Robust and Computationally Efficient Methods for High-Throughput Drug Screening Studies

by

Zoe L. Rehnberg

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2022

Doctoral Committee:

Assistant Professor Johann Gagnon-Bartsch, Chair
Associate Professor Arvind Rao
Professor Kerby Shedden
Assistant Professor Jonathan Terhorst

Zoe L. Rehnberg
zrehnber@umich.edu
ORCID iD: 0000-0001-6207-4660

ACKNOWLEDGMENTS

I would first like to express my immense gratitude to my advisor and mentor Johann Gagnon-Bartsch. Thank you for believing in me and supporting me unconditionally, especially when I did not believe in myself. I would not have completed this degree without you. In addition to being an outstanding research advisor, you have also shown me how to be a more effective, interested, and competent teacher. As an excellent educator, your presence in this department has helped propel me towards a teaching career of my own. Finally, I would like to thank Jillian, Anja, and Piper for sharing Johann with me – I know his time is valuable and I feel lucky to have gotten so much of it.

Next, I would like to thank my committee members Kerby Shedden, Jonathan Terhorst, and Arvind Rao. Insightful discussions with all three have contributed to this work. I am also grateful to Laura Heiser for helpful discussions, particularly about Chapter 2.

Thank you to all the members of the JGB research group: Greg Hunt, Julie Deeke, Robyn Ferg, Ed Wu, Nora Yujia Payne, Kristen Hunter, Charlotte Mann, and Juejue Wang. Your questions, contributions, and insights, not to mention your friendship, have been invaluable over the past six years.

I would also like to thank the undergraduate students who joined me on this research journey, both through the Big Data Summer Institute and the Undergraduate Research Program in Statistics, including Zhihao Guo, Nicole Kim, Benjamin Rappoport, Kathy Huo, Vivian Wong, Wenxin Guo, and Yuki Low. Your interest in and excitement for research has been energizing.

Financially, this research was supported in part by the National Science Foundation research training grant DMS-1646108.

Outside of my research, I would like to thank the applied reading group folks for keeping me excited about statistics over the past two years. Thank you especially to Nora, Rob, Drew, and Avery for the meaningful discussions and your friendship. Y'all remind me that being a statistician is cool.

Finally, I would like to thank the people who have kept me going for the past six years, the people who made me go on walks around campus when the sun was shining, the people who never said no to an after-work happy hour, the people who baked me goodies, the people who rode with me on the pedal pub, the people who listened to my doubts and told me it would be okay. I owe huge thanks to Laura Niss for being my rock, for showing me there is more to life than this program, and for being there every step of the way. You make me a more interesting person.

All my gratitude to Charlotte Mann for supporting me, celebrating me, and loving me at my best and at my worst. You have given me more than I can express.

To my parents and siblings: Thank you for believing in me since day one, for always taking my phone calls, and for never letting me back down from this challenge. You are my biggest cheerleaders and my biggest inspiration. I'm honored to bring one more Dr. Rehnberg, and more importantly, one more educator, into the family.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

ABSTRACT

There is an increasing emphasis on the utility of large-scale biological experiments to advance our understanding of human biological processes. The analysis of data from these studies, however, can face challenges associated with data size and complexity. For instance, two large pharmacogenomic databases, the Genomics of Drug Sensitivity in Cancer (GDSC) and the Cancer Cell Line Encyclopedia (CCLE), have been widely used to explore genetic predictors of drug sensitivity and to develop and study hypotheses about new anti-cancer therapies. At the same time, several papers have reported only moderate levels of agreement in drug sensitivity estimates between GDSC and CCLE with the discordance largely attributed to experimental and analytical factors, including differences in cell viability assay, range of tested drug concentrations, and construction of dose-response curves. There has been no published in-depth exploration, however, of the raw drug screening data from GDSC and CCLE. Therefore, we examine the raw data from both studies and identify technical variation such as complex spatial biases and batch-specific outliers. We show how these errors propagate through downstream calculations of relative viability and measures of drug sensitivity. Additionally, we note that technical error can interact with aspects of plate design such as the location of control wells along plate edges and the consistent orientation of drugged wells across replicates creating challenges for analysis. These findings highlight the importance of exploring the raw drug screening data prior to pursuing an analysis. They also inform a number of strategies for improving experimental design, such as randomized plate layouts.

To eliminate the effects of such between-plate variation in high-throughput drug screening studies, intensity measurements for treated wells are often normalized to the control wells. Such normalization allows for comparability across plates and across studies. However, within-plate variability, including spatial biases, cannot be alleviated by normalization to the controls. Therefore, we provide a normalization framework that addresses multiple types of spatial effects and can handle complex plate layouts. We carefully apply this normalization framework to the drug screening data from GDSC. Our normalization produces more reliable measures of drug sensitivity than current methods.

Finally, many existing methods for high-dimensional classification, including those used for pharmacogenomic data, require a substantial amount of computing time and power. Specifically, the use of cross-validation for tuning parameter selection and error estimation can be particularly

time-consuming. Therefore, we introduce an approximate leave-one-out cross-validation approach for principal component linear discriminant analysis that is computationally more efficient than existing methods. In particular, our method obviates the need to select tuning parameter values and optimizes computational efficiency through a series of matrix downdates. We apply our method to simulated data as well as to pharmacogenomic data from GDSC. For the type of genomic data for which this method is intended, it has comparable accuracy to existing approaches, while improving on computation time.

CHAPTER 1

Introduction

The analysis of data from biological experiments has long been a goal of statisticians. In recent years, however, many biological datasets have become increasingly large and incredibly diverse, and the questions scientists use those data to answer are more complex and precise than ever. In particular, the analysis of many modern biological datasets aims to push forward our understanding of human biological processes and improve precision medicine. An important source of data for improving precision cancer treatments, for instance, are large-scale cancer drug screening studies. The first step in developing new cancer treatments, drug screening studies involve testing a wide range of potential anti-cancer drugs on a diverse set of cancer cell lines, cancer cells that have been harvested from a tumor and grown in a laboratory for research purposes. The goal of such drug screening studies is to capture each drug's efficacy at inhibiting cancer cell growth.

The Genomics of Drug Sensitivity in Cancer (GDSC) project and the Cancer Cell Line Encyclopedia (CCLE) are two large-scale and publicly available cancer drug screening studies we focus on in this thesis (*Yang et al.*, 2013; *Barretina et al.*, 2012). These studies contain efficacy data for hundreds of potential anti-cancer drugs tested on over one thousand cancer cell lines at several concentrations each (see Appendix A for more details). These studies also provide detailed data about the cell lines themselves, including gene expression, methylation, and copy number variation levels. In the analysis of these data, it is common to combine estimated drug efficacy with cell line genomic information to identify genomic predictors of drug sensitivity.

One of the main benefits of these databases is their size; GDSC and CCLE contain a tremendous amount of information from many different sources. At the same time, their size and complexity complicate the analysis process. Throughout this dissertation, we consider and address several of the challenges that arise in the use of large-scale biological data. Before introducing some of those challenges, we note that in this chapter, and in this thesis, we largely focus our discussion on data from large-scale cancer drug screening studies. Many of our ideas and methods, however, can also be applied to a broad range of high-throughput biological experiments.

When datasets are large, they can suffer from particularly complex errors and widespread noise. While we expect the data from any experiment to contain errors, the errors are likely to be more

complicated and diverse in large and high-throughput experiments. In particular, both the amount of data collected and the length of time over which data collection takes place increases the likelihood that the type and structure of error will vary widely throughout the dataset. For example, error structure is likely to be different for data collected on the fifth day of the study and for data collected two and half years later. Additionally, we expect batch effects to be more widespread, corresponding, for instance, to different laboratories, different machines, and even different seasons of the year.

The manual identification of such errors that may be preferred for small datasets will no longer be feasible when the collected data span millions of rows (e.g., the raw GDSC datasets 1 and 2 and CCLE data have 3.7, 6.6, and 2.3 million rows, respectively). To address these issues, it is crucial to understand the types and frequencies of errors that are present in a given dataset (discussed in Chapter 2 for GDSC and CCLE). Eliminating the effects of these errors can be facilitated by smart experimental design choices and targeted data preprocessing techniques (discussed in Chapters 2 and 3, respectively).

Further, it is often desirable to combine multiple large biological datasets across studies for a unified analysis. This can be a way to combine different types of data (e.g., drug efficacy with gene expression levels) or a strategy for increasing the number of biological replicates. When combining data sources, however, potential differences in experimental procedures become important. For instance, GDSC and CCLE both measure drug efficacy, but they differ in the size of microplates on which the tests are performed, the assay for quantifying cell growth, and the number and concentration of tested drug doses, among many other factors (*Yang et al.*, 2013; *Barretina et al.*, 2012). Further, just as the types of errors can vary throughout a single dataset, there can also be vastly different error structures between datasets. Together, these differences make it difficult to meaningfully combine data or compare results across studies (addressed in Chapter 3).

Additionally, the unique features of each drug screening study can create challenges for the development of data processing and analysis methods that are broadly applicable. When each study uses its own experimental design that includes different types of control wells, plate layouts, and patterns of missing values, it may not be feasible to use the same processing and analysis techniques across all studies. At the same time, it is not practical to develop specific processing methods for each new experiment. Overall, the challenges of analyzing a single large biological study are exacerbated when combining the analysis of several.

Finally, the sheer size of many such datasets can affect the feasibility of a desired analysis. Particularly, many large-scale biological experiments produce high-dimensional data, with hundreds of thousands of features for each observation or sample in the study. In such a high-dimensional setting (n observations $\ll p$ features), standard statistical techniques will be ineffective. Instead, a proper analysis will require additional assumptions about the data structure or the use of regu-

larization. While there exist many techniques for handling such high-dimensional settings, they can be computationally expensive. Even beyond high-dimensionality, the computation time and memory needed to process and analyze a large dataset poses challenges. Some procedures will be impossible to perform on a personal computer, while others will simply take an impractical amount of time to complete. In Chapter 4, we discuss the computational difficulties of modern data-analysis techniques for large-scale data. We show the long computation times of traditional cross-validated classification methods and introduce a new, and faster, algorithm.

The rest of this thesis is organized as follows. In Chapter 2, we do a deep dive into the raw and processed drug screening data in the GDSC and CCLE databases. Motivated by extensive literature on the disagreement between the two studies and our own findings of disagreement within each study, we identify several types of technical variation in the raw data and highlight how these errors propagate through downstream calculations. Additionally, we note that technical error can interact with aspects of plate design, such as the location of control wells and the consistent orientation of drugged wells across replicates, creating challenges for analysis. These findings highlight the importance of exploring the raw drug screening data prior to pursuing an analysis and inform a number of strategies for improving experimental design, such as randomized plate layouts.

In Chapter 3, we introduce a new framework for normalizing the raw data from large biological studies. As previously described, such data can suffer from substantial non-biological variation both within and across assays and studies. Many existing normalization strategies are only able to address one type of technical variation. Others are not effective in settings with complex experimental designs. Therefore, we provide a normalization framework that addresses multiple types of technical variation and can handle complex data settings. We carefully apply this normalization framework to the drug screening data from GDSC, addressing many of the errors outlined in Chapter 2. Our normalization produces more reliable measures of drug sensitivity than current methods.

In Chapter 4, we introduce a fast and approximate version of leave-one-out cross-validation for high-dimensional linear discriminant analysis (LDA). While LDA is a common and simple linear classifier, adapting it to high dimensions can be extremely computationally intensive. Our approach to high-dimensional LDA combines dimensionality reduction via principal components with covariance matrix regularization in such a way that no tuning parameter selection is needed. Additionally, we introduce approximations in the LOO CV fitting procedure, implement quick downdating for large matrix calculations, and take advantage of the data structure to avoid redundant calculations. We combine these techniques in the nPC-LDA with FAST-CV algorithm. On the type of genomic data for which it was developed, nPC-LDA with FAST-CV is a cross-validated classifier that performs substantially faster than, but with comparable accuracy to, existing cross-validated methods.

Finally, in Chapter 5, we conclude and discuss future possibilities for this work. This includes methods for flagging widespread technical errors in drug screening data and connecting the design of drug screening experiments to principled data processing methods.

CHAPTER 2

Technical Variation in Drug Screening Studies

2.1 Introduction

The Genomics of Drug Sensitivity in Cancer (GDSC) and the Cancer Cell Line Encyclopedia (CCLE) are two large-scale pharmacogenomic studies containing a wide range of genetic and pharmacological data (*Yang et al.*, 2013; *Barretina et al.*, 2012). These publicly available databases have been widely used to explore genetic predictors of drug sensitivity and to accelerate the discovery of novel anti-cancer therapies (*Weinstein and Lorenzi*, 2013; *Genomics of Drug Sensitivity in Cancer*). In the years since their publication, however, GDSC and CCLE have been at the center of a broad discussion about the concordance of pharmacogenomic data across studies (*e.g.*, *Haibe-Kains et al.* (2013); *The Cancer Cell Line Encyclopedia and Genomics of Drug Sensitivity in Cancer Investigators* (2015); *Safikhani et al.* (2016a); *Bouhaddou et al.* (2016); *Geeleher et al.* (2016); *Mpindi et al.* (2016); *Safikhani et al.* (2016b)). While the gene expression data from GDSC and CCLE has been found to be highly concordant, several papers have reported only moderate levels of agreement for the drug screening data between the two studies. These previous publications use a wide range of analysis techniques, but focus almost exclusively on analyzing summarized drug sensitivity measures such as the drug concentration at which 50% of cell growth is inhibited (IC_{50}) and the area under the dose-response curve (AUC).

The primary contribution of this chapter is a thorough investigation of the raw intensity data from GDSC and CCLE. We identify systematic and consequential technical error in the raw data from both studies. Notably, the same types of error, including spatial effects, checkerboard pattern, batch-specific outliers, and noise, are present in both GDSC and CCLE. This technical variation is likely an important factor in the previous reports of inconsistency in drug sensitivity measures. Our findings highlight the importance of exploring the raw data before beginning an analysis, and we provide a Shiny app to facilitate such an exploration of the GDSC and CCLE data.

This chapter also demonstrates the ways in which technical error can interact with aspects of the experimental design, including plate layout and the location of control wells. Such interactions

can cause systematic errors in sensitivity estimates with implications for downstream analysis. We discuss small changes to design, including employing randomization and multiple forms of replication, that can mitigate the effects of these errors.

2.2 Previous Comparisons of GDSC and CCLE

In a comparison analysis of GDSC and CCLE, *Haibe-Kains et al.* (2013) found high concordance for gene expression data, both within GDSC (median Spearman rank correlation of 0.97) and between GDSC and CCLE (median correlation of 0.85). A comparison of drug sensitivity, as measured by IC_{50} and AUC, however, resulted in only moderate to poor concordance (median correlation of 0.28 for IC_{50} and 0.35 for AUC). The authors found similarly low consistency after discretizing drug response to calls of "sensitive", "intermediate", and "resistant" for each cell line, and also after removing insensitive cell lines. Further, they showed that discrepancies in drug sensitivity data, rather than in genomic data, drove inconsistencies in significant genetic predictors of drug sensitivity selected in GDSC and CCLE.

A series of follow-up studies attempted to find both better consistency between the two studies and explanations for the purported differences (*The Cancer Cell Line Encyclopedia and Genomics of Drug Sensitivity in Cancer Investigators*, 2015; *Haverty et al.*, 2016; *Pozdeyev et al.*, 2016; *Bouhaddou et al.*, 2016; *Geeleher et al.*, 2016; *Mpindi et al.*, 2016; *Safikhani et al.*, 2016b; *Rahman et al.*, 2018; *Hu et al.*). Somewhat better agreement was found by using different measures of concordance, different measures of drug sensitivity, and different comparison techniques.

In a combined analysis, *The Cancer Cell Line Encyclopedia and Genomics of Drug Sensitivity in Cancer Investigators* (2015) noted that high correlation should not be expected when most cell lines are insensitive to the tested drug, as was true for many compounds in these studies; in such situations, noise is the dominant effect. To alleviate this issue, the authors measured consistency with Pearson correlation and truncated IC_{50} estimates for insensitive cell lines to the maximum tested drug concentration. Using these methods, they found improved concordance (median correlation of 0.54 for IC_{50} and 0.45 for AUC). In their analysis, however, up to 98% of IC_{50} estimates were truncated for a given drug, leading to concerns that the reported correlations were overestimated (*Pozdeyev et al.*, 2016).

Several studies accounted for differences in the range of tested drug concentrations between GDSC and CCLE. Somewhat better consistency was found for both AUC (*Pozdeyev et al.*, 2016) and IC_{50} (*Rahman et al.*, 2018) when adjusting by the range of tested concentrations. *Bouhaddou et al.* (2016) tried to find improved consistency by fitting dose-response curves with only the overlapping doses between GDSC and CCLE. When considering IC_{50} estimates for sensitive drug-cell line combinations, however, the authors found Pearson correlation greater than 0.5 for only two of

the six drugs with sufficient cell lines.

Revisiting their earlier analysis (in *Haibe-Kains et al.* (2013)) with updated data, *Safikhani et al.* (2016b) chose the best combination of sensitivity measure and consistency metric for each individual drug. They found moderate to good concordance for five drugs, but a lack of concordance or an insufficient number of sensitive cell lines for the remaining ten drugs. Further, whether comparing between cell lines or across cell lines (*Geeleher et al.*, 2016; *Mpindi et al.*, 2016), gene expression data remained more consistent than drug sensitivity data, confirming their previous findings.

It has been suggested that the general discordance in drug sensitivity can largely be attributed to experimental and analytical factors. Such factors include differences in cell viability assay, management and delivery of compounds, cell culture conditions (seeding density, culture media, etc.), range of tested drug concentrations, and construction of dose-response curves (*Hatzis et al.*, 2014; *Haverty et al.*, 2016; *Ding et al.*, 2017; *Larsson et al.*, 2020). *Wang et al.* (2020) also recognize the potential presence of variability in raw drug screening data that can lead to unacknowledged uncertainty in summaries of drug sensitivity.

2.3 Overview of the Data

GDSC The version 17 release of the GDSC drug sensitivity data contains 1,057 cancer cell lines and 265 anti-cancer drugs scanned on 15,631 plates (*Yang et al.*, 2013). While additional GDSC data has since been released, version 17 contains the data we have most thoroughly examined and will focus on in this chapter. Each plate, scanned at either Massachusetts General Hospital (MGH) or Wellcome Trust Sanger Institute (WTSI), was plated with one cell line, but multiple compounds. Each compound was tested over a 256-fold concentration range, at either 9 doses with two-fold dilution or 5 doses with four-fold dilution. Between the two sites, there were 125 different plate layouts, with different plate sizes (384- or 96-well), compounds, and number and location of control wells (Figure 2.1). All plates included both untreated control wells (containing cells, but no compound) and blank control wells (containing no cells and no compound). Every plate also had wells with missing intensity measurements; most missing values were due to the use of propriety compounds for which no data was publicly released, while others were due to quality control failures. Two different assays were used to capture cell viability, SYTO 60 and Resazurin. These were constant across a plate.

While most drug-cell line combinations appear in the GDSC database at most once, four drugs (AZD6482, refametinib, PLX-4720, and pictilisib) were replicated over the same range of concentrations for more than 700 cell lines. These replicates consist of 3,232 drug-cell line combinations scanned on 4,161 different plates. Three of these drugs were tested twice at WTSI, while AZD6482 was repeated between the two sites. Replicate assays for AZD6482 typically took place within two

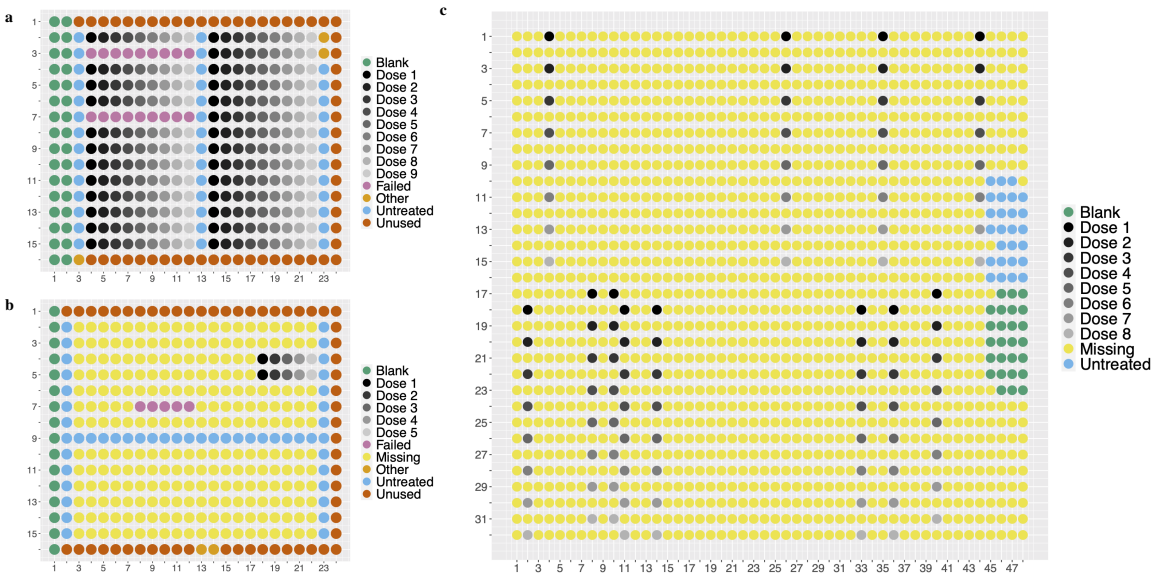

Figure 2.1: Layout of GDSC and CCLE plates. Example plate layouts for (**a**) a GDSC plate with drugs applied at 9 concentrations, (**b**) a GDSC plate with drugs applied at 5 concentrations, and (**c**) a CCLE plate.

years, while replicates within WTSI tended to be either about one year or three years apart.

CCLE The raw CCLE drug screening data contains information about 613 cancer cell lines and 27 compounds scanned on 14,187 plates (*Barretina et al.*, 2012, 2019). Each compound was tested over 8 concentrations with 3.16-fold dilution and a maximum concentration of 8 μM. Replicates are available for almost every drug-cell line combination (median 4, minimum 1, maximum 10), but like GDSC, all replicates are on separate plates, i.e., there are no within-plate replicates. The screening was done on 1,536-well plates, though intensities for the vast majority of wells were not released; the most complete plates have intensities for fewer than 10% of wells (Figure 2.1; Appendix B.1). Every plate contains both untreated and blank control wells. Cell viability was captured using Cell Titer Glo, a method that uses ATP as an indicator of viable cells.

Within-Study Replication Existing literature has investigated the concordance of drug sensitivity measures for drug-cell line combinations replicated between GDSC and CCLE. Both studies, however, have internal replication, and we used these repeated measurements to evaluate AUC agreement within each study. For GDSC, we focused on the four replicated drugs described above. For CCLE, we considered all 27 drugs; when there were more than two replicates for a given drug-cell line combination, we randomly selected two to consider.

We found varying levels of consistency across cell lines for each drug. Similar patterns were apparent for both GDSC (Figure 2.2; Table B.1) and CCLE (Figure B.1). Median Pearson correlation for CCLE replicates is 0.65, with a range from 0.34 to 0.85. As expected, correlation tends to

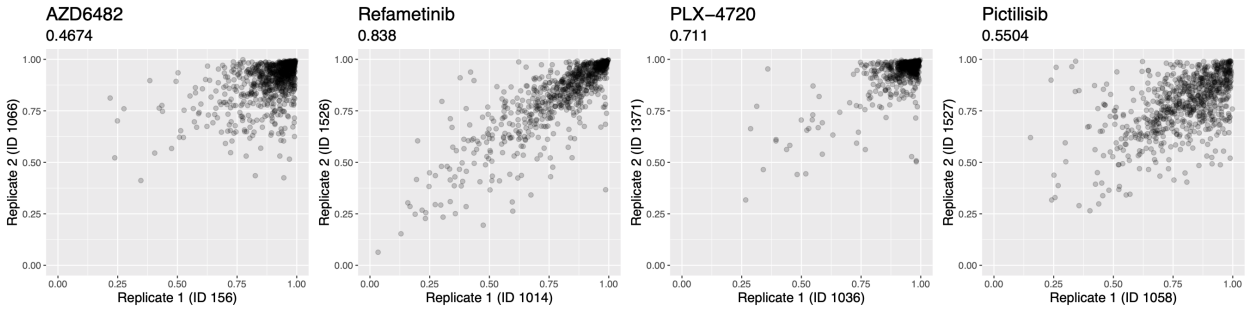

Figure 2.2: Within-study agreement for GDSC drugs. AUC estimates for the four drugs replicated within GDSC. Each point is one cell line (n = 829, 801, 844, and 758, respectively). Pearson correlation is provided at the top of each plot.

be higher for more broadly effective compounds (median correlation of 0.64 for narrowly effective and 0.78 for broadly effective CCLE drugs; Appendix B.2). Previous work suggests that noise is an important factor when low consistency is observed for drug sensitivity measures, especially for narrowly effective compounds (*The Cancer Cell Line Encyclopedia and Genomics of Drug Sensitivity in Cancer Investigators*, 2015). This chapter shows that, in addition to noise, other types of technical variation are also affecting these levels of concordance.

2.4 Technical Variation

Many GDSC and CCLE plates are high-quality. Figure 2.3a-b shows a GDSC scan with no apparent spatial effects or outliers, but with several drugs that are effective at high concentrations. This is the biology of interest, and it is clearly apparent. Many other plates in these studies, however, have noticeable errors. We identified four types of technical error that we believe are contributing to the lack of concordance in drug sensitivity. These errors appeared in both the GDSC and CCLE data and are outlined below.

Spatial Effects Many plates show systematic spatial bias. Figure 2.3c shows a plate-wide diagonal gradient with higher intensities in the upper left fading to lower intensities in the bottom right. Frequently, spatial bias appears as this type of systematic and gradual change in well intensity over a whole plate. We quantified the extent to which such spatial effects exist on GDSC plates and found that more than half of all plates scanned at WTSI have horizontal spatial effects with a magnitude of at least 0.1 $\log_2$ units (Appendix B.3). This causes the viability in untreated control wells, for instance, to vary by more than 10% across the plate (note: $2^{0.1} \approx 1.1$). As we discuss below, this type of spatial bias can have a large impact on relative viability and drug sensitivity estimation.

