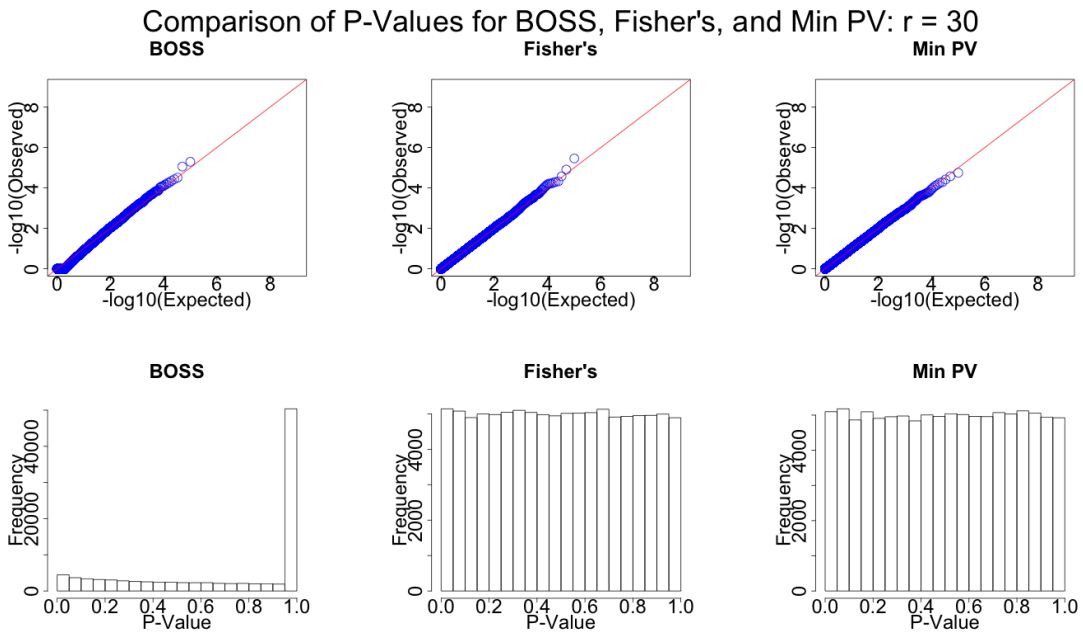

Figure 4.7.: Comparison of Null Distributions from Combining Independent P-values. The quantile plots of BOSS, Fisher's and the MinPV method all have tail distributions consistent with the uniform distribution. While Fisher's and the Min PV method are uniformly distributed in the histograms, the scaling of the RBMC BOSS p-values result in many p-values close to one.

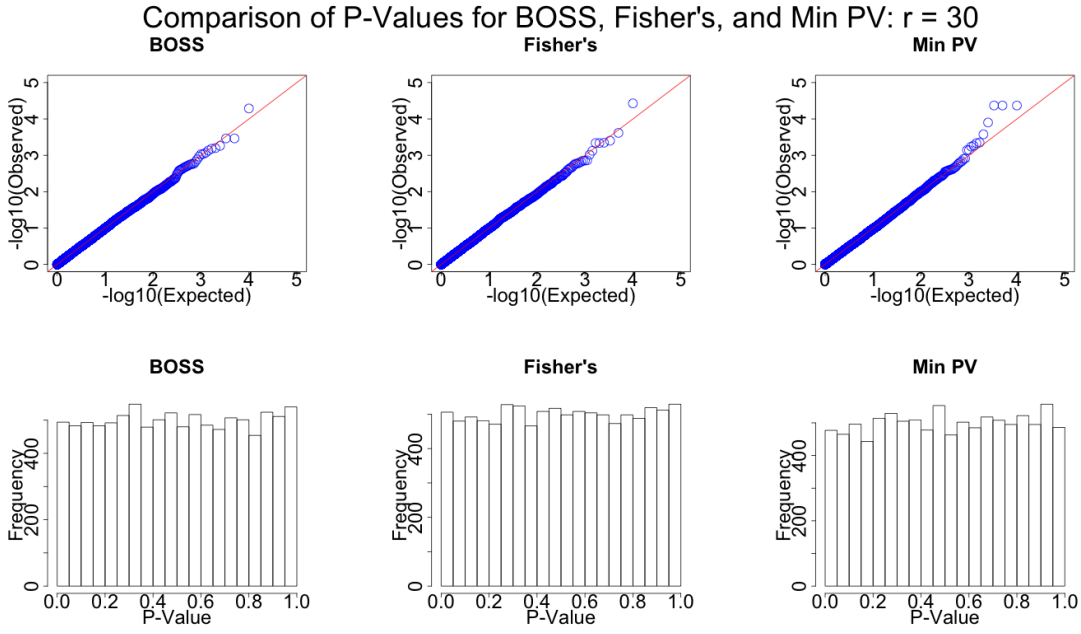

Figure 4.8.: Comparison of Null Distributions from Combining Dependent P-values. The quantile plots and histograms of the BOSS, Fisher's and the MinPV methods using importance sampling all have null distributions consistent with the uniform distribution.