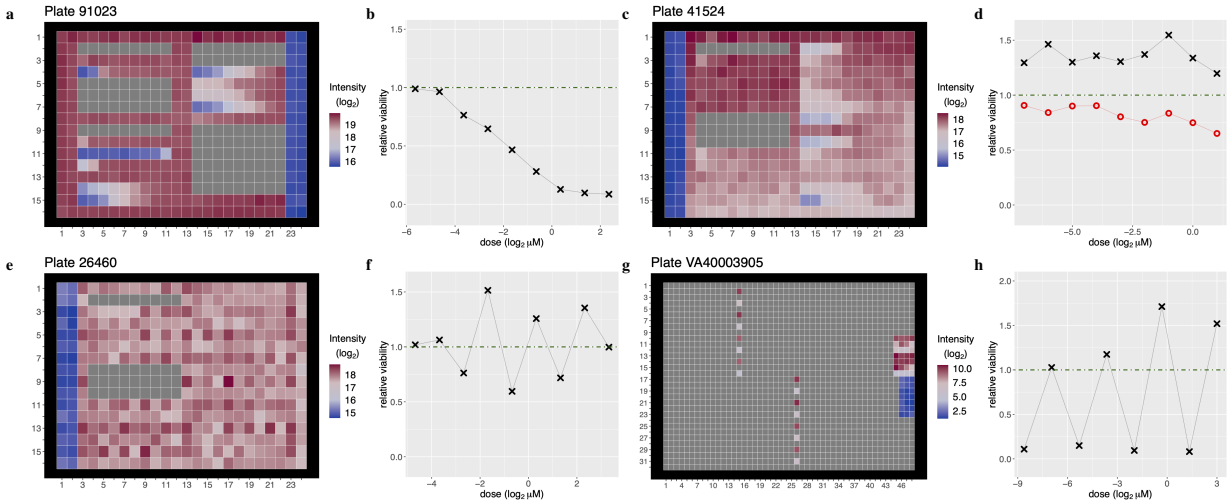

Figure 2.3: Technical error in raw GDSC and CCLE data. (**a**) A high-quality GDSC plate with no spatial effects, but with several effective drugs. Each cell displays well intensity on the $\log_2$ scale. Grey cells indicate missing intensities. (**b**) Dose-response curve for an effective drug on GDSC plate 91023 (plot a; row 15, columns 3–11). Relative viabilities are normalized to the median of the untreated controls. (**c**) A GDSC plate with systematically higher intensities in the upper left and lower intensities in the lower right. The magnitude of horizontal spatial effects is approximately 0.4 $\log_2$ units. The vertical spatial effects appear to be larger, but are more difficult to quantify (Appendix B.3). (**d**) Two dose-response curves from GDSC plate 41524 (plot c). The drug in row 4 (columns 4–12) is in black; the drug in row 12 (columns 4–12) is in red. Neither drug appears particularly effective, but the spatial effects create a dramatic shift in relative viabilities. (**e**) A GDSC plate with a checkerboard pattern. (**f**) Dose-response curve for row 15 (columns 4–12) of GDSC plate 26460 (plot e). (**g**) A CCLE plate with a checkerboard pattern. (**h**) Dose-response curve for column 26 of CCLE plate VA40003905 (plot g).

In CCLE, the sparsity of the released data prevents similar visualization of plate-wide gradients. Further, the untreated control wells are always located on the right edge of the plate, making it difficult to use them to identify spatial effects (Figure 2.1). Nonetheless, evidence of spatial bias is apparent when examining wells treated with the lowest drug doses. We would expect intensities in these wells to be no greater than intensities in the untreated control wells. However, the intensity in the lowest dose drugged well is at least 10% higher than the median untreated control intensity for 32% of all CCLE drug-cell line combinations.

Checkerboard Pattern Another form of systematic variation is a checkerboard pattern, which is characterized by alternating wells of high and low intensity (Figure 2.3e-h). When a strong checkerboard pattern is present, the majority of wells are surrounded by wells with higher (or lower) intensities, regardless of drug concentration. We find that about 10% of GDSC plates contain a substantial amount of checkerboard pattern (Appendix B.4; Figure B.3). Observations for any given CCLE plate are too sparse to similarly calculate prevalence.

Batch-Specific Outliers We found groups of plates containing local artifacts that are repeated across many scans. For example, almost all GDSC plates scanned on November 9, 2011 containing AZD6482 (drug 1066) have large outliers in columns 5 and 6 of row 7 (Figure B.4). Other examples are discussed in Appendix B.5.

Noise Many plates exhibit strong and seemingly random variability that is neither spatial nor checkerboard (Figure B.4).

2.5 Implications for Downstream Analysis

The technical errors described above can impact AUC estimation and contribute to the reported inconsistency in drug sensitivity. These errors interact with each other, with plate layouts, and with other aspects of the experimental design to create challenges for downstream analytical methods.

2.5.1 Impact of Technical Variation

When analyzing drug screening data, the raw intensities obtained through the screening process must be normalized into relative viabilities. We ideally want to normalize each treated well by the intensity that would have been observed if that well had not been treated. Therefore, a common approach to obtain relative viabilities is to divide raw drugged intensities by the median intensity of the untreated control wells. This is the approach used in the CCLE study and in this chapter (see Chapter 3 for a more in-depth discussion of relative viability normalization). Notably, the normalizing factor used in this approach (the median intensity of the untreated control wells) is the same for all wells on a plate; there is no allowance for spatial variation. In actuality, however, plates are affected by spatial bias and checkerboard pattern, causing the appropriate normalizing factor to vary across a plate. Therefore, the median of the untreated control wells is not sufficient, and spatial bias can cause shifts in relative viabilities and disagreement between replicates (Figure 2.4a-b).

Such inaccuracy in relative viability estimation further affects drug sensitivity measures, like AUC (Appendix B.6). Consider, for example, a plate containing spatial effects on the order of 0.25 $\log_2$ units (14.5% of GDSC plates scanned at WTSI have spatial effects larger than 0.25; Appendix B.3). On this plate, consider a drug with no effect such that all relative viabilities should equal 1. If the drug is tested in a high intensity region of the plate compared to the untreated controls, however, the spatial effects might cause all relative viabilities to equal 1.2 (note: $2^{0.25} \approx 1.2$). The AUC would be 1.2, and the drug would accurately be labeled ineffective.

Alternatively, consider a drug tested in a low intensity region of the plate such that the spatial

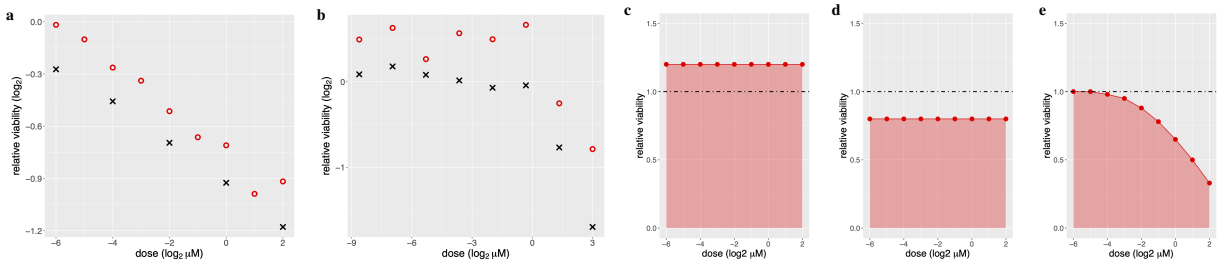

Figure 2.4: Spatial effects in raw data shift dose-response curves. (**a**) Dose-response curves for two screens of pictilisib on cell line NKM-1 in GDSC and (**b**) dovitinib on cell line Hs 578T in CCLE. Black points correspond to replicate 1 and red points correspond to replicate 2. The shape of the dose-response curves is similar across replicates, but the red observations are shifted above the black observations, likely a result of spatial bias. Relative viabilities are plotted on the $\log_2$ scale to highlight that differences in relative viability are proportionally consistent across doses. (**c**) Hypothetical dose-response curve for an ineffective drug where spatial effects cause the relative viabilities to be high. The AUC is 1.2. (**d**) Dose-response curve for an ineffective drug where spatial effects cause the relative viabilities to be low. The AUC is 0.8. (**e**) Dose-response curve for an effective drug. The AUC is 0.8.

effects cause all relative viabilities to equal 0.8, with an AUC of 0.8. Without context, this AUC could reasonably indicate an effective drug (Figure 2.4c-e), suggesting that spatial effects in the raw data can impact the interpretability of downstream drug sensitivity measures.

Many commonly used plate layouts present challenges when trying to analytically adjust for this type of spatial bias (Figure 2.1). Untreated control wells are often placed around the edges of a plate or in a single block, which prevents them from fully capturing technical variation, particularly spatial effects. Additionally, for each drug, consecutive doses are applied to consecutive wells, making it difficult to deconfound spatial gradients and biology. Together, these factors complicate the process of understanding and correcting spatial bias. There are similar challenges with checkerboard pattern, extreme outliers, and random noise, making it difficult to accurately estimate the appropriate normalizing factor and calculate relative viabilities when these errors compound.

The compounding challenges of technical variation and plate design do not only produce unreliable AUC estimates for individual drug-cell line combinations, but can also produce errors that are systematic in nature. In our analysis, we noticed several GDSC and CCLE drugs producing consistent responses across cell lines: in addition to drugs with broad and tissue-specific effects, there were several compounds that appeared to be widely promoting cell growth on hundreds of plates (Figure 2.5a). Further investigation into these apparent growth promoting drugs, however, indicated that technical effects related to plate layouts, not biology, is causing this phenomenon. The CCLE drug L-685458 is a particularly interesting example that highlights the compounding challenges of technical error and plate design (arrow in Figure 2.5a).

Drug L-685458 was tested 2,004 times on 607 different cell lines in the CCLE study. Of these tests, 1,497 (75%) have an AUC larger than 1, suggesting L-685458 promotes growth. If in fact

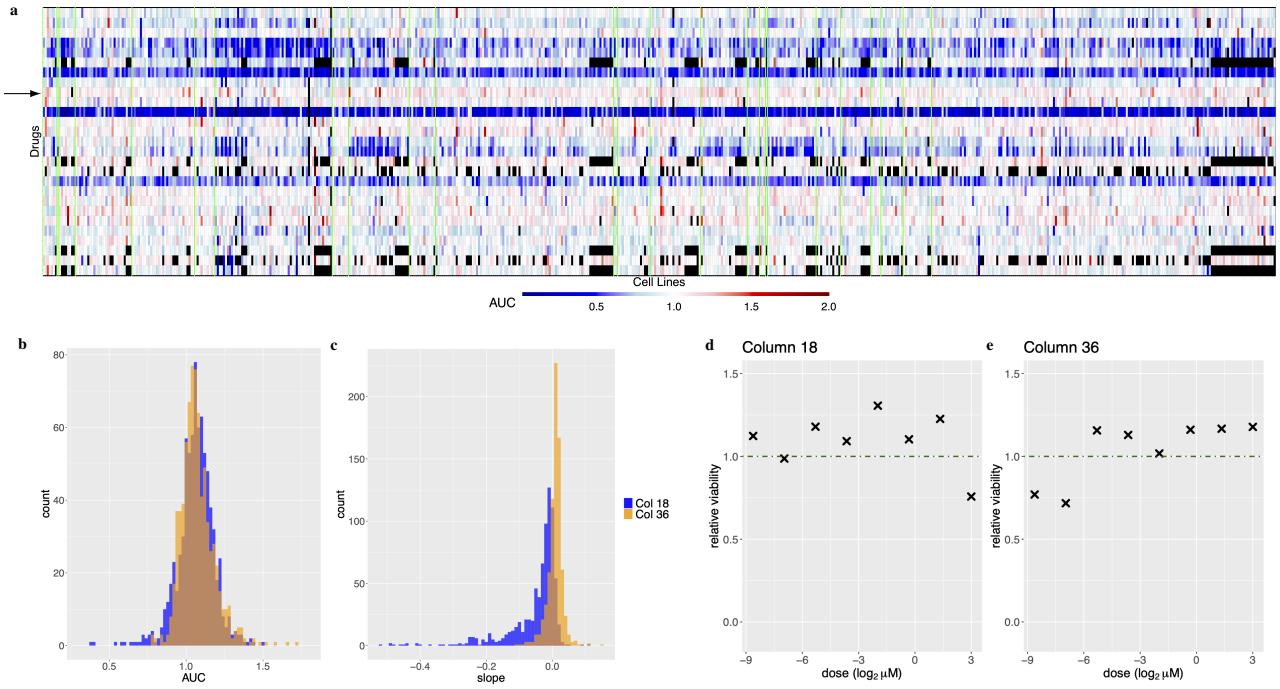

Figure 2.5: Effects of technical variation and plate design on drug L-685458. (**a**) AUC values for all CCLE drug-cell line combinations (one value of each replicate was randomly chosen). Black cells indicate drug-cell line combinations that were not tested. The arrow points to CCLE drug L-685458. See Figure B.5 for a more detailed plot. (**b**) AUC estimates for tests done in column 18 and in column 36 for L-685458. The distributions are similar. (**c**) Slopes for tests done in column 18 and in column 36 for L-685458. Column 18 has largely negative slopes while column 36 has largely positive slopes. The same cell lines were tested in each column. (**d**) Dose response plot for L-685458 tested on cell line RKO in column 18 and (**e**) column 36. The AUC values are similar across replicates, but the column 18 slope is negative while the column 36 slope is positive.

this drug promotes growth, we might also expect to see dose-response curves that are increasing. Therefore, we fit a simple linear regression for each cell line, regressing $\log_2$ intensity on $\log_2$ drug dose and used slope as an indicator of dose-response relationship. Overall, the slopes are not suggestive of broad growth promotion (mean: -0.0213; median: -0.00232; Figure B.6; Appendix B.7).

Further, each plate contained drug L-685458 in either column 18 or 36, and we identified substantial differences in dose-response results between the columns. For the 380 cell lines that were tested at least once in both column 18 and 36, we found almost identical distributions of AUC values between the columns, but very different distributions of slope (Figure 2.5b-c). The majority of tests done in column 18 have slopes less than 0, while the majority of tests done in column 36 have slopes greater than 0. Further, for a given cell line, the slopes for tests done in column 18 are systematically lower than the slopes for tests done in column 36 (Figure B.6).

Thus, the large AUC values for drug L-685458 do not necessarily indicate growth promotion.

There are hundreds of cell lines with an AUC greater than 1 and a slope less than 0. Further, the differences between column 18 and column 36 indicate clear technical effects that may be explained by the layout of CCLE plates. Drugs tested in column 18 are tested on the top half of the plate (rows 2 through 16) and the highest doses are near the plate's top edge. On the other hand, drugs tested in column 36 are tested on the bottom half of the plate (rows 18 through 32) and the lowest doses are near the plate's bottom edge (Figure 2.1). Therefore, edge effects can cause the lowest doses in column 36 and the highest doses in column 18 to have artificially low intensities. Further, a checkerboard pattern can cause similar issues and exacerbate existing edge effects (Figure 2.5d-e). Therefore, we conclude that large AUC values for drugs like L-685458 are caused by the compounding of spatial effects, checkerboard pattern, and plate layout, not widespread growth promotion.

Similar interplay between technical error and plate design is likely also affecting drugs that appear to be widely inhibiting growth. Therefore, some of the biological signal we are most interested in could simply be a technical effect and a result of where a drug is consistently tested on the plate.

The importance of testing location can also be seen when examining the consistency of drug sensitivity measures across replicates. Within CCLE, we found AUC correlation to be quite high when both replicates were tested in the same location on different plates (median: 0.77); correlation was low when replicates were tested in different locations on different plates (median: 0.54; Figure 2.6; Appendix B.8; Figures B.7-B.8). This suggests strong systematic differences based on the location in which a drug-cell line combination is tested.

While there is good replicability for drug-cell line combinations tested in the same location, this does not suggest that using consistent plate layouts is the best strategy for producing high quality data. If each drug is always tested in the same location, then location-dependent spatial effects and technical artifacts will be confounded with biology, preventing sensitivity from being accurately

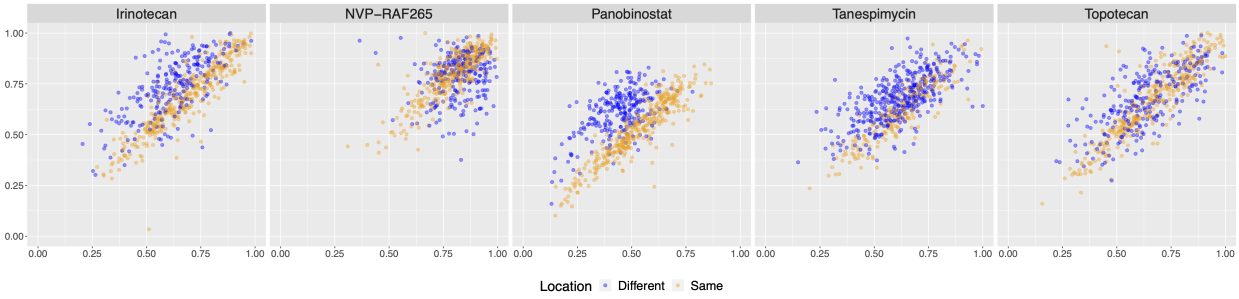

Figure 2.6: Testing location affects AUC agreement. Replicated AUC values for five CCLE drugs. Each point is one cell line (n = 603, 597, 597, 609, and 598, respectively). There is good agreement for replicates that were tested in the same location across plates.

compared across drugs. Randomization of plate layouts, in addition to replication, however, will allow this type of technical variation to be quantified and addressed, providing comparability across both drugs and cell lines.

2.5.2 Challenges for Analytical Methods

Many methods that are traditionally used to process and analyze drug screening data are not successful when plate layout, technical variation, and biology interact (Appendix B.9).

In particular, two common methods to address spatial bias, linear regression and loess regression, attempt to estimate the spatial effects on each plate and remove them. These techniques, however, do not work well for the GDSC and CCLE studies. Agreement between replicates does not improve after applying a linear regression spatial correction to the GDSC data (Figure B.9). Further, it is infeasible to widely apply a loess spatial correction due to the layout of many GDSC and all CCLE plates.

In both GDSC and CCLE, consecutive drug doses are placed in consecutive wells (every other well for CCLE). This design makes it difficult to differentiate between a gradient in drugged intensities caused by biology and a gradient caused by technical error. Trying to regress out spatial effects on a plate with several effective drugs can end up introducing spatial bias to that plate (Figure B.10).

Both GDSC and CCLE cap relative viabilities at 1 when estimating AUC values, a technique that attempts to mitigate errors, but can eliminate important biological information. Consider, for instance, a drug-cell line combination where the shape of the dose-response curve indicates the cell line is sensitive to the drug, but spatial effects have caused all relative viabilities to be larger than 1. Capping will eliminate the variability in the relative viabilities and incorrectly make the cell line look insensitive. As a result, capping relative viabilities often increases concordance between replicates, but the improved agreement may be largely artificial (Figure B.11).

Finally, fitting parametric dose-response curves is a common method for summarizing drug-cell line relationships. It is difficult, however, to accurately fit such a curve to drug screening data affected by the types of technical variation present in GDSC and CCLE, particularly strong checkerboard patterns (Figure B.12).

2.6 Non-Technical Variation

While we have focused on the technical error present in GDSC and CCLE, non-technical factors also appear to contribute to the discordance between replicates, as discussed in previous work (*Hatzis et al.*, 2014; *Haverty et al.*, 2016; *Ding et al.*, 2017; *Larsson et al.*, 2020). We found

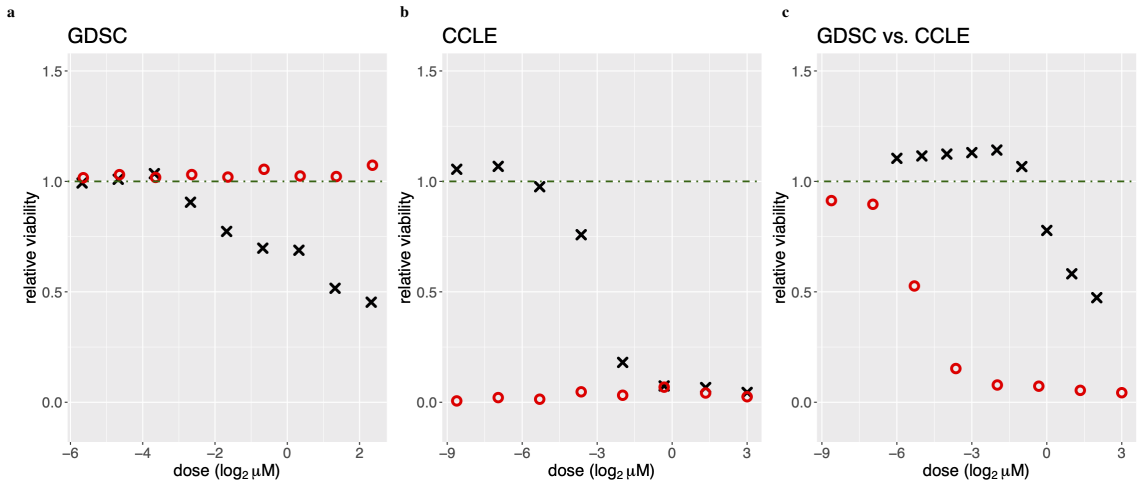

Figure 2.7: Disagreement caused by non-technical error. Dose-response curves for (**a**) two scans of drug AZD6482 on cell line Mo-T in GDSC (replicate 1 in black, replicate 2 in red), (**b**) two scans of irinotecan on cell line GIST882-F in CCLE, and (**c**) two scans of sorafenib on cell line NB-4, one in GDSC (black) and one in CCLE (red). For each plot, the dose-response relationship across replicates is different. These differences could indicate a difference in the biological response of the cells to the drug across the repeated measurements.

many drug-cell line combinations with fundamentally different dose-response relationships across replicates (Figure 2.7). These discrepancies seem to indicate a difference in the biological response of the cells to the drug across the repeated measurements. Differences in experimental factors like seeding density and culture media could be contributing to the discordance in these dose-response relationships. Alternatively, biological factors like genetic variation within cell lines have been shown to have a considerable impact on drug response metrics (*Ben-David et al.*, 2018).

2.7 Experimental Design

For both the GDSC and CCLE studies, we have seen that systematic technical variation, aspects of experimental design such as plate layout, and the interplay between them, impede the accurate estimation of drug sensitivity. Relatively small changes to experimental design and data sharing procedures, however, could go a long way towards mitigating the inevitable errors present in high-throughput drug screening data and making downstream statistical analysis more tractable (Figure 2.8a). Chief among these are randomizing plate layouts, employing multiple forms of replication, and releasing full data and documentation.

In particular, randomizing the entire plate layout, both control wells and drugged wells, will make it easier to handle systematic bias (*Niepel et al.*, 2019). With a randomized layout, the control wells are scattered across the plate, rather than placed along edges or in a single block, making them more useful and representative. Further, randomization reduces confounding between technical

variation and biology, allowing the unbiased estimation of each. For example, if the heatmap of a randomized plate displays a spatial gradient or a checkerboard pattern, a simple regression can address these spatial effects without inadvertently removing biological signal. In this way, randomization allows the use of general analytical methods.

Replication is another important aspect of experimental design, and it can take many forms. For each drug-cell line combination, replication can occur within a single plate, where either the entire dose range or just a few doses are repeated in different locations; alternatively, replicates can be tested across different plates. In any form, replication provides many distinct benefits for identifying, quantifying, and fixing errors. For instance, comparing replicates provides a straightforward way to perform quality control and identify batch-specific outliers, spatial bias and checkerboard pattern. Replication is also critical for identifying other sources of variability, such as the non-technical variation we found in both GDSC and CCLE (Figure 2.7). All types of replication also allow for quantifying error; measuring the variability between replicates enables estimation of uncertainty in raw intensity measurements and summary values such as AUC. Finally, replication can also help reduce and remove errors. Most famously, averaging replicates can reduce overall error; this is particularly true when replication is combined with randomization. Moreover, comparing replicates improves the ability to estimate and remove systematic variation such as spatial effects. In particular, within-plate replication of wells treated with high drug doses allows the estimation of spatial effects in low intensity wells. This is valuable as the magnitude of spatial variation may differ across drug dose, and the untreated control wells are only able to accurately estimate spatial effects in high intensity wells.

Notably, both randomization and replication allow for the comparison of intensities across location. Within-plate replication provides direct comparability for wells containing the same drug dose in two different locations. This improves the ability to differentiate between drug effects, spatial effects, and noise. Alternatively, randomization provides comparability between locations when averaging across many scans. In general, comparability across location translates to comparability across drugs, which is particularly important for identifying the most effective drug for a given cell line.

Finally, combining both randomization and replication on the same plate allows for the quantification of error, including obtaining more accurate standard error estimates and performing principled statistical inference. In particular, the use of both randomization and replication helps to justify the assumptions behind methods like analysis of variance (ANOVA). When both within-plate and between-plate replication are present, such methods can be used to quantify the contributions of within-plate variability, between-plate variability, and inherent biological variability to overall error. In general, designing plates with both randomization and replication will allow for a wide range of analyses to be performed.

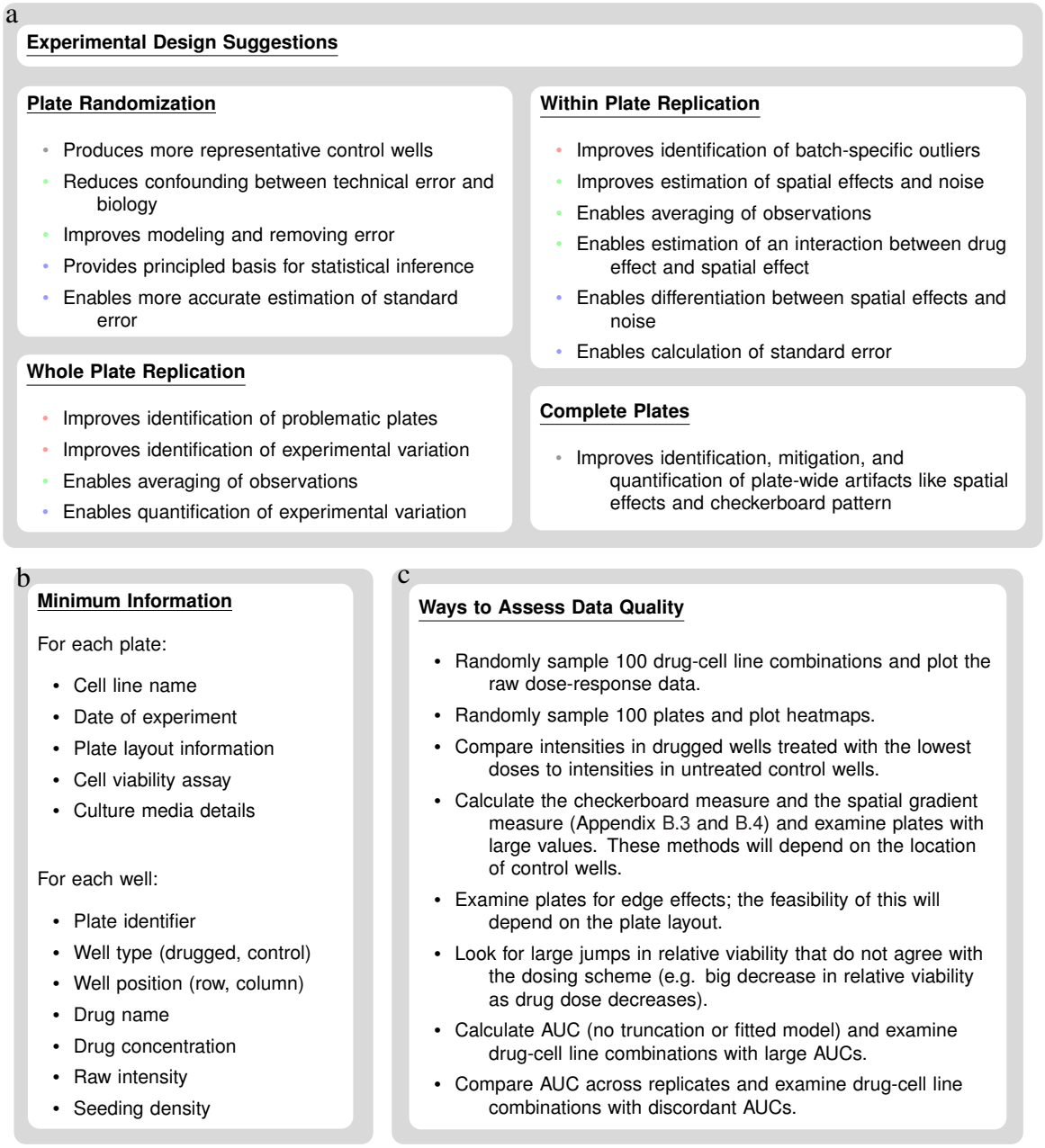

Figure 2.8: Considerations for designing and analyzing drug screening studies. (**a**) Ways in which improved experimental design can benefit the analysis of drug sensitivity data. Red bullets indicate benefits for identifying error, green bullets indicate benefits for fixing error, blue bullets indicate benefits for quantifying error, and gray bullets indicate benefits for all three. (**b**) The minimum information about a drug screening experiment that is needed for accurate and effective downstream analysis. (**c**) Suggestions for assessing the quality of raw drug screening data.

Additionally, many more processing and analysis techniques would be feasible if all collected data were released. All GDSC and CCLE plates that were publicly released contain missing values; for CCLE, more than 90% of all wells are missing. This incompleteness of the data creates an

additional challenge for analysis. Many of these missing values are due to the proprietary nature of tested compounds. To mitigate this problem, experiments could be designed so that proprietary compounds are tested on separate plates from those compounds that can be publicly released. If such separation is not feasible, then the proprietary drugs could be anonymized and included in the full data release. Similarly, observations that have failed quality control could also be released, with a quality control flag, as they might provide useful information for understanding plate-wide technical variation. Releasing all data at every processing level will facilitate a more reliable and accurate analysis (Figure 2.8b).

2.8 Discussion

Both GDSC and CCLE contain similar technical artifacts in their raw drug screening data, including spatial effects, checkerboard pattern, batch-specific outliers, and noise. Such artifacts are inevitable in high-throughput experiments and, on their own, do not necessarily prevent new discoveries. Many of the biological signals of interest in drug screening studies are reasonably strong, including individual cell lines that are highly sensitive to a specific drug ("hits"), as well as broader classes of cell lines that are at least moderately sensitive to a drug. It is possible to discover these strong signals even in the presence of technical variation. The ability to do so, and the reliability of those findings, however, depends on experimental design. In GDSC and CCLE, the interplay between systematic technical error and features of design produces confounding between error and biology and reduces the reliability of apparent biological signals.

Small changes to plate design can substantially improve the reliability of analyses (Figure 2.8a). Specifically, randomized plate layouts and within-plate replication can help mitigate the effects of complex errors and produce independence between error and the biology of interest. While raw data might contain strong artifacts, these experimental design choices can go a long way towards lessening the impact of such technical variation and producing more reliable downstream analysis. Several of these features are already in use, including between-plate replication in CCLE and some within- and between-plate replication in newer GDSC releases. Employing multiple forms of replication consistently and introducing plate randomization will improve data quality even further. Notably, the high concordance we found for CCLE replicates tested in the same location across plates is strong evidence that simply adding randomization to existing experimental methods could greatly improve data analysis (Figure 2.6).

These changes to design, as well as the release of full experimental information, are particularly important given the trend toward widespread use of public data from a few large pharmacogenomic studies. Well-documented experimental protocols and complete data will allow researchers from around the world to use these large databases effectively and efficiently for anti-cancer drug dis-

covery.

For existing drug screening data from studies like GDSC and CCLE, a cautious assessment of data quality is important (Figure 2.8c). Whether using these data to identify novel anti-cancer drugs, validate the results from an independent study, or examine the relationship between a single drug and cell line, the technical variation and experimental design considerations outlined in this paper may be relevant. To that end, our Shiny app allows the visualization of individual dose-response curves and plate-wide heatmaps to facilitate an exploration of the GDSC and CCLE data. Developing methods to quantify the prevalence of each type of technical error on individual plates could also provide further understanding of data quality. In general, thoroughly examining drug screening data at every level and carefully considering the impact of the experimental design will allow data from high-throughput drug screening studies to be used more effectively and reliably.

CHAPTER 3

Flexible and Spatially Varying Normalization for Well-Based Assays

3.1 Introduction

Data collected in high-throughput well-based biological assays must be adjusted to reduce both between-plate and within-plate variability. For example, in the drug screening studies discussed in Chapter 2, researchers test a wide range of potential pharmaceutical drugs against cell lines, exploring their effectiveness at inhibiting cell growth. These experiments produce an intensity measurement for each well on the microplate that quantifies cell activity in that well. The magnitude of intensity measurements, however, can vary widely from plate to plate due to factors like differences in lab conditions across experiments. Therefore, to accurately analyze drug screening data, raw intensities are often normalized into relative viabilities that share a unified scale; this allows comparability across plates and across studies.

In addition, we, and others, have shown how individual plates can be impacted by within-plate technical variation, including complex spatial biases (*Brideau et al.*, 2003; *Mpindi et al.*, 2015; *Caraus et al.*, 2015; *Haverty et al.*, 2016; *Niepel et al.*, 2019; *Wang et al.*, 2020; *Rehnberg et al.*, In Preparation). These errors, such as row and column effects, edge effects, and checkerboard patterns, must be eliminated for accurate analysis of the data.

Some plate layouts, however, can make it difficult to accurately separate biological signal from technical artifacts within a given plate. Often, control wells are located on plate edges and consecutive drug doses are tested in consecutive wells which can lead to confounding between spatial gradients and biology. Further, publicly released data may have large numbers of wells with missing intensity measurements corresponding to tests of proprietary compounds for which the data were not released. Together, these greatly increase the difficulty of adjusting for between-plate and within-plate variability and calculating accurate normalized scores.

The main contribution of this chapter is a framework for normalization in the context of such complex spatial errors and complex plate designs. Specifically, we frame the normalization prob-

lem as a counterfactual estimation problem. In this setting, the quality of our normalization depends on our ability to accurately estimate the counterfactual for each treated well. Therefore, we leverage the flexibility of this framework and carefully combine two counterfactual estimation approaches that, together, produce more reliable results than either on its own.

While this framework is broad enough to accommodate different types of well-based biological data, we focus much of this chapter on data from high-throughput drug screening studies. For example, our normalization method takes advantage of the sparsity of drugs that are effective at low concentrations in such studies. We use these low dose treated wells alongside control wells to improve the performance of our normalization procedure.

Further, we carefully apply this normalization framework to the drug screening data from GDSC (*Yang et al.*, 2013). When applied to GDSC, our normalization approach produces more reasonable estimates for area under the dose-response curve (AUC) than existing methods. We produce fewer overly large AUC estimates, while still capturing small AUCs corresponding to sensitive drug-cell line combinations. We also improve agreement in AUC estimates for replicated drug-cell line combinations for many of the tested compounds.

3.2 Normalization Frameworks

There are many different techniques for processing data from microplate-based biological experiments that fall under the umbrella of "data normalization". Each of these methods specifies a different targeted output and makes different assumptions about the experimental design and error structure in the data. We make the general claim, however, that, at its core, data normalization is simply a process that aims to accurately isolate the biology of interest and minimize the influence of errors, while often adding interpretability, as well.

More specifically, we assert that a normalization method should

(a) produce scores that are on an interpretable scale.

(b) produce scores that can be meaningfully compared within plates, across plates, and across studies.

(c) remove errors, artifacts, and non-biological variation in the data.

(d) keep all biological signal.

Many existing normalization techniques for well-based assays do not meet all of these goals in complex experimental and data settings. In particular, existing methods tend to focus either on mitigating between-plate variability or removing within-plate errors. Existing methods also typically make strict assumptions about data structure to simplify their approach. These assumptions,

however, are often unrealistic in complex data settings, causing the methods to fall short of their stated goals. We discuss several of these existing methods, and their drawbacks, in Section 3.2.1. In Section 3.2.2, we introduce a new normalization framework based on counterfactual estimation. This approach is flexible, interpretable, and addresses all four normalization goals.

3.2.1 Existing Methods for Normalization

Consider an experiment with plates indexed by $k = 1, \ldots, K$ and individual wells indexed by row $i = 1, \ldots, I$ and column $j = 1, \ldots, J$. For well (i, j) on plate k, the observed intensity is denoted by Y_{ijk}. Several existing methods for normalizing these observed intensities are discussed below.

Relative Viabilities Relative viability methods aim to quantify the cell growth in well (i, j) as a proportion of uninhibited cell growth. The normalized values are on a unified scale from 0 to 1, giving them a clear interpretation: a relative viability of 0 indicates no cell growth in well (i, j), while a relative viability of 1 indicates uninhibited cell growth in that well. This approach aims to eliminate between-plate variability by relating each well intensity on plate k to that plate's reference intensity (*Mpindi et al.*, 2015).

The intensities in untreated control wells, i.e., wells that contain cells but no drug, are often used to gauge what uninhibited cell growth looks like. A common approach to relative viability normalization, therefore, is to use the median intensity of the untreated control wells on plate k to scale the drugged intensities on that plate. This produces what we will call untreated control (UC) relative viabilities:

$$\mathrm{V}_{ijk}^{uc} = \frac{Y_{ijk}}{\tilde{U}_k}, \tag{3.1}$$

where $\tilde{U}_k$ is the median of the untreated control wells on plate k. This relative viability definition is used, for instance, in a large drug screening study, the Cancer Cell Line Encyclopedia (CCLE) (*Barretina et al.*, 2012).

The UC normalization method improves the interpretability of drug screening data by relating each raw intensity to the intensity associated with uninhibited cell growth. Additionally, it improves the comparability of data across plates and across studies by mitigating the plate-to-plate differences in the baseline magnitude of well intensities. The quality of this normalization method, however, can be affected by the presence of technical variation *within* a given plate (*Mpindi et al.*, 2015). As discussed in Chapter 2, many plates suffer from within-plate spatial bias; UC normalization does not address this type of error. More generally, because UC normalization simply rescales the data on a plate by plate basis, it is unable to mitigate the effects of any within-plate variability.

There are many variations on this relative viability calculation. In addition to the untreated control wells, some methods also depend on the use of blank control wells. These wells contain no cells and no drug. Blank controls correspond to a viability of 0, or the well intensity when all cell activity has stopped. The normalization procedure applied by GDSC, for instance, uses both types of control wells to calculate relative viabilities,

$$V_{ijk}^{GDSC} = \frac{Y_{ijk} - \bar{B}_k}{\bar{U}_k - \bar{B}_k},$$

where $\bar{B}_k$ is the mean of the blank controls on plate k and $\bar{U}_k$ is the mean of the untreated controls on plate k (*Vis et al.*, 2016). The use of blank control wells can add substantial noise into the relative viability calculation; therefore, in this chapter, we choose to focus on the UC relative viabilities defined in Equation 3.1 (see Appendix C.2 and *Mpindi et al.* (2015)).

Another common aspect of relative viability normalization is truncation. In an attempt to prevent noise from pushing relative viabilities outside the reasonable range, these methods may include truncation at a minimum relative viability of 0 and a maximum relative viability of 1. Such truncation, however, could also eliminate meaningful biological signal. This could happen, for instance, on a plate with technical error that manifests as a gradient in well intensities. If the drugged wells on such a plate are located in a high intensity region compared to the control wells, the calculated relative viabilities will be larger than 1. Truncation in this case will eliminate all variability, both technical and biological.

Z-Scores Z-score normalization indicates how many standard deviations the intensity in a given treated well on plate k is from the mean treated intensity on that plate (*Mpindi et al.*, 2015; *Murie et al.*, 2014; *Caraus et al.*, 2015). The Z-score for treated well (i, j) on plate k, for example, is calculated as follows:

$$Z_{ijk} = \frac{Y_{ijk} - \bar{Y}_k}{s_k},$$

where $\bar{Y}_k$ and s_k are the mean and standard deviation (SD) of the treated wells on plate k, respectively.

It is an implicit assumption of this method that the SD of the treated wells should be the same across plates. In reality, however, the SD may vary meaningfully from plate to plate (*Brideau et al.*, 2003). Importantly, some of this variation is caused by the selection of drugs tested on each plate and is, therefore, related to the biology of interest. For example, a plate with all ineffective drugs will have a small SD, while a plate with a mixture of effective and ineffective drugs will have a large SD. As a result, the same Z-score will be calculated for both a slightly low intensity well on a plate with a small SD and for a very low intensity well on a plate with a large SD. That is, the

Z-score for a given well depends on the intensities in the other drugged wells on the same plate, implying that Z-scores are not necessarily comparable across plates.

Another clear drawback of Z-scores, like relative viability normalization, is their inability to deal with spatial effects (*Brideau et al.*, 2003; *Caraus et al.*, 2015). Finally, this method ignores the presence of any control wells that were included in the experiment for the purpose of between-plate normalization.

Spatial Regression Adjustment With regression-based methods, the raw intensities for a given plate are regressed on plate location, often row and column number. The residuals from the fitted model indicate the well intensities that remain after removing within-plate spatial effects. A separate regression is fit for each plate, allowing spatial bias to differ across plates. Further, any type of regression, i.e., linear, loess, etc., can be used to adjust the raw data. One such regression-based method is as follows (*Mpindi et al.*, 2015):

$$\hat{Y}_{ijk} = Y_{ijk} - (\hat{Y}^r_{ijk} - \text{median}(\hat{Y}^r_{ijk})), \tag{3.2}$$

where $\hat{Y}^r_{ijk}$ is the fitted value from the regression for well (i, j) on plate k and the median is taken across all such wells on plate k. The resulting shifted residuals, $\hat{Y}_{ijk}$, can be used in place of raw intensities when calculating relative viabilities.

A related spatial regression adjustment method is based on fitting a separate regression to each row and column of each plate:

$$\hat{Y}_{ijk} = Y_{ijk} \times \frac{\bar{r}_i}{r_{ij}} \times \frac{\bar{c}_j}{c_{ij}}, \tag{3.3}$$

where $\bar{r}_i$ is the mean of the row i fitted values, r_{ij} is the row i fitted value for column j, $\bar{c}_j$ is the mean of the column j fitted values, and c_{ij} is the column j fitted value for row i (*Caraus et al.*, 2015).

These regression-based methods aim to eliminate within-plate spatial bias, but they can be ineffective, and even harmful, when plates are not randomized (*Caraus et al.*, 2015). On plates where consecutive drug doses are tested in consecutive wells, for instance, an effective compound appears as a gradient across treated wells. A regression adjustment method can falsely identify these biological effects as spatial bias and inadvertently introduce error to a previously clean plate (*Rehnberg et al.*, In Preparation). This concern is particularly relevant for the method introduced in Equation 3.3 where a separate regression is fit to each row and column of the plate. For regression-based methods to be effective, therefore, randomization is key. Further, these methods must be paired with other normalization approaches that target between-plate variability.

B-Scores The B-score normalization method relies on Tukey's iterative median polish algorithm to remove row and column effects (*Mpindi et al.*, 2015; *Brideau et al.*, 2003; *Makarenkov et al.*, 2007; *Caraus et al.*, 2015). Like the regression-based methods above, B-scores aim to eliminate within-plate spatial bias and can be used in lieu of raw intensities in relative viability calculations. For well (i, j) on plate k, the B-score is calculated as

$$B_{ijk} = \frac{Y_{ijk} - (\hat{\mu_k} + \hat{r}_{ik} + \hat{c}_{jk})}{MAD_k},$$

where $\hat{\mu}_k$, $\hat{r}_{ik}$, and $\hat{c}_{jk}$ are the global effect, row effect, and column effect estimated by median polish for plate k, respectively. MAD_k is the median absolute deviation (MAD) of the residuals, $e_{ijk} = Y_{ijk} - (\hat{\mu_k} + \hat{r}_{ik} + \hat{c}_{jk})$, for all wells on plate k.

A related procedure, the spatial polish and well normalization (SPAWN) method, is an adaptation of the B-score that uses trimmed mean instead of median in the polish algorithm (*Murie et al.*, 2014; *Caraus et al.*, 2015). Further, the score in well (i, j) is shifted by the median of the scores in location (i, j) across all plates and then scaled again by the MAD of plate k. This additional step allows SPAWN to address both within-plate and between-plate variation.

By relying on the polish algorithm, as with regression-based techniques, these methods implicitly assume that every row and column of every plate contains only a small number of wells sensitive to the applied drug, i.e., wells with low intensities (*Mpindi et al.*, 2015). When plate layouts are not randomized, however, it is typical for a single column to contain the highest concentration of all drugs tested on that plate. This column will tend to have systematically low intensities. The polish algorithm can therefore introduce bias by picking up on this biological signal and removing it. Further, many publicly released screening datasets have large numbers of missing wells. While the polish algorithm can handle missing values, taking the median or trimmed mean of a row with only three non-missing wells is not a meaningful way to capture spatial effects.

Finally, the MAD re-scaling in the B-score denominator means that the normalized score for a given well depends on the efficacy of the other drugs on the plate (*Mpindi et al.*, 2015). This has a similar effect to the SD re-scaling of Z-scores, as discussed above. Namely, testing a drug-cell line combination on two different plates containing different sets of compounds will produce different MAD denominators and different normalized values. This means the scores are not entirely comparable across plates or studies.

Well Correction The well correction procedure focuses on mitigating artifacts that affect each well location across time (*Caraus et al.*, 2015; *Makarenkov et al.*, 2007). For a batch of plates tested within a week, for instance, this method could eliminate the effects of cell line drift that may have occurred throughout the testing window. Well correction involves multiple steps and aims to eliminate non-biological variation both within and across plates (Algorithm 1).

Algorithm 1: Well Correction

Perform Z-score normalization within each plate.
for *each well location* **do**
    Linear (polynomial, spline, etc.) regression of Z-score on plate number.
    Residualization based on regression fit.
    Z-score normalization across plates.
end

This procedure, however, may not be effective when the plates under consideration have been tested over several years, as is common in many large-scale drug screening studies. A simple regression of well intensities across time will not be adequate to eliminate such between-plate variation. Additionally, well correction can be ineffective when all plates in the study have consistent layouts (*Caraus et al.*, 2015). If, for instance, well (i, j) always contains the highest drug dose across plates, that well may have consistently low intensities. This drug effect will be incorrectly identified as a systematic error specific to that well, captured in the intercept of the regression, and eliminated during residualization. Therefore, the use of consistent plate layouts can cause the well correction method to remove biological signal of interest.

Control Plate Regression The control plate regression (CPR) procedure uses plates containing only control wells in addition to plates containing only treated wells, instead of relying on in-plate controls. In this normalization method, each treated well is regressed on the control plate well located in the same position. CPR combines within-plate spatial correction and between-plate normalization into a single procedure (*Murie et al.*, 2014). The CPR score for well (i, j) on treated plate k is calculated as

$$\text{CPR}_{ijk} = \frac{Y_{ijk} - (\hat{\mu}_k + \hat{\beta} P_{ij})}{\hat{s}_k},$$

where $\hat{\mu}_k$ is the global effect for treated plate k, $\hat{\beta}$ is the estimated coefficient, P_{ij} is the intensity in well (i, j) on the control plate (or the median for well (i, j) across many control plates), and $\hat{s_k}$ is the robustly estimated scale parameter (e.g., re-scaled MAD of the residuals) for plate k.

A drawback of CPR is the use of completely separate control plates. Not only does this increase the total number of plates that need to be tested, it also necessitates the assumption that the structure and magnitude of spatial effects on separately constructed and scanned treated plates and control plates are the same. In an attempt to satisfy this assumption, many control plate replicates must be run close in time to, and interspersed with, the treated plates they hope to reflect (*Murie et al.*, 2014). Even with these considerations, however, plate construction can lead to plate-specific effects that differ between treated and control plates (*Mpindi et al.*, 2015).

Finally, as with the B-score and SPAWN methods, the MAD re-scaling in the CPR score denominator means that the normalized CPR score for a given well depends on the efficacy of the other drugs on the plate. The scores are not entirely comparable across plates or studies.

Comments Each of the outlined normalization methods specifies one or more types of non-biological variation it is trying to remove. Spatial regression methods, for instance, are targeting within-plate spatial bias, while relative viability methods attempt to eliminate between-plate variability in the magnitude of well intensities. Often, however, the types of errors present in large-scale biological data are complex and require multiple of these methods to be used in tandem.

Further, each normalization technique must also make a set of assumptions about the study design and the error structure in the data. These assumptions can include the use of randomized plate layouts, consistency in data variability from plate to plate, and the presence of only one type of error in the raw data. If the specific assumptions are not met, as is often the case, these existing normalization methods can be ineffective, and even harmful. In many real-world settings, they are unable to consistently meet all of our specified normalization goals.

3.2.2 Counterfactual Estimation Framework

We introduce a normalization framework that aims to be broadly effective for a diverse set of normalization problems and data settings. This framework is grounded in the idea of counterfactual estimation, a concept we borrow from causal inference (*Splawa-Neyman et al.*, 1990; *Rubin*, 1974). Specifically, in an ideal normalization setting, the intensity of each treated well would be normalized by the intensity that would have been observed if that well had not been treated. We define the counterfactual, denoted by Y^0_{ijk}, as the intensity that would have been observed in treated well (i, j) if no drug had been applied. The only difference between the observed intensity in well (i, j) and the counterfactual in the same well is the drug treatment. Therefore, the ratio of the observed intensity to the counterfactual gives the effect of the drug, which is the quantity of interest (e.g., $V_{ijk} = \frac{Y_{ijk}}{Y^0_{ijk}}$).

It is only possible, however, to observe one potential intensity, either Y_{ijk} or Y^0_{ijk}. Therefore, for each drugged well, the counterfactual must be estimated. This makes the normalization problem a counterfactual estimation problem where the quality of our normalization depends on our ability to obtain good potential outcome estimates.

Importantly, this framework provides a useful starting point for addressing all of our normalization goals. The scores produced by this approach have a simple interpretation; they simply result from relating each observed data value to the counterfactual. Further, the use of a careful counterfactual estimation approach can ensure that scores are comparable within and across plates, have

as much error removed as is feasible, and retain all biological signal.

Additionally, the flexibility of this framework makes it easy to combine multiple counterfactual estimation approaches in an ensemble method. Each individual approach will have both strengths and weaknesses, but we can combine them in a manner that emphasizes the strengths of each. This type of ensemble can be particularly useful in complex data settings and increase our ability to remove non-biological variation while avoiding the biological signal of interest. We outline our use of such an ensemble estimation method within the counterfactual framework for normalizing drug screening data below.

3.3 Flexible Normalization for Drug Screening Studies

We develop a normalization procedure for processing data from large-scale drug screening studies within the counterfactual estimation framework. This approach is motivated by common classes of errors in this type of data and by the structure of many high-throughput drug screening studies. We develop two counterfactual estimation approaches, including one that allows the counterfactual to vary spatially. Our final normalization procedure carefully combines these two estimation methods in an adaptive manner, capitalizing on the strengths of each.

3.3.1 Motivation

We aim to use the counterfactual estimation framework to calculate relative viabilities for large-scale drug screening data. This is not, however, a straightforward task. Several of the normalization methods introduced above fit into the counterfactual estimation framework, including UC relative viability normalization (Equation 3.1) and regression-based normalization (Equation 3.2). These methods highlight some of the challenges associated with counterfactual estimation.

In UC normalization, for instance, the counterfactual is estimated by the median of the untreated control wells. Importantly, this approach relies on a single estimated counterfactual value for the entire plate. As detailed in Chapter 2, large-scale drug screening studies have been shown to contain a wide range of technical variation, including spatial gradients and checkerboard patterns (Figure 3.1). When a plate contains such spatial biases, simple counterfactual estimation techniques, such as the UC relative viability calculation, struggle. The single summary value cannot accurately represent the untreated intensity for every drugged well on the plate. The median of the untreated controls, for instance, will overestimate the true counterfactual value for a drugged well located in a low intensity region of the plate compared to the control wells. To accurately estimate relative viabilities, therefore, the counterfactual must be able to vary as spatial effects vary across the plate.