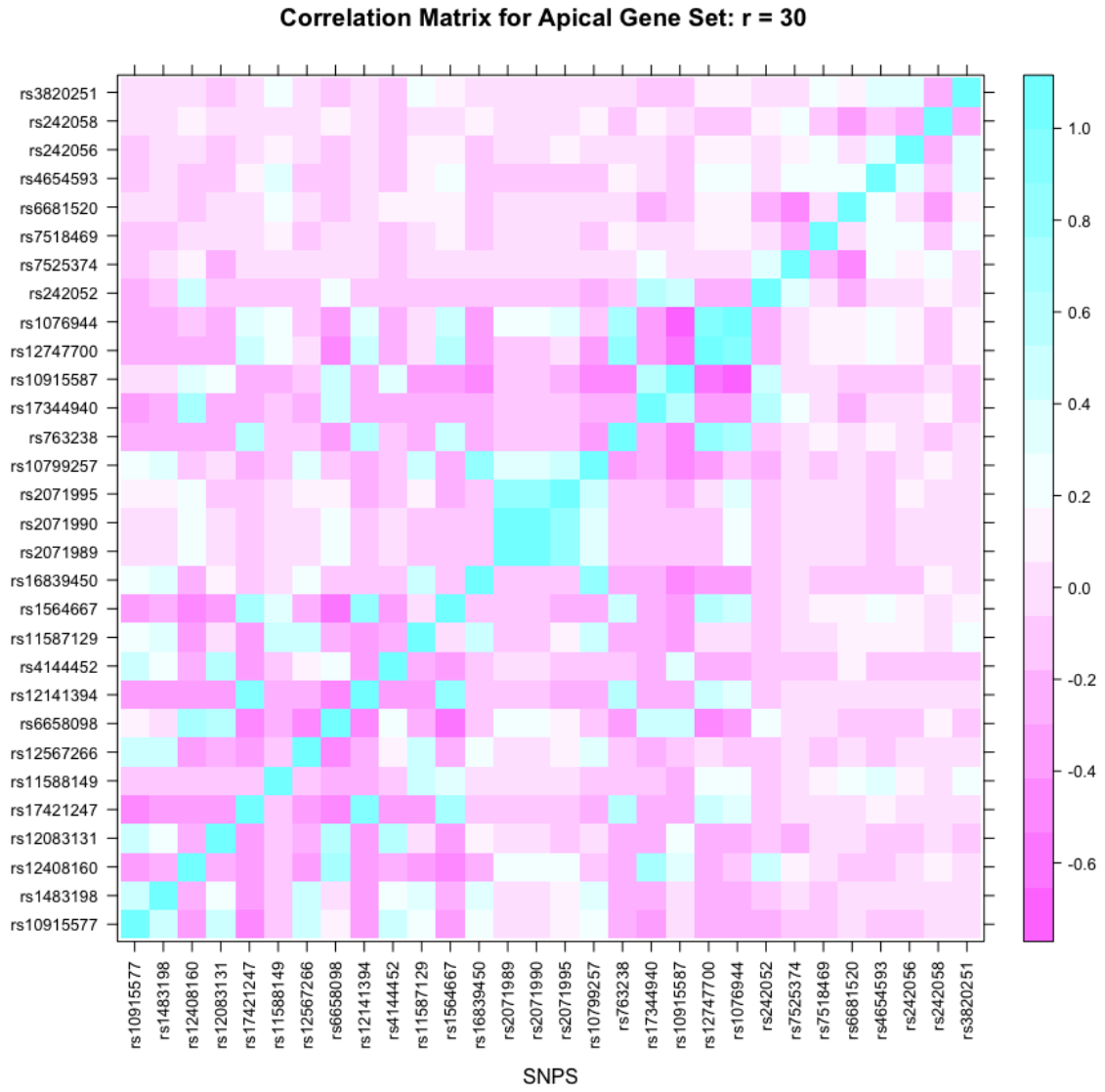

Figure 4.9.: Correlation Matrix of Thirty Variants in the Apical Gene Set. This figure shows the correlation matrix used for the simulation of dependent p-values and analysis of the null distributions of the BOSS, Fisher's, and Min PV methods using importance sampling.