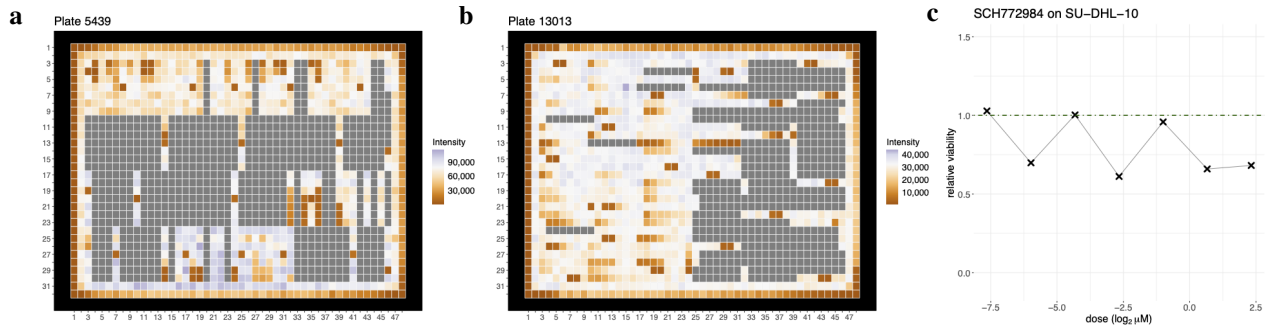

Figure 3.1: Technical variation in raw drug screening data. **(a, b)** Heatmaps of two plates containing spatial effects. Grey cells indicate missing wells. For each heatmap, white corresponds to the median intensity of the untreated control wells on the plate. **(c)** Dose-response curve for a drug-cell line combination with a pronounced checkerboard pattern.

Regression-based normalization, as in Equation 3.2, produces a value that can be interpreted as a spatially-varying counterfactual estimate. In this approach, the counterfactual for each individual well is estimated by that well's regression fitted value, centered by the plate median. While this method gives a different counterfactual estimate for each well on the plate, it depends on the spatial regression only picking up on spatial bias and not biology. Many commonly used plate layouts, however, are not randomized; they have (1) control wells placed around the edges of a plate or in a single block, (2) consecutive drug doses applied to consecutive wells, and (3) drugs tested in consistent locations across plates. Together, these aspects of design can confound the biology of interest with spatial effects and interfere with counterfactual estimation via a regression-based method.

Overall the UC relative viability and regression-based normalization methods struggle to produce reliable counterfactual estimates. The challenges they face highlight the complex nature of counterfactual estimation. In response, we combine two different, and complementary, estimation approaches, loosely based on the above UC and spatial regression approaches, into a single ensemble method. We discuss this method below.

3.3.2 Normalization Procedure

We developed an ensemble method for normalizing drug screening data that depends on two counterfactual estimation approaches; one is based primarily on the untreated control wells and the other on the drugged wells. Rather than simply averaging these two estimates, we approximate the complex error structure of each estimator and use an adaptive procedure to combine them. For each drugged well, our normalization method produces a likely estimate of the counterfactual, given the observed data. We then use these counterfactual estimates to normalize each observed intensity and obtain relative viabilities:

$$\hat{V}_{ijk} = \frac{Y_{ijk}}{\hat{Y}^0_{ijk}}. \tag{3.4}$$

We introduce the two approaches for estimating Y^0_{ijk} and the adaptive method for combining them, below.

Approach 1 We use the UC normalization method as our first counterfactual estimation approach. For each plate, Y^0_{ijk} is estimated by the median of the untreated control wells on that plate (Equation 3.1). The biggest benefit of this method is that it is universally applicable; we can quickly calculate the median of the untreated control wells for any plate, regardless of the layout and the amount of missing data. In particular, Approach 1 can be applied to all plates that contain untreated control wells, including those in the GDSC and CCLE studies. Further, this method uses untreated wells to estimate the intensity that would have been observed if other wells had also been untreated. This is a very interpretable and intuitive approach.

This method does, however, have limitations, many of which we have previously discussed. Specifically, Approach 1 provides the same counterfactual estimate for every well on a given plate. This is true regardless of any underlying spatial variation in well intensity. Additionally, while this method can be applied to any plate that contains untreated control wells, it will not always produce a high-quality counterfactual estimate. Plates tested in the CCLE study, for instance, have all of their untreated control wells placed along the right edge of the plate (Figure 2.1c). With this format, the untreated controls may suffer from edge effects or have systematically different intensities than what would have been observed for untreated control wells placed in the center or on the left side of the plate.

Approach 2 To complement the strengths and weaknesses of Approach 1, we use a localized counterfactual estimate for Approach 2. This estimation method depends on a unique feature of high-throughput drug screening studies. Specifically, in such studies, many drugs are ineffective, particularly at low concentrations. This means that the intensity in the wells treated with the lowest drug concentration is similar to the intensity that would have been seen in that well if no drug had been applied (i.e., $Y_{ijk} \approx Y^0_{ijk}$ when well (i, j) contains the lowest drug concentration). Further, we might expect $Y_{ijk} \approx Y^0_{i'j'k}$ for some well (i', j') close to (i, j). Therefore, Y_{ijk} could be a good estimate of the counterfactual for nearby wells treated with higher concentrations of the drug.

More formally, consider a drug that is tested at d doses in d consecutive wells on plate k. For this drug-cell line combination, we use the intensity from the well treated with the lowest drug dose as our counterfactual estimate. All d wells where this drug is tested on plate k will have the same estimated counterfactual.

Unlike with Approach 1, however, there are circumstances in which this method can seriously fail. Specifically, if a drug is effective at the lowest dose, the intensity in that well will be low. This lowest dose treated intensity will *not* be similar to the expected untreated intensities for that drug, and the Approach 2 estimate of the counterfactual will be poor. On a smaller scale, the presence of technical errors, including random noise, can cause the intensity in the lowest dose treated well to be a poor estimate of the counterfactual in nearby wells. Finally, if d is large, spatial effects may also hurt our estimation quality.

Adaptive Combination Procedure We aim to combine the counterfactual estimates from Approach 1 and Approach 2 in a way that capitalizes on the strengths of each and produces an estimate that is plausible with the observed data. We base our combination procedure on the following intuition. When the drug of interest is ineffective at low concentrations, the intensity from the well treated with the lowest drug dose will likely be a better estimator for Y^0_{ijk} than the untreated controls. That is, we prefer the counterfactual estimate from Approach 2 (Figure 3.2ab). If the drug is effective at low doses, however, it will be better to estimate Y^0_{ijk} with the median of the untreated control wells. In this scenario, we prefer the counterfactual estimate from Approach 1 (Figure 3.2c).

It is not always simple, however, to determine the best estimation approach. While Figure 3.2a-c shows three dose-response curves where the observed intensities, both drugged and control, clearly indicate the appropriate counterfactual estimate, the situation is not as clear for Figure 3.2d. In particular, we do not know what the behavior of this drug-cell line combination would be

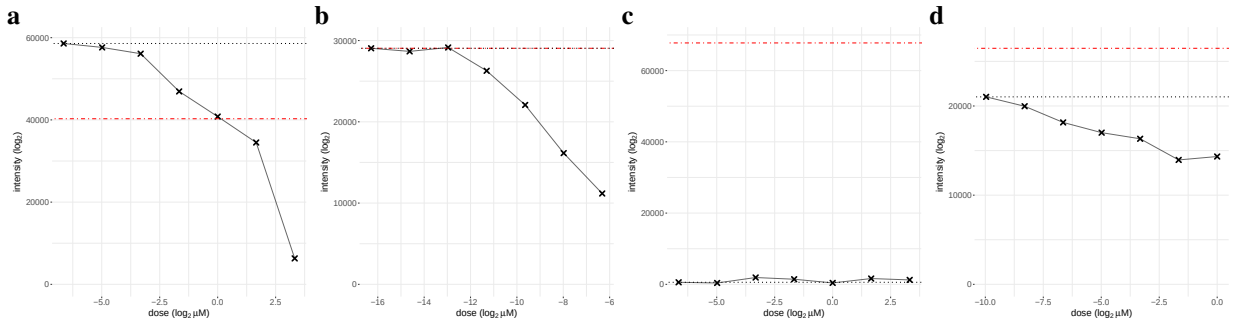

Figure 3.2: The estimated counterfactual must be appropriate given the observed data. **(a)** A dose-response curve where the median of the untreated control wells does not accurately represent uninhibited growth for this drug-cell line combination. Estimating the counterfactual with the lowest dose intensity will be preferred. The black dotted line indicates the lowest dose intensity; the red dashed line indicates the median of the untreated controls. **(b)** A dose-response curve where the median of the untreated control wells coincides with the lowest dose intensity. Either counterfactual estimation approach will be appropriate. **(c)** A dose-response curve where the drug is effective at all tested doses. Estimating the counterfactual with the median of the untreated controls will be preferred. **(d)** A dose-response curve where it is not clear which counterfactual estimation approach will be preferred.

at drug doses lower than those tested. Further, it is not obvious whether or not spatial artifacts are present on this plate. Therefore, we need an automated normalization procedure that will take in the observed data and produce the most likely counterfactual estimate.

To combine our counterfactual estimates in this manner, we must understand the error structure of each estimator. To do so, we introduce some notation. For plate k, we let the median of the untreated control wells be denoted $\tilde{U}_k$. This is the counterfactual estimate from Approach 1. We define $\varepsilon_{ijk} = \tilde{U}_k - Y^0_{ijk}$, where ε_{ijk} gives the error of $\tilde{U}_k$ as an estimator for Y^0_{ijk}. It follows that $\tilde{U}_k = Y^0_{ijk} + \varepsilon_{ijk}$. Importantly, because the counterfactual, Y^0_{ijk}, cannot be observed, the value of ε_{ijk} is also never observed. Nevertheless, we assign ε_{ijk} a density f and estimate the error distribution $f(\varepsilon)$ from the data. We estimate this distribution across all plates by taking the difference between the median of the untreated control wells and each lowest dose treated well on the plate, taking care to ignore effective drugs (see Appendix C.4 for details). The resulting estimated error distribution is symmetric, but has heavy tails due to the presence of noise, outliers, and spatial bias that can impact the quality of the median of the untreated controls as a counterfactual estimate.

In our complementary approach, we consider the drug-cell line combination tested in the d consecutive wells that includes well (i, j). We let the intensity of the well treated with the lowest dose of that drug be denoted L_{ijk}. This is the counterfactual estimate for well (i, j) from Approach 2. We let δ_{ijk} be the error of L_{ijk} as an estimator for Y^0_{ijk}, and define $\delta_{ijk} = L_{ijk} - Y^0_{ijk}$. It follows that $L_{ijk} = Y^0_{ijk} + \delta_{ijk}$. Again, we never observe the value of δ_{ijk}, but we assign δ_{ijk} a density g and estimate the error distribution $g(\delta)$ from the data. We estimate this distribution by taking the difference in intensity measurements between lowest dose treated wells that are tested in adjacent locations. The resulting estimated distribution of δ_{ijk} is entirely less than 0 and has a long left tail (see Appendix C.4 for details). This tail is caused by drugs that are effective at low doses.

With these estimated distributions, we now use a procedure that selects the most likely value of the counterfactual for well (i, j) on plate k given the observed data for that well ($\tilde{U}_k$ and L_{ijk}). Optimizing a parameter value given data is essentially the idea of maximum likelihood. Therefore, we write down the likelihood of Y^0_{ijk}, which is a function that depends only on the difference between L_{ijk} and $\tilde{U}_k$:

$$\begin{aligned}\mathcal{L}(Y^0; \tilde{U}, L) &= f(\tilde{U} - Y^0) g(L - Y^0) \\ &= f(\varepsilon) g\big((L - \tilde{U}) + \varepsilon\big).\end{aligned}$$

The above formulation depends on the assumption that, for any given well, ε_{ijk} and δ_{ijk} are independent. In practice, this is not true. As is often seen with the naive Bayes classifier, however, we believe this approach can be effective even when the independence assumption is violated.

Therefore, we maximize $\mathcal{L}(Y^0; \tilde{U}, L)$ to get an estimate for the counterfactual for well (i, j) on

plate k, $\hat{Y}^0_{ijk}$. With this counterfactual estimate, relative viabilities are calculated as in Equation 3.4. This relative viability is the ratio of the drugged intensity in well (i, j) on plate k to the most likely estimate of the counterfactual for that well. We expect this normalized score to isolate the drug effect and be interpretable across plates and studies.

3.3.3 Extensions

One of the biggest strengths of this ensemble normalization procedure is the flexibility it provides. In particular, the counterfactual estimation methods used in Approach 1 and Approach 2 can be tailored to the specific experimental design or error structure of the study being analyzed. We have, for example, tailored the above procedure to the GDSC data described in Section 3.4 below.

We first focus on Approach 1 and its difficulty with within-plate spatial bias. In some versions of the GDSC data, the untreated control wells are spread across the entire plates. Therefore, instead of estimating the counterfactual with the median of the untreated controls, we use those wells to model the spatial effects across each plate. That is, instead of using a single summary measure of the control wells as an estimate of the counterfactual, we fit a spatial model that predicts what the counterfactual should be for each drugged well.

Further, as in Approach 2, we take advantage of the ineffectiveness of most drugs at low concentrations. We use the drugged wells treated with the two lowest drug doses to supplement the untreated controls. This strategy allows us to use a larger number of wells to model spatial variation. Specifically, for plate k, we let the intensities in the untreated control wells and the wells treated with the two lowest doses of each drug represent "untreated" intensities. We then use a robust loess regression to estimate the spatial variation in these "untreated" wells and to produce a prediction of the untreated intensity for each drugged well (Appendix C.4.1 has a more precise definition; *Cleveland et al.* (1992)). For GDSC, we use this prediction in place of the median of the untreated control wells as the counterfactual estimate from Approach 1.

We also improve Approach 2, tailoring it to mitigate the effects of common errors in the GDSC data. In particular, the intensity from the lowest concentration drugged well can be affected by technical errors. The presence of random noise and extreme outliers, for instance, can cause the observed intensity in any single well to be noisy and a poor estimate of Y^0_{ijk}. Additionally, the presence of a checkerboard pattern could cause us to systematically over- or underestimate the counterfactual for every well on the plate.

Therefore, for each drug-cell line combination, instead of simply estimating the counterfactual with the intensity from the well treated with the lowest drug dose, we fit a dose-response curve to the d observed measurements (Appendix C.4.2). We then take the average of the fitted values for the wells treated with the two lowest drug concentrations. This average is our GDSC-specific

counterfactual estimate from Approach 2. We use the average of the two lowest concentrations, rather than just the lowest concentration, to mitigate the effects of checkerboard pattern and noise. Further, we developed a dose-response curve fitting procedure that is robust to outliers and checkerboard pattern (Appendix C.4.2). Specifically, we iterate between regressing out checkerboard pattern and fitting a logistic curve to the residuals.

These improved counterfactual estimates are still combined in the maximum likelihood manner previously introduced. We apply this normalization procedure to the GDSC data below.

3.4 Application to Drug Screening Data

We apply our tailored normalization method to drug screening data from GDSC (*Yang et al.*, 2013). For this application, we use a different version of the data than in Chapter 2; here, we focus on release 8.2 of the GDSC2 data. In this version of the study, 196 compounds were tested against 809 cancer cell lines. All tests were conducted on 1,536-well microplates, each containing one cell line and multiple drugs. Each drug was tested at 7 concentrations, and consecutive drug doses were applied to consecutive wells. In addition to drugged wells, each plate also had untreated control wells which contain cells, but no compound. All plates also have wells with missing intensity values; 18% of plates have missing values for more than half of all wells.

In GDSC, many drug-cell line combinations were tested more than once, either on the same plate or across different plates. We considered these replicates separately when doing our normalization (i.e., we did not average replicate observations). Additionally, we implemented a set of quality control measures before analyzing the GDSC data. We developed these measures to identify and eliminate the noisiest plates for which normalization would not be effective (Appendix C.2 has details).

3.4.1 Results

For each drug-cell line combination in the GDSC study, we calculated relative viabilities using the UC normalization and our new normalization method (denoted EML for empirical maximum likelihood), both with and without truncation at 1. We then calculated AUC estimates from each set of relative viabilities, resulting in four AUC estimates for each drug-cell line combination. We found that our normalization produces a more reasonable distribution of AUC values than the UC normalization does (Figure 3.3a). Our method produces far fewer AUCs larger than 1. This illustrates our ability to handle spatial effects that cause insensitive drug-cell line combinations to have overly large UC relative viabilities and AUCs. Further, our normalization method still captures small AUC values for sensitive drug-cell line combinations.

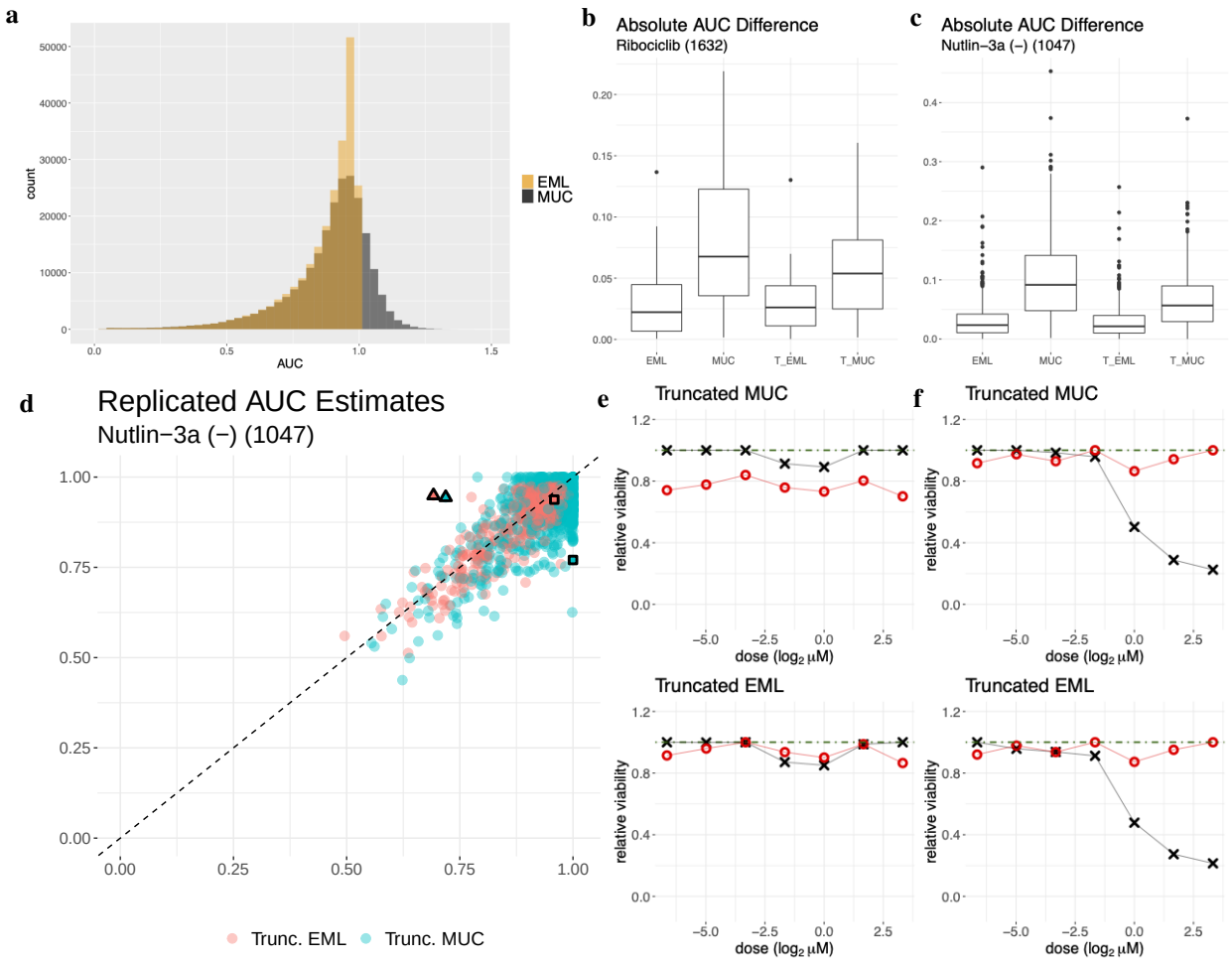

Figure 3.3: Our flexible and spatially-varying normalization method improves AUC reliability. (**a**) Histogram of AUC values calculated with the UC and our (EML) normalization methods (n = 231,209 drug-cell line combinations). Our normalization produces fewer unreasonably large AUC values while still capturing low AUC values for sensitive drug-cell line pairs. AUC is the numerically integrated area under the dose-response curve. (**b**) Absolute difference in AUC for ribociclib (n = 47 cell lines) and (**c**) nutlin-3a (-) (n = 753 cell lines). Our normalization produces better agreement for both drugs. `T_EML` and `T_MUC` indicate the truncated versions of the EML and MUC relative viabilities. (**d**) Replicated AUC values for the drug nutlin-3a (-) (n = 753 cell lines). The squares indicate a cell line where our normalization improves agreement; the triangles indicate a cell line where agreement is poor for both normalization methods, potentially reflecting biological differences between the replicates. (**e**) Dose-response plots for the cell line marked with the squares. (**f**) Dose-response plots for the cell line marked with the triangles.

Next, we considered AUC agreement for replicated drug-cell line combinations. For each replicated drug, we compared the median absolute difference in AUC values across replicates when using our method and when using UC normalization. A smaller median absolute difference, and a distribution of differences closer to 0, indicates better agreement between replicates. To compare the normalizations, we defined three categories: drugs where we increase agreement, drugs where agreement is similar across normalizations, and drugs where we decrease agreement (Appendix

C.5.2). Overall, we found that our normalization method increases agreement in AUC estimates over truncated UC normalization for 37% of drugs, while decreasing agreement for less then 3%. The drugs where our method produces better AUC agreement include ribociclib (Figure 3.3b) and nutlin-3a (-) (Figure 3.3cd).

We further investigated the impact of testing location on the performance of our normalization method. Specifically, a drug-cell line combination can be replicated in the same location across plates or in different locations across plates. We considered drugs that were replicated on at least 10 cell lines with each replication scheme and found that our normalization method has a more pronounced positive effect for replicates tested in different locations. Our normalization increased agreement in AUC estimates over truncated UC normalization for 17% of drugs whose replicates were tested in the same location across plates, but for 54% of drugs whose replicates were tested in different locations. In both scenarios, our method did not decrease agreement for any drugs. This increased benefit for replicates tested in different locations indicates that our normalization method is mitigating the impact of location-specific systematic effects on AUC estimates (see e.g., Figure 2.6).

Finally, we can see the impact of our normalization method on individual drug-cell line combinations. For many combinations, agreement is improved over UC normalization; for others, agreement is poor regardless of the normalization method used. For example, while agreement for nutlin-3a (-) largely improves with our normalization (Figure 3.3de), there are some cell lines for which agreement remains poor (Figure 3.3df). For such drug-cell line combinations, we often observe that the dose-response relationship is completely different across replicates, potentially indicating a difference in the biological response of the cells to the drug. In such cases, we believe that the dose-response curves for these replicates simply cannot be made to agree, i.e., the differences are beyond the scope of any normalization, and additional investigation is needed to understand the discrepancies (*Rehnberg et al.*, In Preparation).

3.5 Discussion

Normalization is a vital step in the analysis of drug screening data. A good normalization procedure needs to remove technical variation, both within plates and between plates, to produce biologically meaningful normalized scores. This task is made challenging by the presence of complex spatial biases and the use of non-randomized plate designs that can cause confounding between technical error and the biology of interest. Our relative viability normalization method specifically targets these issues and improves the reliability of drug sensitivity results in the GDSC study. With this approach, we frame normalization as a counterfactual estimation problem.

This normalization framework is effective for the GDSC drug screening data, and we expect

some of our innovations to be more broadly applicable. For instance, there is a large class of normalization problems that can be viewed through a counterfactual estimation lens. For some applications, the counterfactual definition may be a bit less concrete than the counterfactual used for drug screening studies. For example, when normalizing microarray data to eliminate batch effects, we may think of the counterfactual as the data we would have observed if no batch effects had been present. The counterfactual in this case is arguably more hypothetical, but may still be a fruitful way to frame the normalization problem.

There are also several unique aspects of our normalization technique that may be applied more broadly. In particular, we use the wells treated with the lowest drug doses as surrogate untreated control wells. This capitalizes on the fact that, in preliminary drug screens, most compounds are completely ineffective at low concentrations. Such a strategy can help to mitigate concerns about the location of untreated control wells on a given plate and the distance between drugged wells and the nearest control wells.

Additionally, our normalization framework optimizes between two imperfect counterfactual estimation methods. On their own, each approach has substantial drawbacks, while their combination preforms well. When the errors and artifacts in collected data are complex, using this type of flexible ensemble normalization approach may produce more reliable results.

Finally, due to the identification of checkerboard pattern in multiple high-throughput drug screening studies, we incorporate checkerboard mitigation features in our normalization framework. These features include averaging across pairs of wells that may suffer from a checkerboard pattern and implementing a dose-response curve fitting procedure that regresses out the checkerboard pattern before fitting a logistic curve. In general, looking out for the potential consequences of complex spatial biases can improve the analysis of high-throughput biological data.

Applying this, or any, normalization method, however, is not a substitute for improving the experimental design of drug screening studies, as described in Chapter 2. Small changes to design including randomizing plate layouts and consistently using replication will allow straightforward statistical methods to be used to understand, model, and remove many types of technical variation. A randomized plate layout, for instance, reduces confounding between technical variation and biology and allows a simple regression to remove spatial effects without removing biological signal. With this type of design, accurately and reliably normalizing drug screening data becomes a more feasible task.

CHAPTER 4

Computationally Efficient Approximate Cross-Validation for High-Dimensional Linear Discriminant Analysis

4.1 Introduction

High-dimensional classification can be very computationally expensive. Consider a dataset consisting of a thousand observations and half a million features, and where hundreds of separate classifiers must be fit; in pharmacogenomic studies, for instance, a classifier is fit for each of several hundred compounds under consideration. Further, many common classifiers require tuning parameter selection, which is typically done through a cross-validation (CV) procedure. To get accurate error estimates, another round of CV must be performed. Together, these nested CV loops, run for hundreds of classifiers, can take a week to complete. On top of this, researchers might be interested in the effects of different data preprocessing methods on classifier performance or in how the classifier behaves on different subsets of the data. These goals would require the entire analysis to be performed repeatedly, necessitating an impractical amount of time and computing power.