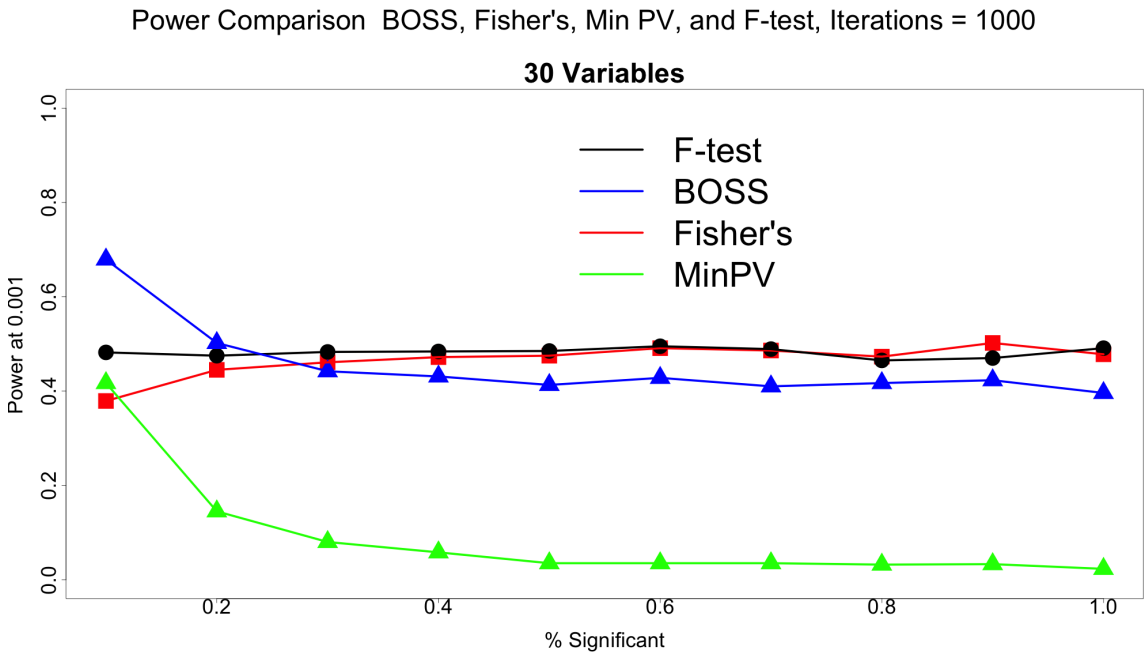

Figure 4.10.: Simulation Study Assessing Power. By simulating non-null relationships between a response variable and thirty explanatory variables with a fixed amount of explained variation, we can compare the power of the various methods. While the BOSS, Fisher's, and Min PV methods use the marginal p-values resulting from a linear model including one parameter at a time, the F-test uses a linear model with all parameters at once. These results indicate that power of each method depends on the percentage of non-null parameters in the model. While BOSS has slightly less power compared to Fisher's method and the F-test when the non-null signal is spread across many parameters, it outperforms other methods when the signal is spread across relatively few parameters.

4.7 Discussion

The work presented here explores various aspects of adaptive testing in the context of exponential order statistics. As explained, BOSS not only combines evidence from multiple sources but is also able to determine the number of variables that contribute to the resulting significant subset of p-values. By employing various computational strategies, BOSS can be efficiently applied to summary statistics of large-scale experiments. These features are extremely useful in the context of genome-wide association studies where deeper interpretation of results can elucidate new directions of scientific investigation.

5. CONCLUSION

As the pace of innovation in technology continues to accelerate, especially when measuring molecular, genomic, and biological phenomena, so to does the number, size, and quality of the data. This trend has fortunately coincided with unprecendented levels of computational resources including both specialized hardware, as well as the availability of open-source algorithms and software. While necessary, these circumstances are not sufficient to fully capitalize on the potential of scientific advancements and the potential to improve human lives. Analysis, interpretation, and validation will always be the most difficult, if not time consuming, hurdle in reaping the rewards of data-driven science. The latter point is not meant to downplay the essential and impressive feats of science and engineering in biology, but rather to recognize inherent difficulties in creating scientific consensus, as well as the high standards within any regulatory process that oversees the approval and use of medical treatments or diagnostics in humans.

Within the context of a specific scientific use-case, scientific advancements, computational resources, and data-driven insights are, of course, intricately connected. The fidelity of modern measurement technologies depends on robust pre-processing of large amounts of raw data. Determining the amount of data to be generated is governed by the fixed and marginal cost of scientific experiments, as well as the minimum quantity needed to discern statistical signal relative to technical and biological noise. The benefits of even incremental modeling improvements can be far reaching if applied to a vast number of scientific experiments so long as they do not exceed real world computational constraints.

5.1 Summary

The motivation and goal for this work is to explore the statistical and computational challenges involved in obtaining insights from a variety of real world data sets. Even with the acknowledgement that each application is biological in nature, there is a great deal of overlap in the approaches required to effectively analyze these data. Deep learning models (Section 2.3) provide a flexible framework for predicting the various tasks involved in quantifying protein expression of electrophoretic cytometry images. As presented in Section 3.2, a similar modeling framework can be employed to flag potentially cancerous mutations in both research and clinical applications by predicting the annotated class of a genomic sequence input, which is of an entirely different form. Further, using experimental annotation for a single variant or to define sets of variants, the methods developed in Section 4.2 inform the genomic basis of human disease in a more interpretable way. Each of these applications also share broader commonalities beyond the modeling approaches and data formats within the areas of technical variation, model evaluation, and data integration.

5.1.1 Accounting for Technical Variation in Data Processing

Decisions regarding how data are prepared, processed, and used can have important consequences in both data analysis and interpretation. For example, testing correlation of protein expression using scEPC images described in Section 2.4.2 requires robust quality control, as well as accurate quantification of the biological signal. In order to avoid bias in the prediction of cancerous mutations, careful consideration must be given to how data are sampled for training models as described (Section 3.3.3). Because of the shear size and differences in quality of the UK Biobank data (Section 4.5.3), only a subset of variants and phenotypes supplied data for the application of BOSS. Independent of the technology, size, and quality, data preprocessing and analysis will always have to take into account technical variation. Failure to understand and/or remove such variation represents a missed opportunity to improve

the precision of results at best, and a path to making erroneous scientific conclusions at worst.