In this chapter, we focus specifically on how these challenges impact a simple linear classifier, linear discriminant analysis (LDA). This method classifies observations based on a linear combination of their features. To build an LDA classifier, we must estimate only class means, class probabilities, and the inverse of the pooled covariance matrix. Despite this simplicity, however, LDA can still suffer from lengthy computation times.

Further, high-dimensional data can cause the usual estimates of population parameters to be poor. For instance, when there are fewer observations n than features p, the sample covariance matrix will be singular. Regularization of the covariance matrix may be necessary to fit a high-dimensional model, including a high-dimensional LDA classifier. Further, such regularization typically requires the selection of the regularization parameter via CV.

Addressing the challenges of high-dimensional classification and lengthy CV computation times is not a new goal. One approach is to reduce the dimensions of the feature matrix via univariate feature selection, principal component analysis, or partial least squares (*Krzanowski et al.*, 1995). The resulting low-dimensional feature matrix can then be used to fit classical models, but choosing the optimal amount of dimension reduction will require CV. Other methods directly target the singularity of the sample covariance matrix; this includes using generalized inverses, such as the Moore-Penrose inverse (*Krzanowski et al.*, 1995; *Guo et al.*, 2006; *Xu et al.*, 2009; *Cai and Liu*, 2011). A more common approach is to create a modified covariance matrix. The independence rule, for instance, diagonalizes the sample covariance matrix based on the assumption of independence between features (*Bickel and Levina*, 2004; *Cai and Liu*, 2011). Alternatively, a large class of methods focus on covariance matrix regularization, creating a modified covariance matrix that is non-singular (*Krzanowski et al.*, 1995; *Thomaz et al.*, 2006; *Guo et al.*, 2006). Again however, many of these regularization methods require CV to tune the regularization parameter, a costly task. One such method that does not require parameter tuning is maximum uncertainty LDA (MLDA) (*Thomaz et al.*, 2006; *Xu et al.*, 2009). This technique modifies the sample covariance matrix so that all eigenvalues smaller than the mean eigenvalue are replaced with the mean. By replacing the null eigenvalues with a non-zero value, the modified matrix will be invertible. *Payne and Gagnon-Bartsch* (2022) also introduce a regularization-based LDA method that does not require parameter tuning. We build upon their method in this chapter.

Approaches that regularize the sample covariance matrix solve the non-invertibility problem caused by high-dimensions; however, the computation time needed to invert the modified $p \times p$ covariance matrix is still $O(p^3)$. This inversion is often the most computationally-intensive step in fitting a model with large p (*Cawley and Talbot*, 2003; *Hastie and Tibshirani*, 2004; *Guo et al.*, 2006; *Treder*, 2018). Therefore, several strategies have been developed to reduce the dimensions of necessary matrix inversions. Particularly, LDA and other methods can be adapted to require the inversion of $n \times n$ matrices rather than $p \times p$ matrices, a big improvement when $n \ll p$. Such dimension-reduction can be achieved via singular value decomposition of the feature matrix (*Hastie and Tibshirani*, 2004; *Guo et al.*, 2006) or via the Sherman-Morrison-Woodbury formula (*van de Wiel et al.*, 2021). Further, *Cai and Liu* (2011) noted that the covariance matrix only appears in the LDA model inverted and multiplied with the class means. Therefore, rather than estimating these quantities separately, they directly estimate their product. This linear program discriminant rule (LPD) eliminates the need to invert a large, and potentially singular, covariance matrix, improving computation time.

There has also been a specific focus on improving the speed of CV implementations; many of these improvements, however, have focused on ridge regression rather than LDA. The most intuitive strategy avoids the recalculation of large matrices at each CV iteration. Instead, intermediate

matrix values can be saved and used for more computationally efficient calculations within the CV loop (*Cawley and Talbot*, 2003; *Treder*, 2018; *Hastie et al.*, 2019; *van de Wiel et al.*, 2021). In particular, *Hastie et al.* (2019) focused on leave-one-out (LOO) CV and introduced a shortcut version for high-dimensional ridgeless regression. This method produces a LOO CV error estimate with computation time equivalent to fitting just a single ridgeless regression model. Others have focused on fast CV for LDA, demonstrating that the covariance matrix does not need to be fully re-calculated and inverted at each CV iteration (*Cawley and Talbot*, 2003; *Treder*, 2018). Finally, the computation speed of CV for ridge regression has been improved with an approximate CV procedure. *Meijer and Goeman* (2013) developed a method that approximates the parameter estimates for each CV iteration with a Taylor expansion around the full data parameter estimates. This strategy requires the full model to be fit only once.

Our main contribution in this chapter is a fast approximation to leave-one-out cross-validated high-dimensional linear discriminant analysis. Our approach to high-dimensional LDA combines dimensionality reduction via principal components (PCs) with covariance matrix regularization. Importantly, we pair these techniques in such a way that there is no need for tuning parameter selection. While our method does require a regularization parameter, we follow the lead of *Thomaz et al.* (2006) and *Payne and Gagnon-Bartsch* (2022) by identifying a natural value of that parameter. Beyond this approach, we vastly improve the speed of CV for high-dimensional LDA. Specifically, we introduce model approximations in the LOO CV fitting procedure, implement quick downdating for large matrix calculations within the CV loop, and take advantage of the data structure to avoid redundant, and expensive, calculations.

Simulations indicate that this approach allows us to fit a classifier and calculate approximate LOO CV accuracies substantially faster than existing methods. Further, our estimates of model performance, obtained via approximate LOO CV, are almost identical to the model performance estimates obtained with classical LOO CV; the approximations aimed at decreasing computation time do not hinder our ability to recover theoretical accuracies. On real pharmacogenomic data, our fast and approximate LOO CV method performs orders of magnitude faster than existing methods and obtains comparable classification quality. We have made this method available through the R package `fastLDA`.

4.2 Background and Motivation

In the standard classification setting, the goal is to predict class labels (e.g., does a cancer drug work?) from a set of features (e.g., tumor gene expression levels). We let Z be an observed predictor matrix with n observations and p features, and let Y be an observed $n \times d$ indicator matrix of class labels. We suppose there are d classes.

While many methods exist for performing classification in a high-dimensional setting (i.e., $n < p$), such an analysis is not always straightforward. Here we discuss existing methods of generalizing the low-dimensional LDA algorithm to high dimensions and the challenges that can arise. Many of these issues are related to tuning parameter selection and computation time.

4.2.1 Linear Discriminant Analysis in High Dimensions

Linear discriminant analysis (LDA) is a common, and simple, classification approach that uses a linear combination of the features to predict class labels. This method assumes that the features are normally distributed, conditional on class label, with mean μ_k for class $k = 1, \ldots, d$ and with common covariance matrix across classes, Σ. The LDA score for class k, obtained for an out-of-sample observation, $\tilde{Z}$, is calculated as follows

$$\hat{s}_k = \tilde{Z}\hat{\Sigma}^{-1}\hat{\mu}_k + log(\hat{\pi}_k) - \frac{1}{2}\hat{\mu}_k^\intercal\hat{\Sigma}^{-1}\hat{\mu}_k, \tag{4.1}$$

where the parameter values $(\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma})$ are estimated from the training data (Z, Y). We can further define $\hat{\beta}_1^{(k)} = \hat{\Sigma}^{-1}\hat{\mu}_k$ and $\hat{\beta}_0^{(k)} = log(\hat{\pi}_k) - \frac{1}{2}\hat{\mu}_k^\intercal\hat{\Sigma}^{-1}\hat{\mu}_k$ so that $\hat{s}_k = \tilde{Z}\hat{\beta}_1^{(k)} + \hat{\beta}_0^{(k)}$. For each out-of-sample observation, the score is calculated for every class. The observation is then assigned to the class that produces the largest score.

A key assumption of LDA is that the features are normally distributed given the class labels. Formally, we must assume $Z|Y = k \sim \mathcal{N}(\mu_k, \Sigma)$ for class $k = 1, \ldots, d$. In this work, we use a more specific version of this assumed model that further assumes the presence of low-dimensional biological factors. As such, we model

$$Z_{n\times p} = L_{n\times\ell}\alpha_{\ell\times p} + \varepsilon_{n\times p}, \tag{4.2}$$

where $L|Y \sim \mathcal{N}(Y\eta, \Psi)$, with $\eta \in \mathbb{R}^{d\times\ell}$. Here, L represents the low-dimensional biological factors and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_p)$ represents the variation not accounted for by L. With this model, the features in Z are related to the class labels in Y through the low-dimensional factor matrix L.

We can understand this model through the lens of binary drug screening classification. In this setting, the variable of interest, Y, indicates whether or not each sample is sensitive to the tested drug. The feature matrix, Z, contains gene expression levels for each of those samples. Gene expression levels, however, will be affected by biological variables beyond our single variable of interest, including original tissue source, tumor size, and cancer metastasis. Not only do these variables affect the expression levels of many genes at once, but they are also related to whether or not a drug is effective. Most of the variation in Z can be captured by these few main biological factors contained in L, making this low-dimensional factor model appropriate.

When modeling this type of data it can be useful to perform principal component analysis and use only the top PCs, corresponding to the few informative dimensions, in model building. Therefore, rather than traditional LDA, principal component linear discriminant analysis (PC-LDA), a related method that first performs dimensionality reduction on the feature matrix, may be preferred. This approach is also beneficial as traditional LDA performs poorly in high-dimensions (*Peck and Van Ness*, 1982). To perform PC-LDA, first compute the singular value decomposition of Z such that $Z = UDV^\intercal$. Then, project Z onto the top $r < n$ right singular vectors of Z to get the $n \times r$ principal component scores matrix $W = ZV$. We are now in a low-dimensional setting and can proceed by training a traditional LDA classifier to predict Y from W. To use this approach, however, we must estimate the best value of r.

An alternative adaptation of LDA for high-dimensional settings is regularized or ridged LDA. This approach focuses on addressing the singularity of the sample covariance matrix that is caused by having fewer observations than features. While there are many different formulations of covariance matrix regularization, a common approach uses

$$\tilde{\Sigma} = \hat{\Sigma} + \gamma I_p, \tag{4.3}$$

where $\gamma \in [0, \infty)$ is the regularization parameter (*Guo et al.*, 2006; *Hastie and Tibshirani*, 2004). The regularized covariance matrix $\tilde{\Sigma}$ will be invertible and can thus be used in place of $\hat{\Sigma}$ in traditional LDA.

While both PC-LDA and regularized LDA are better suited for high dimensions than traditional LDA, they each depend on the use of a tuning parameter. Selecting the optimal number of principal components for PC-LDA and the optimal regularization parameter for regularized LDA are important, and computationally expensive, steps.

4.2.2 Computational Challenges

There are many aspects of training and evaluating a classifier that can be computationally expensive. As indicated above, this includes tuning model parameters. To select the optimal tuning parameter value for a given classification problem, cross-validation (CV), either k-fold or leave-one-out (LOO), is typically performed. To perform 10-fold CV, for instance, the data are divided into 10 complementary training and hold-out sets. For every value of the tuning parameter under consideration, the classifier is built on each of the 10 training sets and then used to predict the outcomes for the corresponding hold-out sets. The tuning parameter value with the best performance across all 10 hold-out samples is typically selected.

In addition to tuning model parameters, it is also desirable to estimate the predictive performance of the classifier on out-of-sample observations. This is again done via CV. Using the same

CV procedure to both tune model parameters and evaluate model performance, however, will give biased accuracy estimates that are overly optimistic. This results from using the same data to select tuning parameters and evaluate performance. Therefore, nested CV, with tuning parameter selection nested within accuracy estimation, is required to obtain accurate performance estimates. Such a nested procedure, however, substantially increases the number of unique models that must be fit and drastically increases computation time.

Further, an exploratory analysis process might require the entire nested CV procedure to be run more than once. For example, if the class labels are created by discretizing a continuous variable, it may be worthwhile to perform the analysis with different numbers of classes and different cut-off thresholds. Similarly, if the feature matrix has missing values, it could be useful to perform classification after implementing several different data imputation methods. More generally, in any analysis, there will be many modeling choices to explore, so the ability to quickly build and evaluate models is important. Additionally, in this type of analysis, it is crucial to get accurate model performance estimates. While rough estimates of accuracy might suffice in other applications, here we are using model performance to evaluate and directly compare the effectiveness of different data preprocessing steps. For this to be an effective approach, our performance estimates need to be both accurate and reliable. Obtaining those accurate estimates, however, comes at a cost; performing nested CV with such a complex and iterative analysis process requires extended computation times.

4.3 Eliminating Tuning Parameter Selection

Prompted by the tension between accurately estimating model performance and running an analysis in a practical amount of time, we implement a high-dimensional LDA method that does not depend on parameter tuning. This approach begins like PC-LDA with the calculation of principal components. To perform traditional PC-LDA, CV is typically used to determine the optimal number of principal components onto which to project the feature matrix. In this approach, however, we eliminate the need to tune that parameter. Instead, we use the maximum number of principal components, n. We will refer to this approach as nPC-LDA.

With this technique, we perform our classification in n dimensions rather than in p dimensions, using reasoning similar to *Ye and Wang* (2006) and *Ramey et al.* (2017). Specifically, in high-dimensional settings, while the feature space is p-dimensional, we observe only $n < p$ vectors in that space. This leaves $p - n$ dimensions where we do not observe any variation. Further, the p-dimensional sample covariance matrix will have at least $p - n$ null eigenvalues.

It turns out, however, that adding a ridge term to the p-dimensional sample covariance matrix (e.g., $\tilde{\Sigma} = \hat{\Sigma} + \lambda I_p$) is equivalent to adding the same ridge term to the n-dimensional principal

component sample covariance matrix; both approaches will produce the same classifier. This is because the d class sample mean vectors, $\hat{\mu}_k$, are orthogonal to the $p - n$ unused dimensions. Further, the covariance matrix appears in LDA only via the product $\hat{\Sigma}^{-1}\hat{\mu}_k$. Therefore, these uninformative dimensions are irrelevant, and anything we do to ridge the eigenvalues corresponding to those dimensions will not matter (*Ramey et al.*, 2017).

In this high-dimensional classification setting, however, using n principal components still produces a singular covariance matrix due to the estimation of the d class means. Therefore, we must introduce regularization for the covariance matrix. We suggest, however, that there is a natural way to perform this regularization without the need to tune the regularization parameter.

Specifically, we are predicting the $n \times d$ class labels Y from the $n \times n$ predictor matrix $W = ZV$. In this setting, Y is an indicator matrix such that $Y_{ik} = 1$ if observation i is in class k and $Y_{ik} = 0$ otherwise; the row sums of Y are equal to 1 as each observation belongs to only one class. We let $\hat{\mu}_k = \frac{1}{n_k} \sum_{i:Y_{ik}=1} W_i$ be the estimated mean vector for class k and

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i:Y_{ik}=1} (W_i - \hat{\mu}_k)^\intercal (W_i - \hat{\mu}_k)$$

be the estimated covariance matrix for class k. We then define the overall covariance matrix to be

$$\hat{\Sigma} = \sum_{k=1}^{d} \frac{n_k}{n} \hat{\Sigma}_k = \frac{1}{n} W^\intercal R_Y W,$$

where $R_Y = I - Y(Y^\intercal Y)^{-1} Y^\intercal$ is the residual operator of Y. We note that $\hat{\Sigma}$ has rank $n - d$ due to the estimation of the d class means. Therefore, $\hat{\Sigma}$ has d null eigenvalues and is singular. To regularize $\hat{\Sigma}$, and create an invertible modified covariance matrix, we replace the d null eigenvalues with a non-zero value, λ. In this setting, a natural choice is to replace the null eigenvalues with a value similar to the other smallest eigenvalues of the covariance matrix (*Thomaz et al.*, 2006; *Xu et al.*, 2009; *Payne and Gagnon-Bartsch*, 2022). We note that the smallest eigenvalues of the covariance matrix are all equal to σ^2, the error variance introduced in Equation 4.2; while a few eigenvalues are larger, most are exactly equal to σ^2. Therefore, in practice, we let λ be the median of the eigenvalues of $G = ZZ^\intercal$. The resulting regularized covariance matrix is defined as

$$\tilde{\Sigma} = \hat{\Sigma} + \lambda P_{W^{-1}Y},$$

where $P_{W^{-1}Y}$ is the projection operator of $W^{-1}Y$. While similar in form to the regularization introduced in Equation 4.3, this formulation is more targeted. Specifically, the matrix $P_{W^{-1}Y}$ is a projection matrix with d eigenvalues equal to 1 and the remainder equal to 0. Further, $P_{W^{-1}Y}$ is in the null space of $\hat{\Sigma}$. Therefore, the eigenvectors of $P_{W^{-1}Y}$ corresponding to an eigenvalue of 1

span the null space of $\hat{\Sigma}$, which corresponds to the eigenvectors of $\hat{\Sigma}$ with an eigenvalue of 0. The modified covariance matrix $\tilde{\Sigma}$, therefore, has only the d null eigenvalues of $\hat{\Sigma}$ replaced by λ.

Together, this combination of PC-LDA and regularized LDA is the basis of our nPC-LDA classifier. For an out-of-sample observation $\tilde{Z}$, we can write down the expression for $\tilde{Z}\hat{\beta}_1 = (\tilde{Z}\hat{\beta}_1^{(1)}, \ldots, \tilde{Z}\hat{\beta}_1^{(d)})$ as follows,

$$\tilde{Z}\hat{\beta}_1 = \tilde{Z}V\tilde{\Sigma}^{-1}\hat{\mu} = \tilde{Z}V[\frac{1}{n}W^\intercal R_Y W + \lambda P_{W^{-1}Y}]^{-1}(O_Y W)^\intercal, \tag{4.4}$$

where $O_Y = (Y^\intercal Y)^{-1}Y^\intercal$ is the regression operator of Y. Equation 4.4 can be further simplified into an expression that is fast to compute. Specifically, we calculate

$$\tilde{Z}\hat{\beta}_1 = n\tilde{Z}Z^\intercal(ZZ^\intercal)^{-1}[I_n - A(I_{2d} + BA)^{-1}B]O_Y^\intercal, \tag{4.5}$$

where

$$\begin{aligned}
H &= Y(Y^\intercal Y)^{-1/2}, \\
K &= \sqrt{n\lambda}(ZZ^\intercal)^{-1}Y(Y^\intercal(ZZ^\intercal)^{-1}Y)^{-1/2}, \\
A &= \begin{bmatrix} -H & K \end{bmatrix}, \text{ and} \\
B &= \begin{bmatrix} H^\intercal \\ K^\intercal \end{bmatrix}
\end{aligned}$$

(see Appendix D.1 for the derivation). This is the main calculation needed to apply the nPC-LDA classifier to an out-of-sample observation.

Performing nPC-LDA on high-dimensional data in the above manner is fairly computationally inexpensive; we have, for instance, eliminated the need for parameter tuning and the inversion of any $p \times p$ matrices. Performing LOO CV to asses nPC-LDA accuracy, on the other hand, is not as efficient. Specifically, implementing LOO CV requires the inversion of several matrices at each of the n CV iterations, a lengthy task even with a moderate sample size. We address these concerns below.

4.4 Fast, Approximate, and Stable LOO CV for nPC-LDA

The nPC-LDA classifier is efficient to implement on high-dimensional data, but still requires CV to evaluate its performance. To increase the computational efficiency of this task, we have identified and implemented several methods that improve the speed of performing LOO CV for nPC-LDA. These techniques include (1) identifying key quantities that can be stably approximated in the LOO procedure, (2) downdating large matrices to reduce the number of redundant calculations, and (3)

taking advantage of the shared structure of drug screening data.

4.4.1 Stable Cross-Validation Terms

We implement approximations in the LOO CV procedure to improve the computational efficiency of evaluating model performance. To do so, we identify key terms in Equation 4.5 that can be approximated instead of fully recomputed at each CV iteration. Specifically, we consider the term $(I_{2d}+BA)^{-1}BO_Y^\intercal$. The resulting matrix has dimension $2d \times d$; in the binary classification setting, for instance, $(I_{2d}+BA)^{-1}BO_Y^\intercal$ will be a 4×2 matrix. More importantly, the dimensions of this matrix do not grow with n.

This low-dimensionality ensures that $(I_{2d}+BA)^{-1}BO_Y^\intercal$ is essentially unchanged whether calculated with the full data or when holding out the ith observation, i.e., this term is "stable" (Appendix D.1.2). Therefore, we calculate this matrix once, with the full data, and then plug it in during each iteration of the LOO loop. This substantially reduces the number of required computations without hurting our ability to accurately estimate model performance.

4.4.2 Matrix Downdating

We reduce the number of expensive and redundant calculations in LOO cross-validated nPC-LDA. Rather than re-calculating large matrix inverses, minus the i^{th} observation, at each iteration of the CV loop, we instead introduce downdates that are less computationally expensive. For example, consider $Z_i Z_{-i}^\intercal (Z_{-i}Z_{-i}^\intercal)^{-1}$, where Z_i is the i^{th} row of Z and Z_{-i} is Z without the i^{th} row. This matrix appears in the LOO version of the nPC-LDA scores (i.e., let $\tilde{Z} = Z_i$ and $Z = Z_{-i}$ in Equation 4.5). Therefore, this matrix must be calculated for each value of i; however, calculating it separately at each LOO iteration would require inverting an $(n-1) \times (n-1)$ matrix n times. Instead of repeatedly performing this inversion, however, we use the following fact:

$$Z_i Z_{-i}^\intercal (Z_{-i}Z_{-i}^\intercal)^{-1} = \frac{-1}{G_{ii}^{-1}} G_i^{-\intercal} D_i, \tag{4.6}$$

where G_i^{-1} is the i^{th} column and G_{ii}^{-1} is the (i,i) element of $G^{-1} = (ZZ^\intercal)^{-1}$, and D_i is the $n \times n$ identity matrix with the i^{th} column removed (Appendix D.1.1 and *Hastie et al.* (2019)). When the full G^{-1} is known, the right side of this equation is fast and simple to calculate; it only requires subsetting and rescaling G^{-1} at each CV iteration.

In particular, the naive formulation, directly calculating $Z_i Z_{-i}^\intercal (Z_{-i}Z_{-i}^\intercal)^{-1}$ at each LOO iteration, requires a computational complexity of $O(n^2p+n^3)$ at all n iterations, for an overall complexity of $O(n^3p+n^4)$. In contrast, calculating $\frac{-1}{G_{ii}^{-1}} G_i^{-\intercal} D_i$ requires the calculation of $G^{-1} = (ZZ^\intercal)^{-1}$ once, with a complexity of $O(n^2p+n^3)$. This calculation, however, will have already been per-

formed when fitting nPC-LDA to the full data, and it does not need to be repeated. Therefore, the complexity of calculating $Z_i Z_{-i}^\intercal (Z_{-i} Z_{-i}^\intercal)^{-1}$ via the downdate within each CV iteration is reduced to $O(n)$, resulting in an overall complexity of $O(n^2)$. This saves substantial computation time over the original calculation.

4.4.3 Data Structure

We further take advantage of downdating the computationally expensive G and G^{-1} matrices within the setting of large drug screening studies. In these studies, the goal is to predict drug efficacy from cell line genomic information for each of the T drugs under consideration. We must build a separate classifier for each drug. Importantly, however, the only information that differs between drugs is the response vector (how the cells respond to the drug) and the subset of cell lines on which the drug was tested. That is, the cell line genetic information itself is independent of the drug.

This observation suggests a downdating procedure. We calculate G and G^{-1} from the full Z matrix that includes all cell lines in the study. Then, for each drug, we downdate G and G^{-1} based on the subset of cell lines upon which the drug was tested. Specifically, let drug $t \in 1, \ldots, T$ be tested against a subset of n_t cell lines, with the subset denoted $\mathcal{T}_t$. To build a classifier for drug t, we need $Z_{\mathcal{T}_t}$, $G_{\mathcal{T}_t}$, and $G_{\mathcal{T}_t}^{-1}$. The computation time to obtain $G_{\mathcal{T}_t}^{-1}$ directly is $O(n_t^2 p + n_t^3)$. We can, however, use downdating to obtain $G_{\mathcal{T}_t}^{-1}$ in a more computationally efficient manner.

Without loss of generality, assume the cell lines we want to drop appear in rows 1 through n_d, while the cell lines in $\mathcal{T}_t$ appear in rows $n_d + 1$ through n of Z, such that $n_d + n_t = n$. Then we can downdate G^{-1} as follows,

$$G_{\mathcal{T}_t}^{-1} = \left[G^{-1} - G^{-1} U (-I_{2n_d} + V G^{-1} U)^{-1} V G^{-T} \right]_{(-1:n_d, -1:n_d)},$$

where $U_{n \times 2n_d} = (b, a)$, $V_{2n_d \times n} = (a^\intercal, b^\intercal)^\intercal$, $a_{n \times n_d}$ contains the first n_d columns of G, and $b_{n \times n_d}$ is the tall identity matrix augmented by zeroes. After the initial calculation of G and G^{-1}, this downdating procedure reduces the total computational complexity for each subsequent drug to $O(n_t^2 + n^2 n_d)$, when n_t is large. Importantly, this downdating procedure eliminates any calculations depending on the number of features p. If n_t is substantially smaller than n, however, it is more computationally efficient to fully recalculate $G_{\mathcal{T}_t}^{-1}$ from scratch.

4.4.4 The Algorithm

We combine the nPC-LDA classification method introduced in Section 4.3 with the computational improvements to LOO CV outlined in Sections 4.4.1 through 4.4.3. Together, these techniques

form a fast and computationally efficient approximate LOO cross-validated nPC-LDA method that we can apply to high-dimensional data.