5.1.2 Evaluating a Model in Terms of the End Use Case

Without well defined metrics, assessing the utility of computational models can be difficult, if not impossible. The context of the end use case can not only shine light on the required performance of a model but also inform decisions on how a model is used. This notion is apparent in Section 2.5.2, where including more, though potentially lower quality, protein expression estimates lead to stronger correlation signal. While the models in Section 3.3 are able to detect previously unseen cancerous mutations, the relative infrequency of these mutations raise the requirements of predictive performance in actual clinical applications. The development of more efficient algorithms for BOSS in Section 4.4 are motivated by the large number of iterations needed to estimate p-values at the extremes required when considering millions of statistical tests.

5.1.3 Integration of Outside Data

Using information from various sources can inform and enable the modeling of complex biological data. The creation and use of the ground truth dataset in Section 2.5.2 relied on known relationships between proteins, the absence of which would have made model evaluation less straightforward. Training a deep learning model to accurately predict the class of somatic mutations in Section 3.3 would not have been possible without the large set of annotations aggregated from various data sources in the publicly available COSMIC Database [40]. Though there are many ways of combining evidence in genome-wide association studies, the applications in Section 4.5 illustrate how BOSS provides a more interpretable means of determining which variables contribute to significant results.

5.2 Future Work

The applications discussed in this work suggest various directions for further investigation. As technologies continue to generate more data, these issues introduced here will only require more attention. Investing in statistical and computational improvements can have an outsized impact because each improvement can be applied to all future experiments involving specific data types. While each application is different, the common themes (the impact of technical variation, realistic model evaluation, and integration of outside data) connecting the analysis of these data serve as general lessons that inform others.

5.2.1 Classification, Segmentation and Quantification of Electrophoretic Cytometry Images

While the work in Chapter 2 develops models for quantifying protein expression of scEPC images, further investigation is needed in the evaluation of the model performance in the protein expression pipeline as a whole. Testing correlation when two proteins are known to be correlated is a useful proof of principle, but overlooks issues that may impact statistical significance. Potential issues may arise due to technical artifacts present in multiple florescent channels at a certain location of an image. This scenario might inflate estimates of protein co-expression if such artifacts are not removed via quality control or denoising. Because the number of quantified protein expression data from each experiment is relatively small, it is feasible to employ permutation-based tests of correlation to mitigate the effect of skewed or non-normal data distributions. Linear models, accompanied by their standard assumption checking procedures, can also be used to understand and eliminate potential biases such as outliers or influential points [112] for making more robust scientific conclusions.

5.2.2 Classifying Genomic Mutations in Cancer Diagnostics

Deep learning model performance is often constrained by the amount of data available for training. The predictive ability of models used to classify cancerous mutations in Section 3.3 is far from perfect. One logical strategy for improvement of predictive performance is obtaining larger amounts of data or different data formats. Including additional data within the positive class such as unvalidated, but likely, mutations may provide some benefit given the high degree of class imbalance. While different hyperparameters (Section 3.3.3) were investigated in terms of differing genomic window sizes, the optimal data format is not clear. Considering even larger genomic windows or sampling more diverse genomic regions has potential to improve classification further. Data augmentation, the process of generating more training data by making alterations to existing data, is yet another avenue of increasing the size of the training dataset. Because this approach is not as straightforward in the genomics context as it is with images, where data augmentation often involves rotating or flipping an image thereby maintaining the class label, further investigation is needed.

While including more positive class examples may improve prediction, it is also possible that there are just too many negative class examples for a model to learn. For this situation, one strategy might be to simulate a large, but fixed, number of negative class examples rather than continuously simulating new examples. While a lot fewer examples would be seen during training, sampling fewer examples multiple times might lead to more confident predictions within the negative class . Information within a larger diversity examples could be obtained by using an ensemble of models trained on different negative class data.

There are also many avenues for improving the classification of cancerous mutation by altering the data sampling mechanism used to simulate negative class mutations. Current hyperparameters for data sampling settings may be unknowingly biasing the training procedure. For example, genomic windows for the negative class are

simulated within a certain range of existing positive class examples. While sampling a larger range may result in a model a broader genomic context, negative examples within a genomic context more localized to a positive example might be more useful in discerning between the two classes.

5.2.3 Best Ordered Subset Selection

The algorithms used within BOSS in Section 4.4 were developed to estimate p-values at the extremes dictated by the large number of tests in genome-wide association studies. While the Rao-Blackwellized Monte Carlo algorithm can effectively estimate small p-values under the assumption of independence, many more samples are necessary when p-values are correlated. While ten million ($1e7$) importance weighted samples across 191 phenotypes achieved the required scale ($5e-8$) of estimated p-values, memory constraints limit the ability to use more samples. Because many of the steps in computing BOSS p-values involve only a subset of the data and are easily parallelized, more efficient and accessible data formats could be used to scale these sampling algorithms further.