As outlined in Algorithm 2, this technique first trains an nPC-LDA classifier on the full data. This is the model that will be used to predict class labels for out-of-sample observations. To assess the performance of that model, we use a fast, approximate, and stable LOO CV approach (FAST-CV). In particular, at each CV iteration, we combine the stable terms that have been calculated from the full data with the downdated matrices that are re-calculated within the CV loop. For each iteration, this produces an approximate nPC-LDA model trained on $n-1$ observations that we use to predict the class label for the held-out sample. The predicted class labels are then used to estimate model performance. Below, we show the improved computation time for this algorithm over existing methods for high-dimensional data. Much of the improvement is due to our approximate LOO CV approach. For a single iteration of our algorithm, the computational complexity of calculating $Z_i\hat{\beta}1$ as in Equation 4.5 is $O(nd^2)$; the computational complexity for the

Algorithm 2: nPC-LDA with FAST-CV

Data: $(Z_{n\times p}, Y_{n\times d})$

Calculate (or downdate):

- $G \leftarrow ZZ^\intercal$
- $G^{-1} \leftarrow (ZZ^\intercal)^{-1}$
- $\lambda \leftarrow$ median of eigenvalues of G
- $\pi \leftarrow$ vector of d class proportions
- $\hat{b}_{stable} \leftarrow (I_{2d} + BA)^{-1}BO_Y^\intercal$ `as defined in Equation 4.5`

Build nPC-LDA on (Z, Y)**:**

- $\hat{b}_1 \leftarrow nG^{-1}(O_Y^\intercal - A\hat{b}_{stable})$ `where` $\hat{\beta}_1 = Z^\intercal\hat{b}_1$ `as in Equation 4.5`
- $\hat{\beta}_0 \leftarrow log(\pi) - \frac{1}{2}\,\text{diag}(O_Y G\hat{b}_1)$

Estimate nPC-LDA performance via FAST-CV:

- **for** i *in* $1, \ldots, n$ **do**
  - **Downdate matrices:**
    - $\hat{b}^{(i)}_{stable} \leftarrow \sqrt{\frac{n-1}{n}}\hat{b}_{stable}$
    - $C^{(i)} \leftarrow \frac{-1}{G^{-1}_{ii}}G_i^{-\intercal}D_i$ `as defined in Equation 4.6`
  - **Calculate non-stable terms from** (Z_{-i}, Y_{-i})**:**
    - $O_Y^{(i)}$ and $A^{(i)}$
  - **Calculate approximate nPC-LDA scores for observation** i**:**
    - $Z_i\hat{\beta}_1^{(i)} \leftarrow (n-1)\left[C^{(i)\intercal}O_Y^{(i)} - C^{(i)}A^{(i)}\hat{b}^{(i)}_{stable}\right]$
    - $\hat{s}_i \leftarrow Z_i\hat{\beta}_1^{(i)} + \hat{\beta}_0$
  - **Assign observation** i **to the class with the largest score.**
- **end**
- **Compare predicted classes to observed classes to estimate model performance.**

entire loop is $O(n^2d^2)$.

4.5 Simulations

We run several simulations to evaluate the performance of nPC-LDA with FAST-CV. In these simulations, we generate data according to two models, "uninformative" and "informative"; these models are modified from the simulation settings in *Payne and Gagnon-Bartsch* (2022). We use the simulated data to compare nPC-LDA with FAST-CV to nPC-LDA with traditional LOO CV, as well as to other LDA-based methods under several CV schemes. We compare both the accuracy of CV error estimates and computation time for each of these approaches. Simulations to evaluate computation time were performed on a 2020 MacBook Pro with a 2 GHz Quad-Core Intel Core i5 processor.

4.5.1 Data Generation

In both the uninformative and informative settings, we generate Y, a binary class vector of dimension n with balanced classes, and Z, an $n \times p$ feature matrix. Under the uninformative model, the feature matrix is Gaussian white noise. We define $Z^{(u)}$ as follows:

$$Z^{(u)}_{n\times p} = \varepsilon_{n\times p},$$

where $\varepsilon \sim \mathcal{N}(0, I_p)$. For the informative model, we let there be $\ell = 3$ latent factors, denoted by L, as introduced in Equation 4.2. These factors are correlated with the class labels and provide useful information for classification. We generate $Z^{(i)}$ as follows:

$$Z^{(i)}_{n\times p} = Z^{(u)}_{n\times p} + L_{n\times \ell}\alpha_{\ell\times p},$$

where $L_{n\times\ell} \sim \mathcal{N}(Y_{n\times 1}\eta_{1\times\ell}, I_\ell)$, $\eta = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$, and $\alpha_{\ell\times p} \sim \mathcal{N}(0, \frac{6}{\sqrt{p}}I_p)$. In this setting the features are informative for predicting class labels.

In these simulations, we let n range from 6 to 1000 and consider p = 20,000, 100,000, and 500,000. For each combination of n and p we generate many simulated data sets and average accuracy rates and computation time across replicates. To make the computations feasible, we vary the number of replicates with n (Appendix D.3).

4.5.2 Methods

We test the speed and model performance estimation capabilities of nPC-LDA with LOO CV and nPC-LDA with FAST-CV on the simulated data. We compare these methods to PC-LDA

with a fixed number of PCs ($r = 3$ and $r = 50$). We perform PC-LDA in R via the `svd()` and `MASS::lda()` functions. We do not use CV to select the optimal r due to computation constraints, as discussed below. Instead, we consider a small value of r (equal to the number of low-dimensional factors in the informative model) and a large value of r.

In these simulations, we use two different CV schemes, LOO and 10-fold, to get CV error estimates for PC-LDA. For both LOO and 10-fold CV, we perform "correct" CV, where the principal components are re-computed within each CV iteration, and "incorrect" CV, where the principal components are only computed on the full data and then subsetted within each CV iteration. The "incorrect" version is implemented as a fast alternative to the "correct" method. When evaluating the accuracy of LOO CV, we were not always able to perform the "correct" CV version as it is too computationally inefficient for the size of our simulated data (Figure 4.2). Additionally, we cannot perform PC-LDA with $r = 50$ on our simulated sample sizes of $n = 6, 10$, and 20. We also did not perform 10-fold CV on our simulated sample sizes of $n = 6$ and 10.

4.5.3 Results

In these simulations, most of the tested methods are able to reliably recover theoretical accuracy rates via CV; nPC-LDA with FAST-CV, however, vastly outperforms the other methods in terms of computation time. We begin our analysis by investigating the performance of the `MASS::lda()` implementation of PC-LDA and the ability of various CV methods to recover theoretical accuracy rates. When performing 10-fold CV in the "correct" manner, we obtain accuracy estimates that are very close to the theoretical accuracy rates. This is true both when PC-LDA is performed with 3 PCs and with 50 PCs (Figure 4.1). Performing CV in the "incorrect" way, however, does not always allow us to accurately recover those true accuracy rates. In particular, in the informative setting, "incorrect" CV substantially underestimates theoretical accuracy when $r = 50$. The underestimation is present for most tested sample sizes (Figure 4.1b). Performing "incorrect" CV for PC-LDA with $r = 3$, on the other hand, allows for good recovery of theoretical accuracy rates at large sample sizes, but overestimates theoretical accuracy at moderate and small sample size. We hypothesize that the overestimation of "incorrect" CV for PC-LDA with $r = 3$ is likely because the generated data actually have three (latent) informative factors. With this setup, "incorrect" CV is using the most informative subset of the full data to build the model on each CV training set. For a given CV iteration, data from the test set is incorporated into the fitted model, producing overly optimistic performance estimates. Therefore, performing CV, whether LOO or 10-fold, in the "correct" manner is necessary for guaranteeing reliable estimates of model performance.

There is a trade-off, however, between accurate error estimation and computation time. Performing "correct" LOO CV, for instance, takes n times as long as performing "incorrect" LOO CV

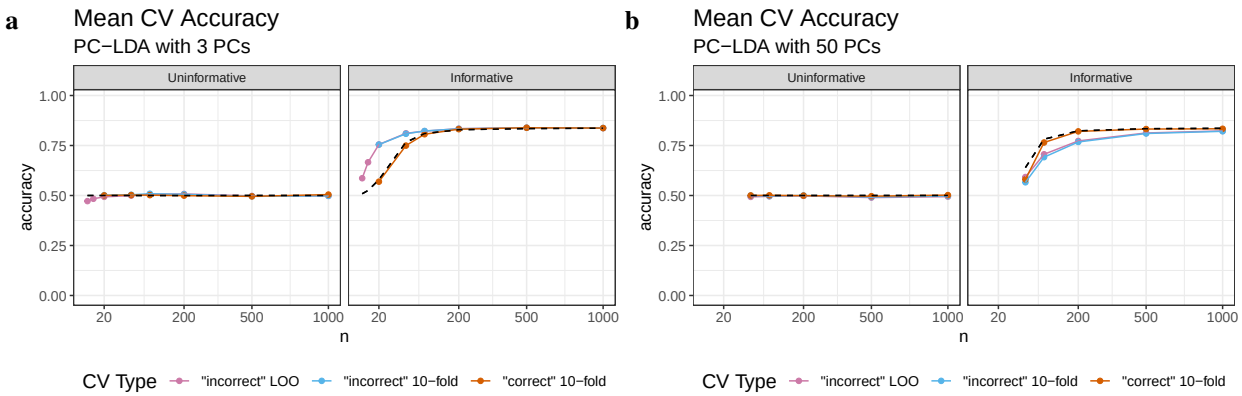

Figure 4.1: Mean CV accuracies for a PC-LDA classifier with **(a)** $r = 3$ and **(b)** $r = 50$ principal components and $p = 20{,}000$ features. Sample size (n) is displayed on a square-root scale. The black dashed lines indicate the theoretical classification accuracy rates for a PC-LDA classifier with r principal components built on the generated data. "Incorrect" CV involves only calculating the principal components on the full data, while "correct" CV involves calculating the principal components for each individual CV training set. Neither PC-LDA with 50 PCs nor 10-fold CV can be run on small sample sizes.

(e.g., 18.5 seconds vs. 0.3 seconds for $n = 60$ and $p = 20{,}000$; Figure 4.2). As n grows, it quickly becomes infeasible to perform "correct" LOO CV. Therefore, 10-fold CV is a commonly used alternative to the LOO approach. When performing 10-fold CV, the "incorrect" method is about 8 times faster than the "correct" method; this difference can be substantial, but does not depend on n.

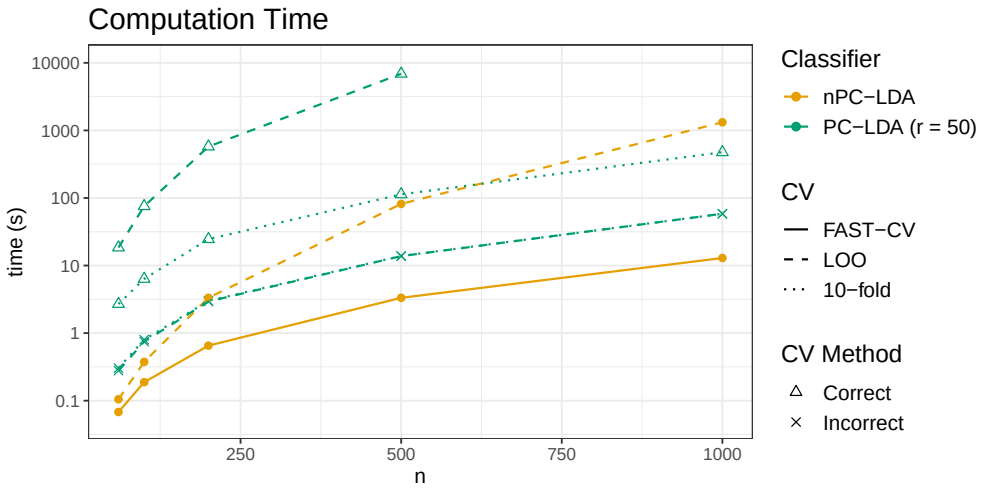

Figure 4.2: Median computation times for $p = 20{,}000$. Time (in seconds) is displayed on a $\log_{10}$ scale. Our implementation of nPC-LDA with FAST-CV has the fastest computation time across all values of n; even at $n = 1000$, it takes just over 10 seconds to complete. Note: we did not run PC-LDA with 50 PCs and LOO CV performed in the "correct" way for $n = 1000$ due to computation constraints.

Lengthy computation times are further extended when using CV to select the optimal number of PCs onto which to project the feature matrix (rather than just selecting a fixed r, as was done in these simulations). Performing nested 10-fold CV, for instance, will take 10 times as long as the "correct" 10-fold CV procedure displayed in Figure 4.2.

Our implementation of nPC-LDA with FAST-CV, however, performs well when considering both speed and accuracy. Across all tested values of n, nPC-LDA with FAST-CV has the fastest computation time when compared to nPC-LDA with LOO CV and all forms of CV for PC-LDA with 50 PCs. It takes just 11 seconds to run nPC-LDA and obtain a FAST-CV accuracy rate when $n = 1000$ and $p = 20{,}000$. Further, the FAST-CV accuracy rates are reliable. In both simulation settings, the approximate LOO CV accuracy of performing nPC-LDA with FAST-CV is almost identical to the classical LOO CV accuracy, both of which closely match the theoretical accuracy rates for nPC-LDA (Figure 4.3). This indicates that the approximations we make in FAST-CV are effective at reducing computation time while not seriously hurting our ability to estimate model performance. Our approximate procedure is fast and recovers the theoretical accuracy rates well.

Overall, these simulations show the strengths of our nPC-LDA classifier and FAST-CV algorithm. Our computation speeds are much faster than traditional methods, even those that sacrifice accuracy to be more computationally efficient. The FAST-CV procedure also recovers true accuracy rates at the level of a full CV procedure. The other tested methods are unable to perform as well on both metrics.

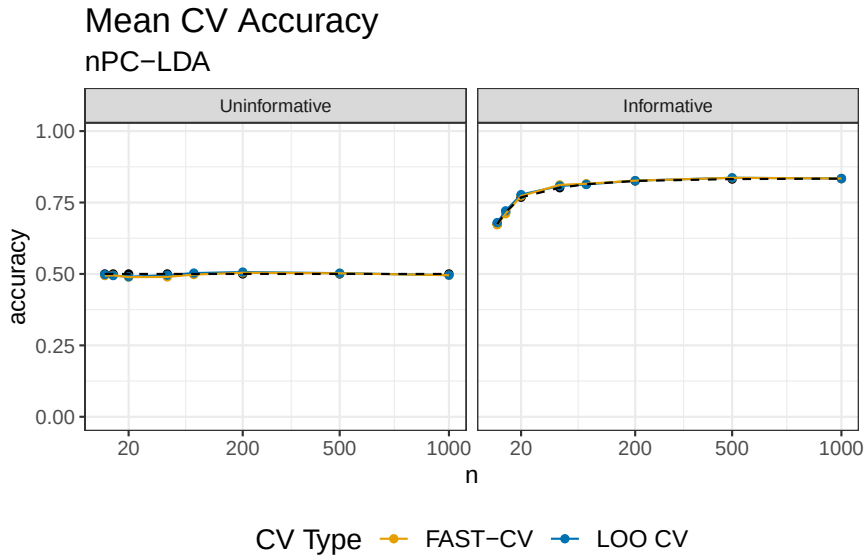

Figure 4.3: Mean CV accuracies for nPC-LDA obtained via LOO CV and FAST-CV for $p = 20{,}000$. Sample size (n) is displayed on a square-root scale. The black dashed lines indicate the theoretical classification accuracy rates for an nPC-LDA classifier built on the generated data.

4.6 Evaluation on Pharmacogenomic Data

We evaluate the performance of our nPC-LDA classifier with FAST-CV on the large-scale and publicly available pharmacogenomic data from the GDSC study (*Yang et al.*, 2013). This database contains genomic information about more than 1000 cancer cell lines in addition to drug efficacy data for almost 200 potential anti-cancer drugs tested on those cell lines. As in Chapter 3, we use release 8.2 of the GDSC2 drug efficacy data in this application.

We first consider predicting cell line tissue type from gene expression levels. GDSC contains several different tissue and tumor type labels for each cell line; we chose to focus on two. The tissue type label separates 968 cell lines into 19 distinct classes such as non-small cell lung cancer, urogenital system cancers, leukemia, etc. The tumor histology label separates 932 cell lines into 11 distinct classes including carcinoma, lymphoid neoplasm, and glioma (we removed 18 histologies that had fewer than 5 cell lines each). These class labels constitute a multiclass classification setting ($k > 2$) for both tissue type ($k = 19$) and histology ($k = 11$). Further, the histology data contains a serious class imbalance: around 60% of cell lines are labeled carcinomas, while only 40% of cell lines are left to the other 10 classes (see Appendix D.4.1for more details). GDSC also provides gene expression data for each cell line. This contains expression levels for 17,737 genes.

We compared the performance of the nPC-LDA algorithm to off-the-shelf random forest and LDA methods. Initially, we estimated model performance via classical LOO CV for nPC-LDA and `MASS:lda`, via classical 10-fold CV for maximum uncertainty LDA (MLDA), and via out-of-bag (OOB) voting for random forests. In terms of estimated classification accuracy, F_1-score, and Matthews correlation coefficient (MCC), the nPC-LDA algorithm performs better than the other tested methods for tissue type classification, but slightly worse than MLDA for histology classification (Table 4.1). Notably, the `MASS` package implementation of high-dimensional LDA

	Tissue Type			Histology		
Model	Accuracy	F_1-score	MCC	Accuracy	F_1-score	MCC
nPC-LDA	0.800	0.788	0.785	0.959	0.853	0.932
Random Forest	0.693	–	0.672	0.901	–	0.831
`MASS::lda()`	0.054	0.052	0.001	0.093	0.060	0.003
MLDA	0.768	0.724	0.751	0.982	0.958	0.970

Table 4.1: Estimated model performance for predicting cell line tissue type and histology from gene expression levels. Model performance was estimated via LOO CV for nPC-LDA and `MASS::lda()`, via 10-fold CV for MLDA, and via OOB voting for random forests. nPC-LDA performs better than random forests and `MASS::lda()`, but comparably with MLDA. The random forests do not have a meaningful F_1-score because precision is undefined for two tissue types and four histologies. Random forests were implemented via the `randomForest::randomForest()` function in R; LDA was implemented via the `MASS::lda()` function in R; MLDA was implemented via the `HiDimDA::Mlda()` function in R.

	Tissue Type				Histology			
CV Method	Accuracy	F_1-score	MCC	Time (s)	Accuracy	F_1-score	MCC	Time (s)
FAST-CV	0.794	0.785	0.780	16.5	0.957	0.854	0.928	15.2
LOO CV	0.800	0.788	0.785	1244.4	0.959	0.853	0.932	1097.7

Table 4.2: Estimated model performance and computation speed for an nPC-LDA classifier predicting cell line tissue type and histology from gene expression levels. Model performance was estimated via FAST-CV and LOO CV. The two methods produce quite similar estimates for all model performance metrics. The FAST-CV algorithm, however, has computation times around two orders of magnitude faster than classical LOO CV.

in R performs no better than random guessing (i.e., $\frac{1}{19} = 0.053$ and $\frac{1}{11} = 0.091$, matching the `MASS:lda()` estimated accuracies). Our nPC-LDA method, however, performs substantially better on both classification problems.

Additionally, as shown in simulations, the model performance of nPC-LDA estimated via FAST-CV is almost identical to the model performance of nPC-LDA estimated via classical LOO CV (Table 4.2). Further, nPC-LDA with FAST-CV achieved this performance with computation speeds more than 70 times faster than nPC-LDA with LOO CV (more than 20 times faster than `randomForest::randomForest()` and more than 40 times faster than `MASS::lda()` and `HiDimDA::Mlda()`; Table D.3). Overall, nPC-LDA implemented with FAST-CV performs quite well, both in terms of model performance and computation speed, for both multiclass data and data with substantial class imbalances (see Appendix D.4.1 for more details).

We also used the GDSC data to predict drug efficacy from cancer cell line gene expression levels. GDSC contains drug efficacy data for 198 potential anti-cancer drugs tested on 805 cancer cell lines (*Yang et al.*, 2013). For each drug-cell line combination, drug efficacy is summarized by the area under the dose-response curve (AUC), where small AUC values indicate an effective drug (equivalently, a sensitive cell line). To create a binary response variable suitable for classification, we discretized the estimated AUC values into calls of "sensitive" (small AUCs) and "insensitive" (large AUCs; Appendix D.4.2). For each drug, we further down-sampled observations from the majority class to impose class balance and improve the interpretability of model performance estimates; this resulted in 147 drugs with sufficient cell lines. The average drug was tested on 238 cell lines (minimum 20, maximum 792). These drug sensitivity calls form the binary response vector in our classification problem. The feature matrix in this analysis is the same gene expression data used for the tissue type classification above. We consider other types of genetic information, including methylation levels, in Section 4.7.

The relationship between drug efficacy and cell line gene expression levels will differ across drugs. Therefore, a separate classifier must be built for each compound under consideration; we must fit 147 separate cross-validated classifiers in this analysis.

We compared the performance of nPC-LDA with FAST-CV to that of an off-the shelf random forest classifier. We first ran nPC-LDA with FAST-CV on the 147 drugs tested on sufficient cell lines, producing approximate LOO CV accuracies for all drugs in about 50 seconds. To achieve this computation speed, we performed the analysis in two steps. In the first step, we used Z, the full predictor matrix with dimension 805×17737, to calculate $G = ZZ^\intercal$, $G^{-1} = (ZZ^\intercal)^{-1}$, and the median of the eigenvalues of G (as the regularization parameter λ) in just 8 seconds. We then used these precomputed quantities to build the 147 separate classifiers. For each drug, our implementation of nPC-LDA with FAST-CV downdated G and G^{-1} to the subset of cell lines on which that drug was tested and then proceeded with classification. In total, this process calculated approximate LOO CV error estimates for all 147 drugs in less than 1 minute, or in about 0.33 seconds per classifier.

In comparison, we built random forest classifiers for just 15 randomly selected drugs. This analysis produced OOB accuracy estimates for those 15 drugs in 22 minutes, or in almost 90 seconds per classifier. At this rate, it would take more than 3.5 hours to build and evaluate all 147 random forest classifiers. The nPC-LDA algorithm, implemented with FAST-CV, clearly outperforms random forests in terms of computation time.

Importantly, the drastically improved computation time of nPC-LDA with FAST-CV does not hurt classifier performance. For the 15 drugs analyzed by both methods, model performance for nPC-LDA, estimated via approximate LOO CV, is comparable to model performance for random forests, estimated via OOB voting. This similarity holds for accuracy, F_1-score, and MCC (Figure

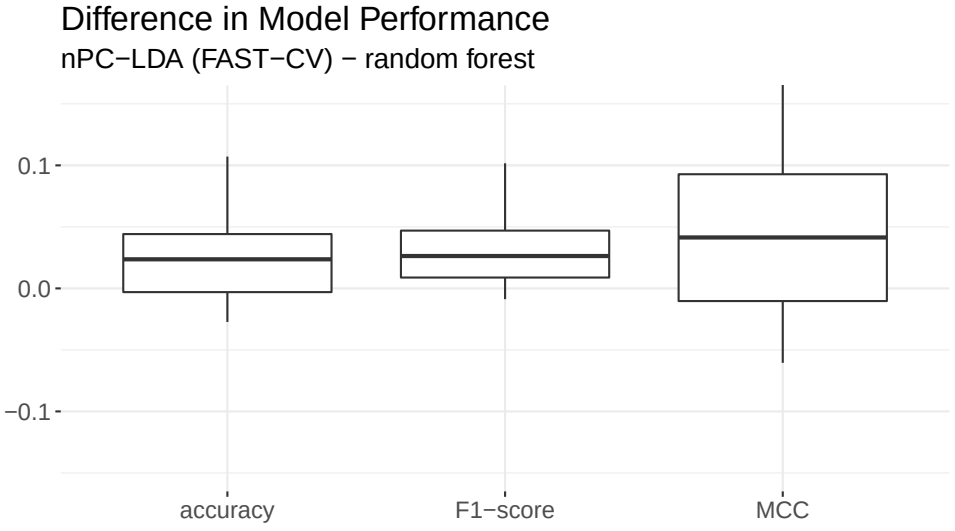

Figure 4.4: Difference in model performance between nPC-LDA with FAST-CV and random forests for 15 randomly selected drugs. The two models have comparable accuracy, F_1-score, and MCC.

4.4), suggesting that not only does nPC-LDA with FAST-CV have incredibly fast computation time, but it also does not hurt classification quality on several common performance metrics.

4.7 Application to Pharmacogenomic Data

We now take advantage of the speed and performance of nPC-LDA with FAST-CV to examine the effects of various modeling decisions on classification quality. For instance, in Section 4.6, we discretized the GDSC-provided AUC estimates into binary calls of "sensitive" and "insensitive" for our classification response vector. There is not, however, a clear and biologically-meaningful way to select the threshold between the two classes. We address this uncertainty by using nPC-LDA with FAST-CV to quickly compare classifier performance for many different potential class thresholds. We also move beyond simple binary classification to consider more complicated classification settings. Further, as the GDSC study contains cell lines from many different types of cancer, we compare classifier performance, and the best class thresholds, between different tissue types. At this stage of the analysis, we continue to focus on predicting drug efficacy, as estimated by the GDSC-provided AUC estimates, from cell line gene expression levels.

In the simple binary classification setting, we look at how model performance varies when we vary the class threshold, m. We let m range from 0.4 to 0.9, labeling cell lines with an AUC below m as "sensitive" and cell lines with an AUC above m as "insensitive". In this analysis, we consider $n = 32$ drugs with sufficient cell lines in each class at all tested thresholds (Appendix D.4.3). Across this range of thresholds, we observe that smaller values of m tend to produce better model performance than larger values of m (Figures 4.5 and D.6; Table D.4). In particular, we observe the best simple binary classification performance at a threshold of around $m = 0.45$ (median accuracy of 0.82; median F_1-score of 0.82; median MCC of 0.63). In general, at smaller threshold values, the fitted nPC-LDA models are better able to discriminate between "sensitive" and "insensitive" cell lines.

In performing this exploratory analysis, we had to fit and evaluate separate classifiers for 32 drugs at 51 different thresholds; this requires the fitting and evaluation of a total of 1,632 models. Using nPC-LDA and FAST-CV, we were able to do this in less than 10 minutes on a personal laptop. Without such speed, exploring the performance of simple binary classification, and finding the best class threshold, may not have been feasible. Therefore, nPC-LDA with FAST-CV is instrumental in improving our understanding of sensitive and insensitive cell lines.

Such simple binary classification, however, depends on the idea that a cell line with an AUC just below the threshold is sensitive while a cell line with an AUC just above the threshold is insensitive. In reality, we do not expect there to be such a definitive division between the two classes, especially when we know the data contain widespread errors and noise. Therefore, we

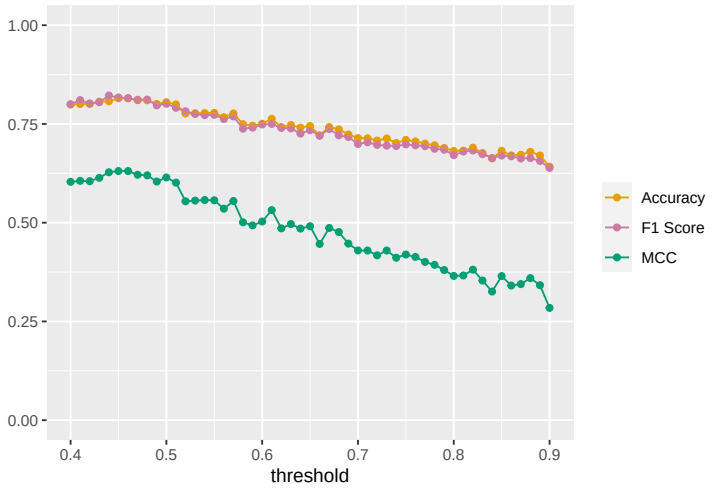

Figure 4.5: Median nPC-LDA performance (accuracy, F_1-score, and MCC) estimated via FAST-CV for binary classification as the binary class threshold varies from 0.4 to 0.9. Performance tends to increase as the class threshold decreases. At each threshold, we performed classification for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds. The median is taken across these drugs.

also consider binary classification where cell lines with moderate AUC values are discarded. More concretely, for two thresholds m_1 and m_2 such that $m_1 < m_2$, we construct our response vector by assigning AUCs $< m_1$ to the "sensitive" class and AUCs $> m_2$ to the "insensitive" class. AUCs between m_1 and m_2 are dropped.

To evaluate model performance when the response vector is constructed with this discretization, we build nPC-LDA classifiers for all combinations of lower threshold m_1 between 0.4 and 0.85 and upper threshold m_2 between 0.45 and 0.9 (Figures 4.6 and D.7). As with the simple binary case, there is better model performance at smaller values of m_1. In particular, we find the best median accuracy and MCC when we let $m_1 = 0.4$ and $m_2 = 0.6$ (median accuracy of 0.87 and median MCC of 0.74; the best median F_1-score is achieved at $m_1 = 0.4$ and $m_2 = 0.65$). Here we are able to fit and evaluate 1,760 nPC-LDA models with FAST-CV in just over five minutes.

Further, we consider how model performance, and the best values of m_1 and m_2, vary across different types of cancer. Specifically, we perform a separate classification analysis for each cancer cell line tissue type. Table 4.3 shows not only how the best values of m_1 and m_2 vary across tissue types, but also how they vary across three widely tested drugs (camptothecin, 5-fluorouracil, and taselisib). In particular, we see that the behavior of leukemia and lymphoma cell lines, two categories of blood cancer, are more similar to each other than to the other tested cancer types. Leukemia and lymphoma have lower selected values of m_1 and m_2 than the other tissue types for camptothecin and are the only two tissue types against which 5-fluorouracil has any effect. This indicates that performing separate analyses for different tissue types and for different classes of

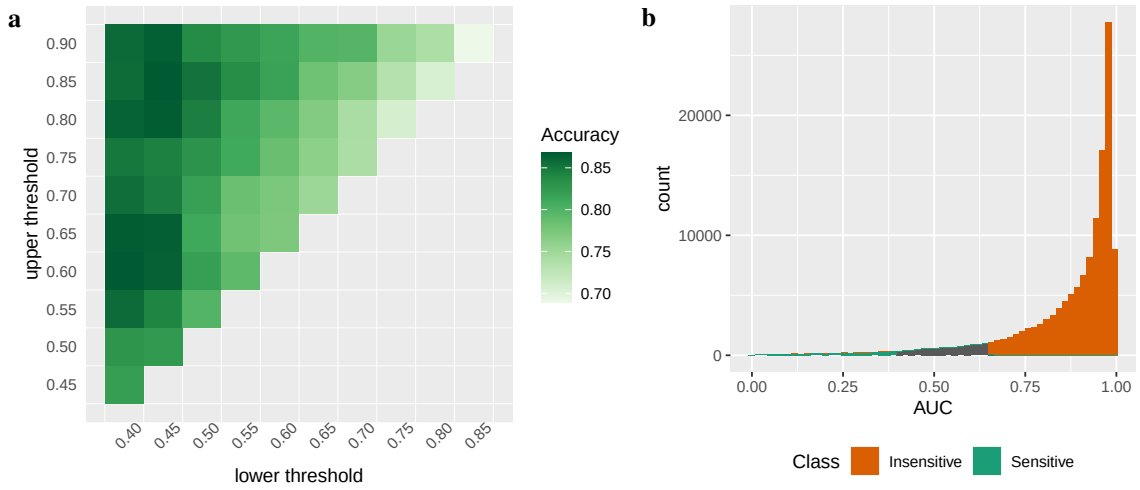

Figure 4.6: Binary classification where moderate AUC values are dropped. **(a)** Median nPC-LDA accuracy estimated via FAST-CV for binary classification where AUC values between the lower threshold (m_1) and the upper threshold (m_2) are discarded. Accuracy tends to be larger for smaller values of m_1. At each set of thresholds, we performed classification for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds. The median is taken across these drugs. **(b)** Histogram of GDSC AUC values colored by class label. Class labels are assigned based on the values of m_1 and m_2 that give the highest median accuracy: $m_1 = 0.4$ and $m_2 = 0.6$.

drugs may improve the quality of classifier performance. It will also, however, increase computation time and resources necessitating the use of nPC-LDA with FAST-CV.

Finally, we introduce different types of genomic data into this classification problem. We expand our set of predictors to include methylation data in addition to gene expression levels. In the GDSC study, there are 725 cell lines for which both gene expression data (17,737 genes) and methylation data (476,559 sites) are available. After concatenating these datasets, our predictor matrix now has dimension $725 \times 494{,}296$.

In this analysis, however, we find that adding methylation data to the gene expression data does

	Camptothecin			5-Fluorouracil			Taselisib		
Tissue Type	Accuracy	m_1	m_2	Accuracy	m_1	m_2	Accuracy	m_1	m_2
urogenital system	0.766	0.8	0.9	–	–	–	0.773	0.7	0.85
NSCLC	0.618	0.8	0.9	–	–	–	0.769	0.75	0.85
leukemia	0.8	0.65	0.75	0.85	0.75	0.85	0.7	0.65	0.7
aero-digestive tract	0.792	0.85	0.9	–	–	–	0.733	0.7	0.85
lymphoma	0.727	0.6	0.8	0.885	0.7	0. 85	0.818	0.7	0.8
breast	–	–	–	–	–	–	0.682	0.8	0.9

Table 4.3: Thresholds corresponding to best model performance for the six largest tissue types and the three most widely used drugs. The "−−" entry indicates there were not sufficient cell lines with AUC values smaller than 0.85 to perform this analysis.

not substantially improve classification accuracy, in general. On its own, methylation data has decent predictive power, producing model performance that is only slightly lower than the performance we observed with gene expression data (Table 4.4). Model performance slightly increases when we consider both methylation and gene expression data together, producing performance that is quite similar to the expression-only analysis. This indicates that the methylation data is not providing a substantial amount of new predictive information.

We are able to explore the influence of the methylation data and come to this conclusion be-

		Gene Expression			Methylation			Both		
m_1	m_2	Accuracy	F_1-score	MCC	Accuracy	F_1-score	MCC	Accuracy	F_1-score	MCC
0.40	0.45	0.820	0.820	0.643	0.800	0.800	0.600	0.821	0.827	0.643
0.40	0.50	0.828	0.829	0.658	0.812	0.808	0.625	0.824	0.828	0.655
0.40	0.55	0.857	0.851	0.715	0.808	0.805	0.616	0.812	0.812	0.630
0.40	0.60	**0.868**	0.863	**0.742**	0.833	0.828	0.668	0.836	0.843	0.673
0.40	0.65	0.866	**0.870**	0.734	0.844	0.839	0.689	0.844	0.839	0.689
0.40	0.70	0.855	0.857	0.719	0.850	**0.851**	0.704	**0.875**	**0.875**	**0.750**
0.40	0.75	0.849	0.845	0.702	**0.857**	0.849	**0.722**	0.844	0.839	0.689
0.40	0.80	0.862	0.866	0.735	0.852	0.848	0.707	0.857	0.857	0.742
0.40	0.85	0.856	0.859	0.714	0.833	0.837	0.676	0.861	0.857	0.728
0.40	0.90	0.858	0.851	0.722	0.833	0.833	0.671	**0.875**	**0.875**	**0.750**
0.45	0.50	0.824	0.830	0.650	0.824	0.824	0.647	0.825	0.829	0.651
0.45	0.55	0.841	0.839	0.684	0.825	0.825	0.656	0.828	0.844	0.670
0.45	0.60	0.862	0.859	0.725	0.845	0.849	0.695	0.840	0.846	0.682
0.45	0.65	0.864	0.861	0.734	0.824	0.816	0.647	0.826	0.829	0.659
0.45	0.70	0.847	0.841	0.695	0.849	0.850	0.699	0.833	0.837	0.681
0.45	0.75	0.843	0.854	0.694	0.853	0.848	0.707	0.845	0.844	0.691
0.45	0.80	0.866	0.866	0.738	0.845	0.843	0.691	0.844	0.840	0.700
0.45	0.85	**0.868**	0.869	0.737	0.826	0.820	0.651	0.845	0.840	0.693
0.45	0.90	0.865	0.867	0.734	0.843	**0.851**	0.688	0.845	0.833	0.693
0.50	0.55	0.798	0.794	0.600	0.810	0.800	0.622	0.829	0.821	0.660
0.50	0.60	0.819	0.815	0.642	0.828	0.819	0.656	0.818	0.816	0.640
0.50	0.65	0.810	0.800	0.623	0.811	0.803	0.623	0.810	0.811	0.624
0.50	0.70	0.818	0.816	0.645	0.824	0.822	0.651	0.833	0.833	0.667
0.50	0.75	0.831	0.829	0.665	0.817	0.821	0.646	0.816	0.821	0.646
0.50	0.80	0.845	0.838	0.691	0.803	0.800	0.616	0.816	0.808	0.640
0.50	0.85	0.852	0.843	0.707	0.833	0.836	0.668	0.843	0.841	0.685
0.50	0.90	0.837	0.836	0.675	0.828	0.821	0.657	0.853	0.841	0.712
0.55	0.60	0.791	0.790	0.586	0.775	0.776	0.564	0.778	0.785	0.561
0.55	0.65	0.780	0.774	0.562	0.786	0.780	0.573	0.800	0.795	0.601
0.55	0.70	0.784	0.786	0.570	0.786	0.776	0.574	0.788	0.782	0.592
0.55	0.75	0.809	0.801	0.621	0.804	0.798	0.611	0.804	0.792	0.625
0.55	0.80	0.812	0.805	0.625	0.807	0.795	0.621	0.811	0.797	0.622
0.55	0.85	0.835	0.830	0.673	0.821	0.808	0.647	0.830	0.820	0.660
0.55	0.90	0.826	0.819	0.654	0.817	0.804	0.638	0.826	0.820	0.666

Table 4.4: Median model performance for nPC-LDA estimated via FAST-CV for predicting binary drug efficacy from gene expression data, from methylation data, and from the concatenation of both. Class labels were assigned by labeling AUC values less than m_1 as "sensitive" and greater than m_2 as "insensitive". Bolded values indicate the best model performance for each predictor set and performance metric.

cause of the computational efficiency of nPC-LDA with FAST-CV. It takes less than 3 minutes to calculate G, G^{-1}, and λ for our 725 $\times$ 494,296-dimensional feature matrix. Once these quantities are calculated, we can then rapidly fit a large number of classifiers with widely varying response vectors. For instance, we fit, and evaluate the performance of, 1,705 classifiers (corresponding to $n = 31$ drugs and 55 sets of thresholds) in about 4.5 minutes.

The data exploration done in this section represents only a small portion of the exploration and analysis that can be performed with these data. Researchers might have other methods of forming the classification response vector, might bring in different genetic information to use as predictors, such as copy number variation or mutation data, or might try other forms of predictor preprocessing, including missing value imputation and predictor transformations. Regardless of the modeling decisions to be explored, the speed and quality of nPC-LDA with FAST-CV provides huge benefits to performing exploratory analysis on large-scale and high-dimensional data.

4.8 Discussion

In the development of the nPC-LDA classifier and our fast, approximate, and stable LOO CV approach, computational efficiency was the main priority. While many classical regression and classification methods have been successfully adapted to high-dimensions, including LDA-based methods, they often require an impractical amount of computation time, especially when n and p are both large. For example, the introduction of a regularization or penalty term, or the use of dimensionality reduction techniques, can be too time- and memory-intensive to complete on a personal computer, particularly when paired with error estimation via nested CV.

In our nPC-LDA method with FAST-CV, however, we adapt LDA to high-dimensional data with an emphasis on fast model building and evaluation. Specifically, by combining covariance matrix regularization with principal components-based dimensionality reduction, we can suggest a natural regularization parameter that eliminates the need for tuning. Further, we forgo exact CV in favor of a more computationally efficient approximate version. We carefully evaluated the R implementation of both nPC-LDA and FAST-CV to speed optimize the code. We offer implementations of both nPC-LDA with LOO CV and nPC-LDA with FAST-CV in the `fastLDA` package.

Further, in our implementation of nPC-LDA with FAST-CV, we prioritized the ability to handle complex and iterative workflows. Beyond using a computationally efficient classifier, we also achieved this via the separability of necessary, but expensive, computations. In particular, nPC-LDA requires the calculation of $G = ZZ^\intercal$, $G^{-1} = (ZZ^\intercal)^{-1}$, and the regularization parameter $\lambda =$ the median of the eigenvalues of G. These three calculations are the most computationally expensive aspect of fitting an nPC-LDA classifier, by far. Once they have been calculated, however,

a new model can be built almost instantaneously to predict any response vector from the feature matrix Z, or from a subset of the samples in Z. Therefore, to improve performance, the user can pre-calculate G, G^{-1}, and λ, providing them as inputs for future models depending on Z. The nPC-LDA algorithm will downdate these quantities rather than re-calculating from scratch, saving computation time. This feature is ideal, for instance, for an iterative workflow that examines classifier performance for different response vectors and the same set of features.

Finally, while our development of both nPC-LDA and FAST-CV was motivated by the challenges of analyzing large-scale drug screening studies, these techniques are more widely applicable. In general, the nPC-LDA classifier, both with LOO CV and FAST-CV, is not tailored to drug screening data. While our techniques can particularly improve computation time via downdating when several hundred classifiers must be fit to similar feature matrices, this is not a requirement for nPC-LDA or FAST-CV to be effective. Overall, if a classifier must be built for a high-dimensional dataset, our algorithm will efficiently fit an nPC-LDA model and obtain accurate approximate LOO CV performance estimates.

CHAPTER 5

Discussion and Future Work

This dissertation has discussed several challenges associated with collecting, processing, and analyzing data from large-scale and high-throughput drug screening studies. Such challenges include the presence of systematic technical errors, the confounding between errors and biological signal, the use of non-optimal experimental designs, and the computational complexity of analyzing high-dimensional data. Each chapter in this dissertation has focused on highlighting and addressing the challenges associated with a different stage in the experimental process.

We begin with data collection and exploration. Chapter 2 makes clear the need to deeply investigate raw drug screening data for the presence of systematic errors. While we focus on the errors in GDSC and CCLE in this dissertation, it is likely that the same errors, as well as new ones, exist in the data collected from large biological experiments more broadly. Deep data exploration is always a necessary first step. Additionally, in this chapter, we identify aspects of the experimental design of drug screening studies that can be improved to aid in the mitigation of such errors in future studies. Implementing these design techniques, such as plate randomization, consistent replication, and the release of full experimental data, will increase the range of statistical methods that can be applied to future drug screening data.

While we advocate for more intentional experimental designs in future drug screening studies, we also acknowledge the large amounts of drug screening data that already exist. Therefore, Chapter 3 focuses on mitigating the effects of technical errors in existing datasets. In particular, we introduce a data processing technique that aims to handle the errors outlined in Chapter 2. We frame data normalization as counterfactual estimation and carefully apply this framework to the GDSC drug screening data through a tailored normalization approach. In other words, we developed a normalization technique that is able to handle the non-optimal plate designs, incredible frequency of missing data, and lack of standardization in plate layouts that GDSC contains. While our normalization method improves upon existing methods, we continue to advocate for the implementation of the experimental design suggestions from Chapter 2. Their use in future studies will make a wider range of statistical tools feasible for normalization.

Finally, in Chapter 4, we move from data collection, cleaning, and processing into data analysis. The nPC-LDA with FAST-CV algorithm we developed was motivated by the computational difficulties of analyzing drug screening data, but its utility is not limited to that setting; rather, it is appropriate for any high-dimensional application. The FAST-CV method does, however, handle a unique feature of drug screening studies particularly well. The effectiveness of each drug in the study needs to be separately modeled; this provides both challenges and opportunities for building and evaluating classifiers. On one hand, this takes the challenges of any high-dimensional analysis and multiplies it by several hundred drugs. On the other hand, we mitigate the adverse effects of fitting hundreds of models by downdating large matrix calculations for each drug. This feature is relevant for any application where the same feature matrix is used across many classifiers.

Together, the chapters of this dissertation provide opportunities for improving the entire workflow of analyzing drug screening studies, as well as other high-throughout biological experiments. As extensively discussed, however, large-scale drug screening studies are incredibly complex. Therefore, there are several areas where this research can be extended.

For instance, there are opportunities for more work to be done on understanding and identifying technical variation in newly produced drug screening data. One strategy involves creating a set of error detection metrics that give quality scores for each microplate or each drug-cell line combination in the study. We have done preliminary work to create metrics for identifying such errors as checkerboard pattern, spatial gradients, and extreme outliers. Each metric quantifies the extent to which one type of error exists on a given plate. Creating valid scores will allow us to eliminate low quality drug screening data from a future analysis. Many of the difficulties of this goal are similar to the challenges outlined in Chapters 2 and 3. Specifically, there are many complex plate layouts used in drug screening studies and there is no standardization in layouts across (or even within) experiments. Therefore, these error detection metrics will either have to be extremely flexible or tailored to a specific study. Further, most plates in drug screening studies contain more than one type of technical error. These metrics need to be able to detect one type of error in the presence of many others. This is a challenging problem. The development of the nPC-LDA with FAST-CV algorithm, however, provides us with a fast validation method. If our metrics are effective, we expect the remaining data to be high quality and therefore to perform better in drug efficacy classification.

Additionally, we are interested in pursuing more work on experimental design for drug screening studies. In particular, we can focus on ways in which those designs can be paired with specific data processing methods. For instance, in existing drug screening studies, there seems to be no clear method behind the use of biological replicates. This includes how many are performed, which doses are replicated, and where that replication occurs on the plate. Carefully designing a plate with an associated data normalization method, however, may lead to the best use of the collected data. This will allow us to know that technical errors can be removed. Further, making

an explicit connection between the experimental design and the quality of downstream data might encourage the use of more complicated, and more effective, designs up front.

APPENDIX A

Large-Scale Cancer Drug Screening Studies

Much of this work focuses on processing and analyzing data from large-scale cancer drug screening studies. These studies are often the first step in the drug discovery process. Researchers will consider a wide range of potential anti-cancer drugs, including experimental compounds, clinically approved drugs that may be effective off-label, and drugs in clinical development (*Yang et al.*, 2013). In a drug screening study, these compounds are tested on a wide range of cancer cell lines. These cell lines consist of cancer cells that have been extracted from a patient's tumor and kept growing in a laboratory. Cancer cell lines are useful for testing new treatments.

Drug screening experiments are performed on microplates, often 96-, 384-, or 1536-well plates. The general experimental procedure for each plate is as follows:

1. Place culture media in each well. Note: plate design may include unused wells around the plate edges that do not receive culture media.

2. Seed every well with the specified density of cells from the cell line under consideration. The same cell line will be placed in every well. Note: plate design may include blank control wells that do not receive any cells.

3. Add drug to each well; different drugs and different drug concentrations may be applied to each well on the plate. Plate design may also include untreated control wells that received cells, but do not receive any drug.

4. Allow the microplate to incubate for a set amount of time, e.g., 72 hours.

5. Add stain to each well.

6. Scan the plate to obtain an intensity measurement for each well.

The intensity measurement for a given well represents the number of active cells in that well. A high intensity indicates the presence of many active cells, implying that the drug was not effective

at inhibiting cell growth. On the other hand, a low intensity indicates the presence of few active cells in that well, implying that the drug was effective at inhibiting cell growth. Overall, each intensity is a measure of drug efficacy.

The data from such large-scale cancer drug screening studies is often paired with genomic information about the cell lines under consideration. Such genomic information can include gene expression levels, methylation levels, copy number variation, and mutation status. Together, these data sources are used to identify genetic features (e.g., specific genes or mutations) that are predictive of drug efficacy. For instance, on a simple scale, we want to identify a set of genes that have different expression levels in cell lines where Drug X is effective and in cell lines where Drug X is ineffective. This setting is further discussed in Chapter 4.

In this dissertation, we specifically use pharmacogenomic data from the Genomics of Drug Sensitivity in Cancer (GDSC) project and the Cancer Cell Line Encyclopedia (CCLE) (*Yang et al.*, 2013; *Barretina et al.*, 2012). More details about these databases are provided in each chapter.

APPENDIX B

Appendices for Technical Variation in Drug Screening Studies

B.1 Data Retrieval

GDSC We obtained data from the Genomics of Drug Sensitivity in Cancer (GDSC) Project (*Yang et al.*, 2013). Plate layouts, raw sensitivity data, fitted parameters for sigmoidal dose-response curves, and R code were downloaded from Github (https://github.com/CancerRxGene; Jan. 2018). Details about tested compounds, cell lines, and the experimental procedure were retrieved from the GDSC website (https://www.cancerrxgene.org/ and https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources/Home.html).

Unless otherwise noted, AUC estimates used in our analysis were provided by GDSC. When we calculated our own estimates, we calculated AUC as the area under the relative viabilities without fitting a dose-response curve. Relative viabilities were calculated as raw intensities divided by the median untreated control intensity. We did not cap our relative viabilities or AUC estimates at 1.

CCLE We obtained processed drug sensitivity data from the Cancer Cell Line Encyclopedia (CCLE) (https://portals.broadinstitute.org/ccle/data; Jul. 2019) (*Barretina et al.*, 2012). We also obtained the raw drug sensitivity data (https://www.nature.com/articles/s41586-018-0722-x) (*Barretina et al.*, 2019).

In the CCLE study, each compound was tested over 8 concentrations with 3.16-fold dilution and a maximum concentration of 8 μM. Before doing our analysis, we removed all drug-cell line combinations with missing intensities; only those with available measurements for all 8 doses were included.

Intensities for the vast majority of wells on CCLE plates were not released; the most complete plates have intensities for fewer than 10% of wells. CCLE documentation indicates that wells flagged as invalid have not been released. Additionally, correspondence with a former CCLE

investigator indicates that privately owned compounds from Novartis may have been tested on these plates, but not included in the public release.

CCLE reports a single AUC estimate for each drug-cell line combination, which is the result of taking the median across all replicates. Therefore, to do our analysis, we recalculated AUC estimates for each replicate individually. We calculated AUC as the area under the relative viabilities without fitting a dose-response curve. Relative viabilities were calculated as raw intensities divided by the median untreated control intensity. We did not cap our relative viabilities or AUC estimates at 1, with the exception of Figure B.1. In this figure, we capped relative viabilities at 1 to more closely match CCLE's processing methods.

B.2 Within-Study Replication

Both GDSC and CCLE have intra-study replication, and we used these repeated measurements to evaluate AUC agreement within each study. We found varying levels of consistency across cell lines for each drug (Table B.1; Figure B.1).

For CCLE, we also considered concordance for narrowly effective and broadly effective compounds, defining these classes as in *Safikhani et al.* (2016b). We identified broadly effective compounds as those with an AUC median absolute deviation (MAD) > 0.13 (7 compounds: `NVP-BAG500-NX-4`, `NVP-LBH589-CU-2`, `NVP-LBN777-NX-1`, `NVP-LBN816-AA-1`, `NVP-LBW624-NX-2`, `NVP-LEE850-NX-1`, and `NVP-LFE158-NX-3`). We identified narrowly effective compounds as those with an AUC MAD ≤ 0.13 and more than 5 cell lines with an AUC less than 0.8 (remaining 20 compounds). Pearson correlation tends to be higher for broadly effective drugs (mean: 0.78; median 0.78, standard deviation: 0.07) than for narrowly effective drugs (mean: 0.60; median 0.64, standard deviation: 0.13).

Drug	n	All AUC	All IC_{50}	n	Sensitive IC_{50}
AZD6482	829	0.47	0.46	115	0.28
Refametinib	801	0.84	0.83	290	0.74
PLX-4720	844	0.71	0.67	42	0.69
Pictilisib	758	0.55	0.54	358	0.47

Table B.1: Within-study agreement for GDSC drugs. Pearson correlation for the four replicated GDSC drugs. The `Sensitive` IC_{50} column (and corresponding `n`) only considers cell lines with an IC_{50} estimate below the maximum tested dose. AUC and IC_{50} estimates were provided by GDSC.