5.3 Novel Contributions

This work focuses on the development of computational models in three biological applications, emphasizing statistical concepts in the evaluation and interpretation of results. Using the labels generated from an existing pipeline, Chapter 2 demonstrates an end-to-end deep learning framework for quantifying protein expression and correlation from single-cell electrophoretic cytometry images. Chapter 3 explores the use of convolutional neural networks in classifying somatic mutations and illustrates their ability to generalize to highly class-imbalanced data from a held-out region of the genome. In Chapter 4, an alternate theoretical formulation coupled with efficient sampling procedures is developed to create a scalable and interpretable method of combining p-values.

REFERENCES

[1] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press Cambridge, 2016.

[3] Christopher Bishop. Pattern recognition and machine learning. *Pattern Recognition and Machine Learning*, 2006.

[4] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

[5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[6] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *arXiv preprint arXiv:1710.05381*, 2017.

[7] Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature Protocols*, 11(9):1757, 2016.

[8] Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. Genotype and snp calling from next-generation sequencing data. *Nature Reviews Genetics*, 12(6):443, 2011.

[9] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 112. Springer, 2013.

[10] Jian Yang, Teresa Ferreira, Andrew P Morris, Sarah E Medland, Pamela AF Madden, Andrew C Heath, Nicholas G Martin, Grant W Montgomery, Michael N Weedon, Ruth J Loos, et al. Conditional and joint multiple-snp analysis of gwas summary statistics identifies additional variants influencing complex traits. *Nature Genetics*, 44(4):369, 2012.

[11] Todd A. Duncombe, Augusto M. Tentori, and Amy E. Herr. Microfluidics: reframing biological enquiry. *Nature Reviews Molecular Cell Biology*, 16(9):554–567, August 2015.

[12] Lionel B. Ivashkiv and Laura T. Donlin. Regulation of type I interferon responses. *Nature Reviews Immunology*, 14(1):36–49, December 2013.

[13] Judith Campisi and Fabrizio d'Adda di Fagagna. Cellular senescence: when bad things happen to good cells. *Nature Reviews Molecular Cell Biology*, 8(9):729–740, September 2007.

[14] Chi-Chih Kang, Kevin A Yamauchi, Julea Vlassakis, Elly Sinkala, Todd A Duncombe, and Amy E Herr. Single cell–resolution western blotting. *Nature Protocols*, 11(8):1508, 2016.

[15] Alex J Hughes, Dawn P Spelke, Zhuchen Xu, Chi-Chih Kang, David V Schaffer, and Amy E Herr. Single-cell western blotting. *Nature Methods*, 11(7):749–755, 2014.

[16] Augusto M Tentori, Kevin A Yamauchi, and Amy E Herr. Detection of isoforms differing by a single charge unit in individual cells. *Angewandte Chemie*, 128(40):12619–12623, 2016.

[17] A. M. Streets, X. Zhang, C. Cao, Y. Pang, X. Wu, L. Xiong, L. Yang, Y. Fu, L. Zhao, F. Tang, and Y. Huang. Microfluidic single-cell whole-transcriptome sequencing. *Proceedings of the National Academy of Sciences*, 111(19):7048–7053, May 2014.

[18] A. K. White, M. VanInsberghe, O. I. Petriv, M. Hamidi, D. Sikorski, M. A. Marra, J. Piret, S. Aparicio, and C. L. Hansen. High-throughput microfluidic single-cell RT-qPCR. *Proceedings of the National Academy of Sciences*, 108(34):13999–14004, August 2011.

[19] Daphne H E W Huberts, Sung Sik Lee, Javier GonzÃąlez, Georges E Janssens, Ima Avalos Vizcarra, and Matthias Heinemann. Construction and use of a microfluidic dissection platform for long-term imaging of cellular processes in budding yeast. *Nature Protocols*, 8(6):1019–1027, May 2013.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[21] David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Computational Biology*, 12(11):e1005177, 2016.

[22] Oren Z Kraus, Jimmy Lei Ba, and Brendan J Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016.

[23] Daniel Stoecklein, Kin Gwn Lore, Michael Davies, Soumik Sarkar, and Baskar Ganapathysubramanian. Deep learning for flow sculpting: Insights into efficient learning using scientific simulation data. *Scientific Reports*, 7:srep46368, 2017.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[25] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[26] Elly Sinkala, Elodie Sollier-Christen, Corinne Renier, Elisabet Rosàs-Canyelles, James Che, Kyra Heirich, Todd A Duncombe, Julea Vlassakis, Kevin A Yamauchi, Haiyan Huang, et al. Profiling protein expression in circulating tumour cells using microfluidic western blotting. *Nature Communications*, 8:14622, 2017.

[27] Kevin A Yamauchi and Amy E Herr. Subcellular western blotting of single cells. *Microsystems & Nanoengineering*, 3:16079, 2017.

[28] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[29] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

[30] Alan Agresti. *Categorical Data Analysis*, volume 482. John Wiley & Sons, 2003.

[31] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[33] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*, pages 2802–2810, 2016.

[34] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[35] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[36] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[38] Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, et al. The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic Acids Research*, 42(D1):D1001–D1006, 2013.

[39] Arthur Lesk. *Introduction to Genomics*. Oxford University Press, 2017.

[40] Sally Bamford, E Dawson, Simon Forbes, Jody Clements, Roger Pettett, Ahmet Dogan, A Flanagan, Jon Teague, P Andrew Futreal, Michael R Stratton, et al. The cosmic (catalogue of somatic mutations in cancer) database and website. *British journal of cancer*, 91(2):355–358, 2004.

[41] Xin Lai, Olaf Wolkenhauer, and Julio Vera. Understanding microrna-mediated gene regulatory networks through mathematical modelling. *Nucleic Acids Research*, 44(13):6019–6035, 2016.

[42] Lindsay S Shopland, Christopher R Lynch, Kevin A Peterson, Kathleen Thornton, Nick Kepper, Johann von Hase, Stefan Stein, Sarah Vincent, Kelly R Molloy, Gregor Kreth, et al. Folding and organization of a contiguous chromosome region according to the gene distribution pattern in primary genomic sequence. *The Journal of Cell Biology*, 174(1):27–38, 2006.

[43] Ludmil B Alexandrov and Michael R Stratton. Mutational signatures: the patterns of somatic mutations hidden in cancer genomes. *Current Opinion in Genetics & Development*, 24:52–60, 2014.

[44] Shai White-Gilbertson, David T Kurtz, and Christina Voelkel-Johnson. The role of protein synthesis in cell cycling and cancer. *Molecular Oncology*, 3(5-6):402–408, 2009.

[45] Natalya N Pavlova and Craig B Thompson. The emerging hallmarks of cancer metabolism. *Cell Metabolism*, 23(1):27–47, 2016.

[46] Ludmil B Alexandrov, Serena Nik-Zainal, David C Wedge, Samuel AJR Aparicio, Sam Behjati, Andrew V Biankin, Graham R Bignell, Niccolo Bolli, Ake Borg, Anne-Lise Børresen-Dale, et al. Signatures of mutational processes in human cancer. *Nature*, 500(7463):415, 2013.

[47] Serena Nik-Zainal, Ludmil B Alexandrov, David C Wedge, Peter Van Loo, Christopher D Greenman, Keiran Raine, David Jones, Jonathan Hinton, John Marshall, Lucy A Stebbings, et al. Mutational processes molding the genomes of 21 breast cancers. *Cell*, 149(5):979–993, 2012.

[48] Thomas Helleday, Saeed Eshtad, and Serena Nik-Zainal. Mechanisms underlying mutational signatures in human cancers. *Nature Reviews Genetics*, 15(9):585, 2014.

[49] Joshua D Campbell, Anton Alexandrov, Jaegil Kim, Jeremiah Wala, Alice H Berger, Chandra Sekhar Pedamallu, Sachet A Shukla, Guangwu Guo, Angela N Brooks, Bradley A Murray, et al. Distinct patterns of somatic genome alterations in lung adenocarcinomas and squamous cell carcinomas. *Nature Genetics*, 48(6):607, 2016.

[50] The Cancer Genome Atlas Webpage. `http://cancergenome.nih.gov/`. Accessed: 2018-05-15.

[51] Johan T den Dunnen, Raymond Dalgleish, Donna R Maglott, Reece K Hart, Marc S Greenblatt, Jean McGowan-Jordan, Anne-Francoise Roux, Timothy Smith, Stylianos E Antonarakis, Peter EM Taschner, et al. Hgvs recommendations for the description of sequence variants: 2016 update. *Human Mutation*, 37(6):564–569, 2016.

[52] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[53] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, 12(10):931–934, 2015.

[54] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26(7):990–999, 2016.

[55] David Harris and Sarah Harris. *Digital Design and Computer Architecture*. Morgan Kaufmann, 2010.

[56] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR, 2017.

[57] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[58] Ron Schwessinger, Maria C Suciu, Simon J McGowan, Jelena Telenius, Stephen Taylor, Doug R Higgs, and Jim R Hughes. Sasquatch: predicting the impact of regulatory snps on transcription factor binding from cell-and tissue-specific dnase footprints. *Genome Research*, 27(10):1730–1742, 2017.

[59] Carlo M Croce. Oncogenes and cancer. *New England Journal of Medicine*, 358(5):502–511, 2008.

[60] Igor B Rogozin, Youri I Pavlov, Alexander Goncearenco, Subhajyoti De, Artem G Lada, Eugenia Poliakov, Anna R Panchenko, and David N Cooper. Mutational signatures and mutable motifs in cancer genomes. *Briefings in Bioinformatics*, 19(6):1085–1101, 2017.

[61] Calvin Wing Yiu Chan, Zuguang Gu, Matthias Bieg, Roland Eils, and Carl Herrmann. Impact of cancer mutational signatures on transcription factor motifs in the human genome. *BMC Medical Genomics*, 12(1):64, 2019.

[62] Steven T Kothen-Hill, Asaf Zviran, Rafael C Schulman, Sunil Deochand, Federico Gaiti, Dillon Maloney, Kevin Y Huang, Will Liao, Nicolas Robine, Nathaniel D Omans, et al. Deep learning mutation prediction enables early stage lung cancer detection in liquid biopsy. 2018.

[63] The world health organization webpage. `https://www.who.int/news-room/fact-sheets/detail/cancer`. Accessed: 2019-02-20.

[64] Cancer research uk webpage. `https://www.cancerresearchuk.org/about-cancer/cancer-symptoms/why-is-early-diagnosis-important`. Accessed: 2019-02-20.

[65] Alexander M Aravanis, Mark Lee, and Richard D Klausner. Next-generation sequencing of circulating tumor dna for early cancer detection. *Cell*, 168(4):571–574, 2017.

[66] Emily Crowley, Federica Di Nicolantonio, Fotios Loupakis, and Alberto Bardelli. Liquid biopsy: monitoring cancer-genetics in the blood. *Nature Reviews Clinical Oncology*, 10(8):472, 2013.

[67] Bhuvan Molparia, Eshaan Nichani, and Ali Torkamani. Assessment of circulating copy number variant detection for cancer screening. *PloS One*, 12(7):e0180647, 2017.

[68] Fatima Zare, Michelle Dow, Nicholas Monteleone, Abdelrahman Hosny, and Sheida Nabavi. An evaluation of copy number variation detection tools for cancer using whole exome sequencing data. *BMC Bioinformatics*, 18(1):286, 2017.

[69] Ivan A Zaporozhchenko, Anastasia A Ponomaryova, Elena Yu Rykova, and Pavel P Laktionov. The potential of circulating cell-free rna as a cancer biomarker: challenges and opportunities. *Expert Review of Molecular Diagnostics*, 18(2):133–145, 2018.

[70] Bryan C Ulrich and Cloud P Paweletz. Cell-free dna in oncology: gearing up for clinic. *Annals of Laboratory Medicine*, 38(1):1–8, 2018.

[71] The Catalogue of Somatic Mutations in Cancer (COSMIC) Website. `https://cancer.sanger.ac.uk/cosmic/download`. Accessed: 2018-06-16.

[72] Julian PT Higgins, Simon G Thompson, and David J Spiegelhalter. A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 172(1):137–159, 2009.

[73] Huann-Sheng Chen, Ruth M Pfeiffer, and Shunpu Zhang. A powerful method for combining p-values in genomic studies. *Genetic Epidemiology*, 37(8):814–819, 2013.

[74] Frank Dudbridge and Bobby PC Koeleman. Rank truncated product of p-values, with application to genomewide association scans. *Genetic Epidemiology*, 25(4):360–366, 2003.

[75] Samsiddhi Bhattacharjee, Preetha Rajaraman, Kevin B Jacobs, William A Wheeler, Beatrice S Melin, Patricia Hartge, Meredith Yeager, Charles C Chung, Stephen J Chanock, Nilanjan Chatterjee, et al. A subset-based approach improves power and interpretation for the combined analysis of genetic association studies of heterogeneous traits. *The American Journal of Human Genetics*, 90(5):821–835, 2012.

[76] Kai Yu, Qizhai Li, Andrew W Bergen, Ruth M Pfeiffer, Philip S Rosenberg, Neil Caporaso, Peter Kraft, and Nilanjan Chatterjee. Pathway analysis by adaptive combination of p-values. *Genetic Epidemiology*, 33(8):700–709, 2009.

[77] Jelle J Goeman, Sara A Van De Geer, Floor De Kort, and Hans C Van Houwelingen. A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics*, 20(1):93–99, 2004.

[78] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[79] Yu-Chen Su, William James Gauderman, Kiros Berhane, and Juan Pablo Lewinger. Adaptive set-based methods for association testing. *Genetic Epidemiology*, 40(2):113–122, 2016.

[80] Bogdan Pasaniuc and Alkes L Price. Dissecting the genetics of complex traits using summary association statistics. *Nature Reviews Genetics*, 18(2):117, 2017.

[81] Benjamin M Neale and Pak C Sham. The future of association studies: gene-based analysis and replication. *The American Journal of Human Genetics*, 75(3):353–362, 2004.

[82] Jimmy Z Liu, Allan F Mcrae, Dale R Nyholt, Sarah E Medland, Naomi R Wray, Kevin M Brown, Nicholas K Hayward, Grant W Montgomery, Peter M Visscher, Nicholas G Martin, et al. A versatile gene-based test for genome-wide association studies. *The American Journal of Human Genetics*, 87(1):139–145, 2010.

[83] Miao-Xin Li, Hong-Sheng Gui, Johnny SH Kwan, and Pak C Sham. Gates: a rapid and powerful gene-based association test using extended simes procedure. *The American Journal of Human Genetics*, 88(3):283–293, 2011.

[84] Il-Youp Kwak and Wei Pan. Gene-and pathway-based association tests for multiple traits with gwas summary statistics. *Bioinformatics*, 33(1):64–71, 2016.

[85] Yangqing Deng and Wei Pan. Testing genetic pleiotropy with gwas summary statistics for marginal and conditional analyses. *Genetics*, 207(4):1285–1299, 2017.

[86] Daniel J Schaid, Xingwei Tong, Beth Larrabee, Richard B Kennedy, Gregory A Poland, and Jason P Sinnwell. Statistical methods for testing genetic pleiotropy. *Genetics*, 204(2):483–497, 2016.

[87] William Poole, David L Gibbs, Ilya Shmulevich, Brady Bernard, and Theo A Knijnenburg. Combining dependent p-values with an empirical adaptation of browns method. *Bioinformatics*, 32(17):i430–i436, 2016.

[88] Qizhai Li, Jiyuan Hu, Juan Ding, and Gang Zheng. Fisher's method of combining dependent statistics using generalizations of the gamma distribution with applications to genetic pleiotropic associations. *Biostatistics*, 15(2):284–295, 2014.

[89] Ronald Aylmer Fisher. Statistical methods for research workers. In *Breakthroughs in Statistics*, pages 66–70. Springer, 1992.

[90] Haikady N Nagaraja. Order statistics from independent exponential random variables and the sum of the top order statistics. In *Advances in Distribution Theory, Order Statistics, and Inference*, pages 173–185. Springer, 2006.

[91] Shunpu Zhang, Huann-Sheng Chen, and Ruth M Pfeiffer. A combined p-value test for multiple hypothesis testing. *Journal of Statistical Planning and Inference*, 143(4):764–770, 2013.

[92] Gary A Churchill and Rebecca W Doerge. Empirical threshold values for quantitative trait mapping. *Genetics*, 138(3):963–971, 1994.

[93] Karen N. Conneely and Michael Boehnke. So many correlated tests, so little time! Rapid adjustment of P values for multiple correlated tests. *American Journal of Human Genetics*, 81(6):1158–1168, December 2007.

[94] Miao-Xin Li, Juilian MY Yeung, Stacey S Cherny, and Pak C Sham. Evaluating the effective numbers of independent tests and significant p-value thresholds in commercial genotyping arrays and public imputation reference datasets. *Human genetics*, 131(5):747–756, 2012.

[95] J Li and L Ji. Adjusting multiple testing in multilocus analyses using the eigenvalues of a correlation matrix. *Heredity*, 95(3):221–227, 2005.

[96] James M Cheverud. A simple correction for multiple comparisons in interval mapping genome scans. *Heredity*, 87(1):52–58, 2001.

[97] Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.

[98] David Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110, 1947.

[99] Andrei Nikolaevich Kolmogorov. Unbiased estimates. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 14(4):303–326, 1950.

[100] C Radhakrishna Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull Calcutta. Math. Soc.*, 37:81–91, 1945.

[101] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT Press Cambridge, 2012.

[102] David Soave, Harriet Corvol, Naim Panjwani, Jiafen Gong, Weili Li, Pierre-Yves Boëlle, Peter R Durie, Andrew D Paterson, Johanna M Rommens, Lisa J Strug, et al. A joint location-scale test improves power to detect associated snps, gene sets, and pathways. *The American Journal of Human Genetics*, 97(1):125–138, 2015.

[103] Lei Sun, Johanna M Rommens, Harriet Corvol, Weili Li, Xin Li, Theodore A Chiang, Fan Lin, Ruslan Dorfman, Pierre-François Busson, Rashmi V Parekh, et al. Multiple apical plasma membrane constituents are associated with susceptibility to meconium ileus in individuals with cystic fibrosis. *Nature Genetics*, 44(5):562–569, 2012.

[104] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, 2015.

[105] Rapid gwas of thousands of phenotypes for 337,000 samples in the uk biobank. `http://www.nealelab.is/blog///2017/7/19/rapid-gwas-of-thousands-of-phenotypes-for-337000-samples-in-the//-uk-biobank`. Accessed: 2018-03-15.

[106] Howard Levene. *Contributions to probability and statistics. Essays in Honor of Harold Hotelling*. Stanford University Press California, 1960.

[107] João Fadista, Alisa K Manning, Jose C Florez, and Leif Groop. The (in) famous gwas p-value threshold revisited and updated for low-frequency variants. *European Journal of Human Genetics*, 24(8):1202, 2016.

[108] Montgomery Slatkin. Linkage disequilibrium, understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9(6):477, 2008.

[109] The Broad Institute Website. `https://data.broadinstitute.org/alkesgroup/LDSCORE/eur_w_ld_chr.tar.bz2`. Accessed: 2018-11-15.

[110] Youngchao Ge, Sandrine Dudoit, and Terence P Speed. Resampling-based multiple testing for microarray data analysis. *Test*, 12(1):1–77, 2003.

[111] Jian Yang, Michael N Weedon, Shaun Purcell, Guillaume Lettre, Karol Estrada, Cristen J Willer, Albert V Smith, Erik Ingelsson, Jeffrey R O'connell, Massimo Mangino, et al. Genomic inflation factors under polygenic inheritance. *European Journal of Human Genetics*, 19(7):807, 2011.

[112] R Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977.

VITA

Emery T. Goossens recieved his Bachelor of Arts degree from New York University in 2010, where he majored in Economics. Emery attended the University of Toronto in 2011 as a non-degree student, taking Mathematics and Statistics courses. He enrolled in the Masters of Science in Statistics program at the University of Toronto in 2013 and received his degree in 2014. He enrolled in the Ph.D program at Purdue University in 2014 and received a Masters of Mathematical Statistics in 2018. After his Ph.D, Emery intends to pursue a career in wrangling data and training models in the biotechnology industry.