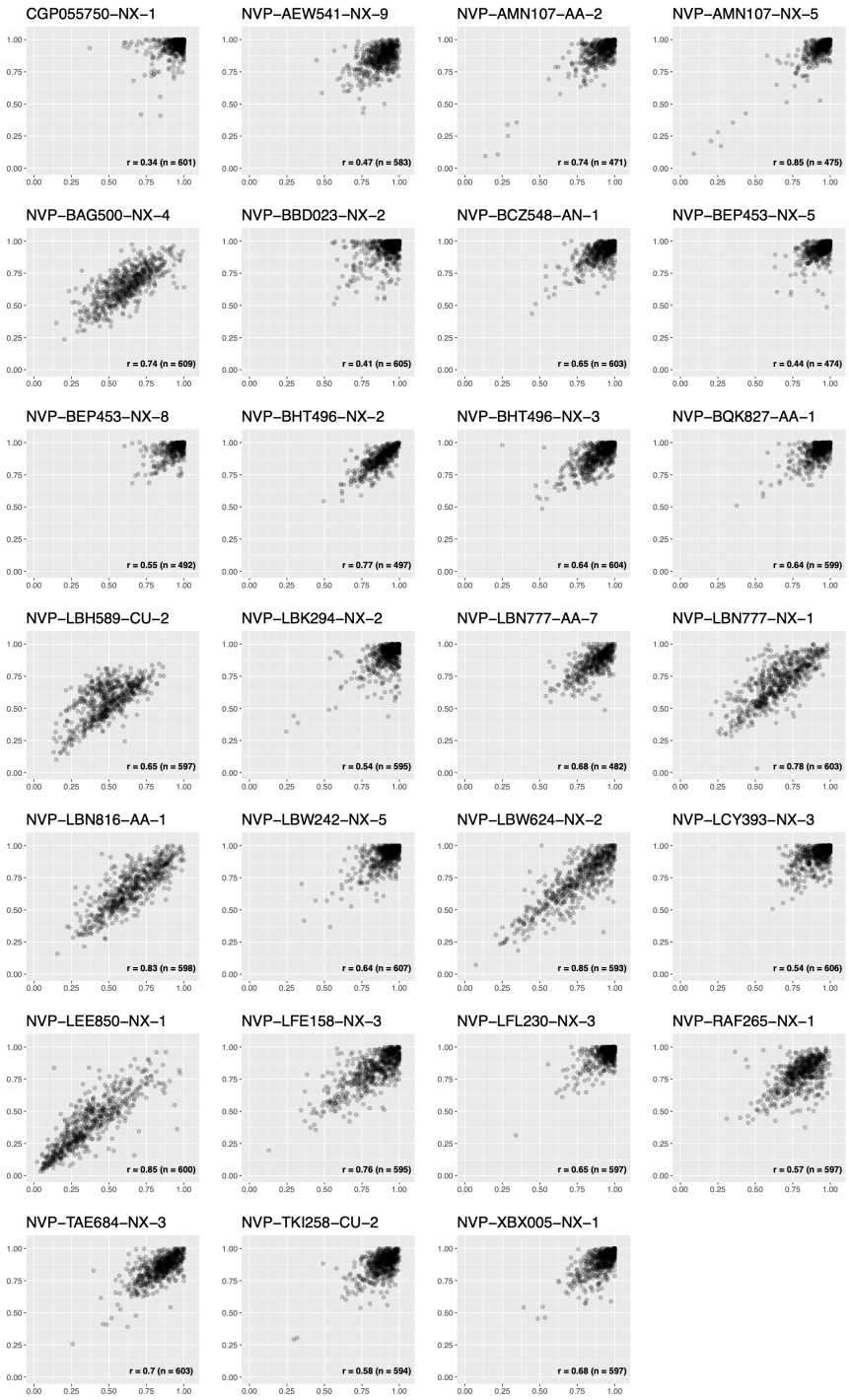

Figure B.1: Within-study agreement for CCLE drugs. AUC estimates for the 27 drugs replicated within CCLE. Each point is one cell line. Pearson correlation (mean: 0.65; median: 0.65; standard deviation: 0.14) and the number of cell lines tested is provided for each drug. Note that CCLE reports a single AUC estimate for each drug-cell line combination, which is the result of taking the median across all replicates. To construct these plots, we calculated AUC estimates for each replicate individually. Relative viabilities were capped at 1 to match CCLE's processing methods.

B.3 Spatial Effects

All GDSC plates scanned at WTSI ($n = 6,682$ plates) have untreated control wells in columns 2 and 23 or in columns 3 and 23. We used these wells to estimate the magnitude of horizontal plate-wide spatial effects. For each plate, we calculated the absolute difference between the median of the $\log_2$ intensities of the untreated controls in the left column (2 or 3) and the median of the $\log_2$ intensities of the untreated controls in the right column (23). For 52.5% of plates, this difference is greater than 0.1; for 14.5% of plates, this difference is greater than 0.25.

The location of control wells on GDSC plates scanned at MGH and on all CCLE plates do not allow a similar quantification of horizontal spatial effects for those settings. Additionally, the plate layouts of all GDSC and CCLE plates do not allow a similar quantification of vertical spatial effects.

We also investigated the consistency of spatial effects across plates. For each of the 125 different plate layouts in the GDSC study, we calculated the median intensity for each untreated control well, with the median taken across the hundreds of plates designed with that layout. For several of those layouts, a clear spatial gradient was visible, highlighting the systematic nature of these spatial artifacts (Figure B.2).

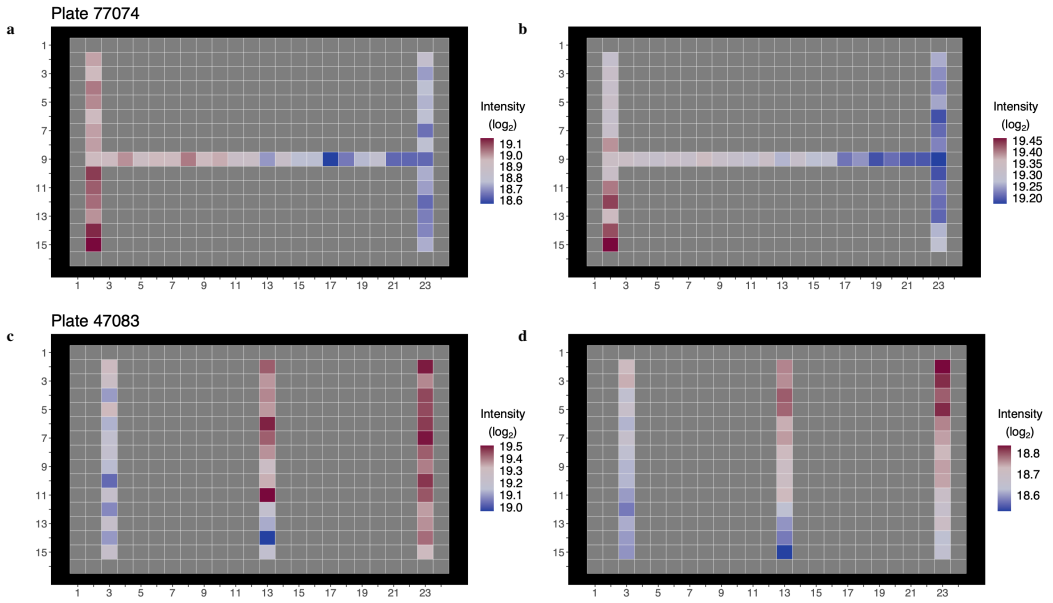

Figure B.2: Visualizing spatial effects in GDSC. (**a**) Spatial effects in the untreated control wells on a GDSC plate. The control wells in columns 2 and 23 are used to quantify the magnitude of horizontal spatial effects. (**b**) Median spatial effects in the untreated control wells, where the median for each well is taken across all plates of this format (n = 232 plates). (**c**) Spatial effects in the untreated control wells on a GDSC plate. The control wells in columns 3 and 23 are used to quantify the magnitude of horizontal spatial effects. (**d**) Median spatial effects in the untreated control wells, where the median for each well is taken across all plates of this format (n = 414 plates).

B.4 Checkerboard Pattern

To quantify the amount of checkerboard pattern present on a GDSC plate, we considered only the untreated control wells. This ensures that no biological signal in the drugged wells influences our measure.

Checkerboard pattern is characterized by alternating wells of high and low intensity. When a strong checkerboard pattern is present, the majority of wells are surrounded by wells with higher (or lower) intensities. Therefore, for each untreated control well, we determined if the intensity in that well was greater than or less than the mean of the intensities in the surrounding untreated control wells. Due to the structure of GDSC plates, most wells were compared to the mean of the two wells on either side. If the well of interest was greater than the mean of the surrounding wells, it was assigned a value of +1; if it was less than the mean of the surrounding wells, it was assigned a value of -1. All wells other than the untreated controls were assigned values of NA and not included in the calculation.

The values of ± 1 and NA were placed in a matrix with the same format as the scanned plate. This matrix was then multiplied element-wise with a pre-constructed checkerboard matrix (a matrix with alternating +1 and -1). We summarized the resulting matrix by taking the proportion of +1's minus the proportion of -1's. A value of 0 indicates no checkerboard pattern in the untreated controls, while a value of +1 or -1 indicates a perfect checkerboard pattern. We calculated this measure for all GDSC plates, excluding 96-well plates which only have 6 untreated control wells ($n = 13,182$ plates).

To identify plates that suffer from a checkerboard pattern, we considered the absolute value of this checkerboard measure. Figure B.3 shows the untreated control wells for two plates with

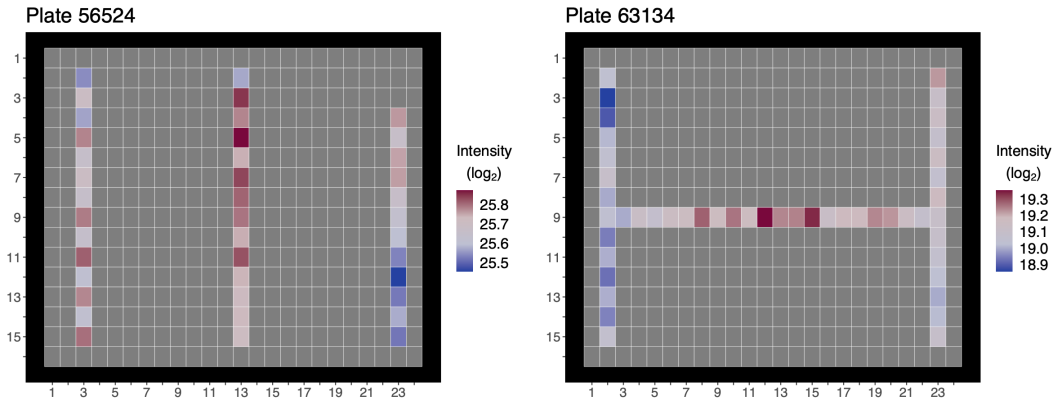

Figure B.3: Quantifying checkerboard pattern in GDSC. Two untreated control heatmaps with a checkerboard pattern. Both plates have checkerboard measure $= 0.5$. Any plate with checkerboard measure ≥ 0.5 is considered to have substantial checkerboard pattern.

a visible checkerboard pattern and an absolute checkerboard measure of 0.5. Any plate with an absolute checkerboard measure ≥ 0.5 is considered to have substantial checkerboard pattern (9.8% of plates).

The layout of CCLE plates prohibited us from constructing a similar measure to quantify checkerboard pattern in the CCLE study.

B.5 Batch-Specific Outliers and Noise

We found several groups of plates in both GDSC and CCLE with consistent technical artifacts. These include GDSC plates containing drug AZD6482 (drug 1066) scanned on November 9, 2011 (Figure B.4a), CCLE plates in batch `2009_12_16 PM` (Figure B.4c), GDSC plates containing drug KIN001-260 scanned on June 21, 2012, and GDSC plates containing drug sepantronium bromide scanned on October 4, 2012.

B.6 Drug Sensitivity Measures

IC_{50} estimates are a commonly used measure of drug sensitivity in large pharmacogenomic studies. We did not focus on IC_{50} in our analysis, however, because of its many challenges. For example, spatial effects can have a large impact on the accuracy of IC_{50} estimates, and IC_{50} estimation depends heavily on the method used to fit dose-response curves.

Additionally, IC_{50} is only a meaningful measure of drug sensitivity if the observed relative viabilities cross 50%. When they do not, IC_{50} cannot be reported or the reported value will be outside the range of tested drug concentrations and, therefore, less reliable. GDSC and CCLE handle this situation differently: GDSC reports extrapolated IC_{50} estimates, while CCLE reports the maximum tested drug concentration as the IC_{50}. In both situations, it is not clear how to interpret the reported value.

For other drug-cell line combinations, IC_{50} does not capture the most important biology. Consider, for instance, a drug-cell line combination with a sigmoidal dose-response relationship that has an upper asymptote at 1 and a lower asymptote at 0.5. In this situation, it is important to note that, while this drug cuts cell growth in half, larger doses will not cause more growth inhibition. IC_{50} cannot capture this relationship.

It has been shown that values based on growth rate inhibition, like GR_{50}, are better at measuring drug efficacy than IC_{50}; however, GR_{50} cannot be calculated without information about cell seeding and cell growth (*Hafner et al.*, 2016). The provided GDSC and CCLE data does not allow the calculation of GR_{50}. Therefore, in this paper, we use AUC to measure drug sensitivity. We

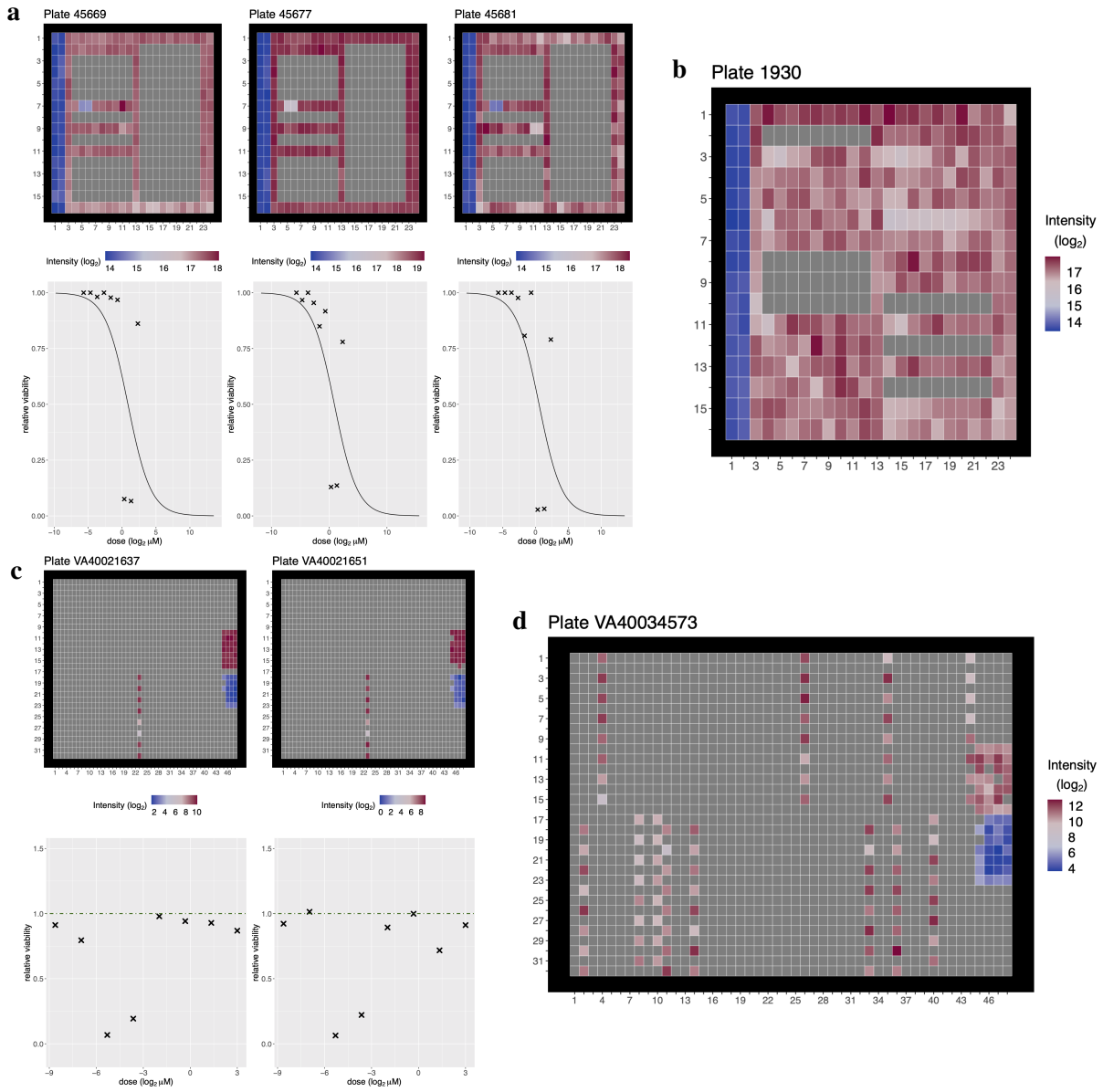

Figure B.4: Batch-specific outliers and noise in raw GDSC and CCLE data. (**a**) Three GDSC scans from November 9, 2011 showing batch-specific outliers. Most plates of this format scanned on this day have extremely low intensities in columns 5 and 6 of row 7. The outliers are clear in the corresponding dose-response curves. Relative viabilities and curves are calculated as in the GDSC study. (**b**) A noisy GDSC heatmap with inconsistent intensities in control wells and unexpected jumps in drugged wells. (**c**) Two scans from CCLE *Batch 2009_12_16 PM* showing batch-specific outliers. Most plates in this batch have extremely low intensities in rows 26 and 28 of column 23. The outliers are clear in the corresponding dose-response curves. (**d**) A noisy CCLE heatmap with inconsistent intensities in control wells and unexpected jumps in drugged wells.

calculate AUC as the area under the relative viabilities and do not fit a dose-response curve. We discuss challenges with AUC estimation in the main text.

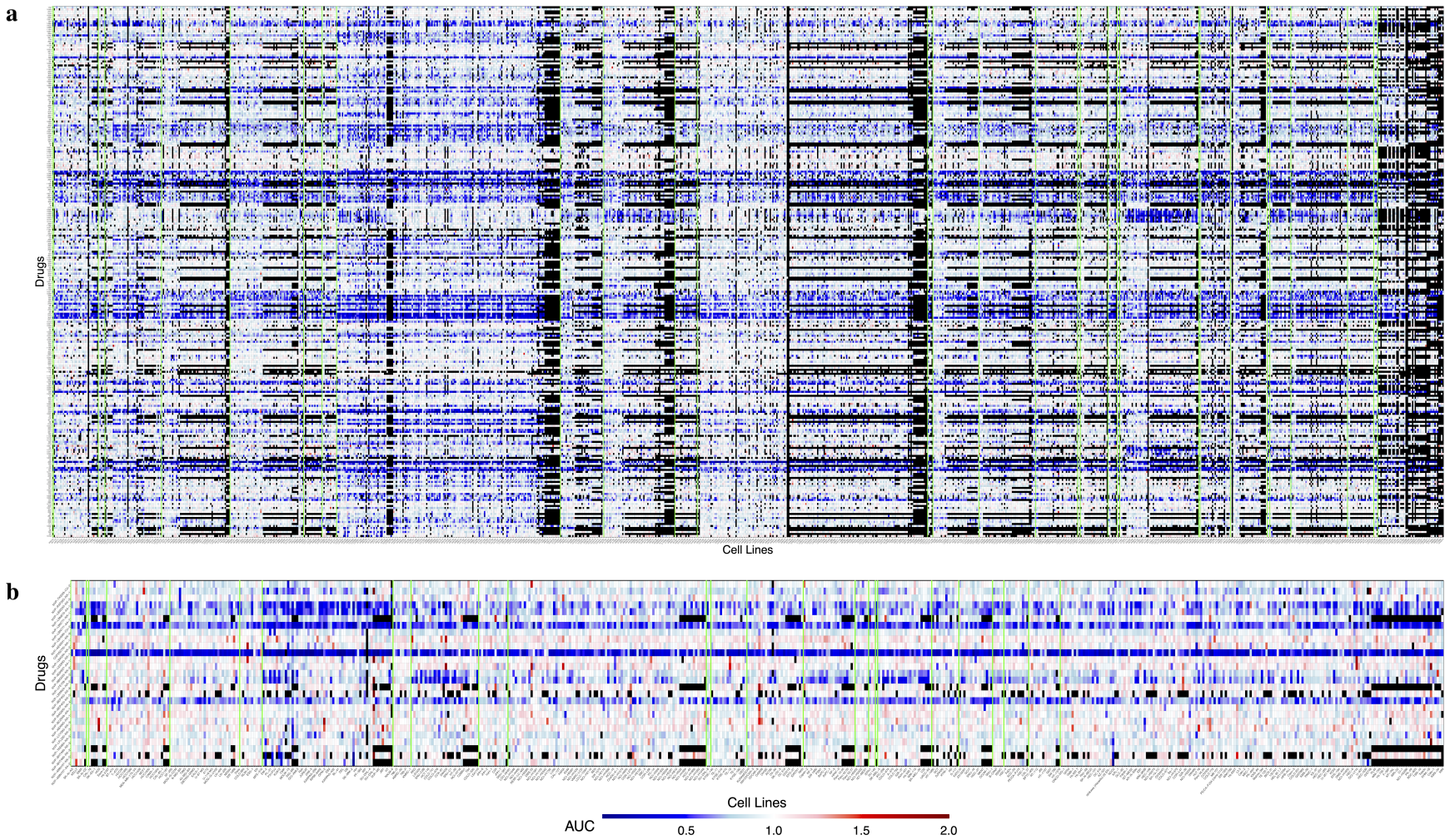

Figure B.5: GDSC and CCLE AUC estimates. AUC values for all (**a**) GDSC and(**b**) CCLE drug-cell line combinations (one replicate was randomly chosen for each). We calculated AUC estimates as the area under the dose-response observations. Black cells indicate combinations that were not tested. Drugs are ordered by target pathway and cell lines are ordered by two tissue type descriptors, site and histology; all annotations came from GDSC. Broadly effective drugs are evidenced by blue stripes across all cell lines.

B.7 CCLE Drug L-685458

The slopes from regressing $\log_2$ intensity on $\log_2$ drug dose for CCLE drug L-685458 do not indicate broad growth promotion. Instead, Figure B.6a shows slopes centered at 0 and indicates that L-685458 is a mostly ineffective compound – some slopes are a little larger than 0, some are a little smaller than 0, and some are much smaller than 0, corresponding to the few sensitive cell lines.

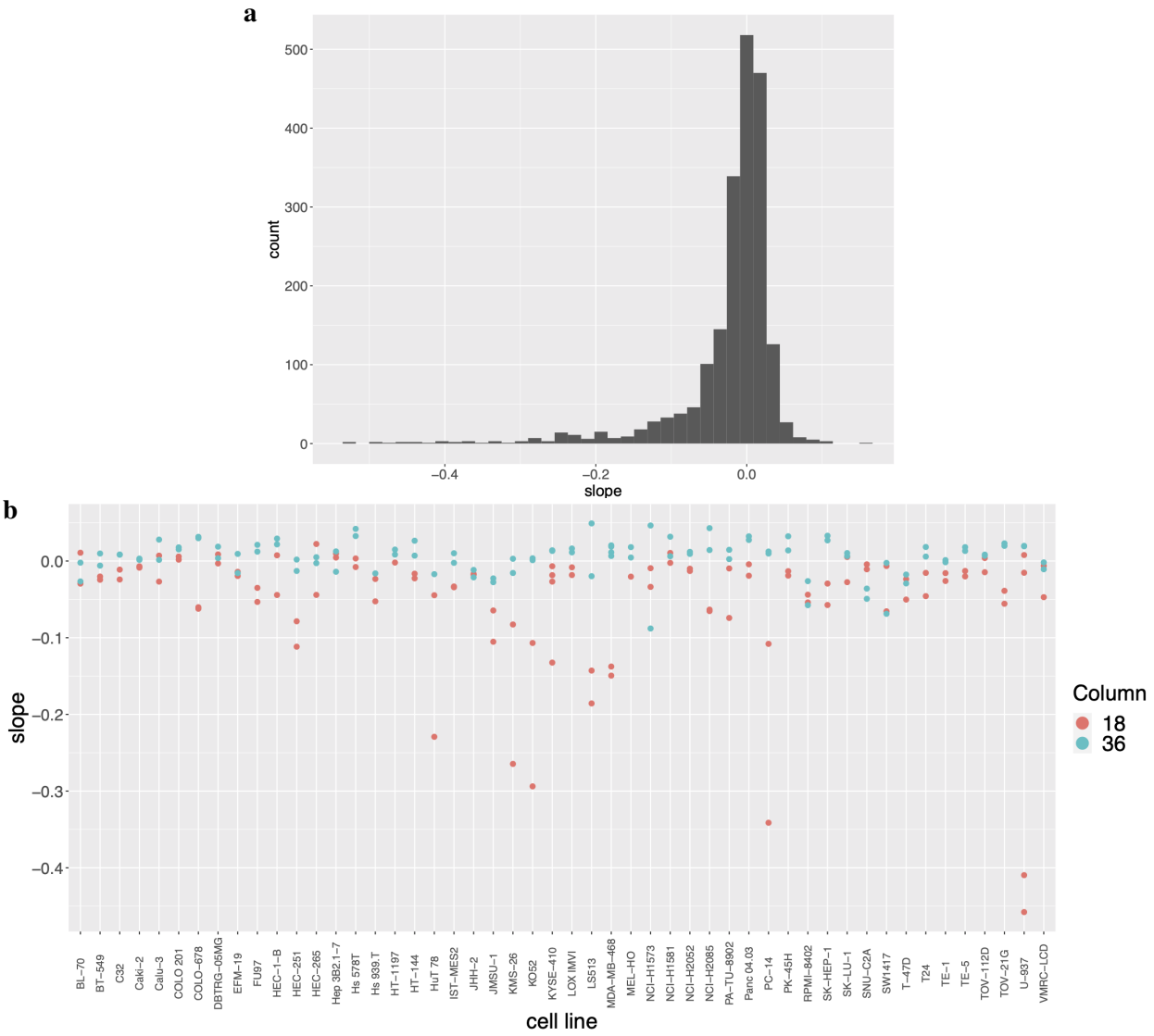

Figure B.6: Interaction of technical error and plate design for drug L-685458. (**a**) Dose-response slopes from regressing $\log_2$ intensity on $\log_2$ drug dose for all cell lines on which CCLE drug L-685458 was tested ($n = 2,004$ tests). (**b**) Slopes for a random sample of 50 cell lines that had drug L-685458 tested in both column 18 and column 36. For the majority of cell lines, the slope from column 18 is less than the slope from column 36.

B.8 Location Effects in CCLE

For each drug in CCLE, we considered the correlation between replicated AUC estimates when those replicates were tested in the same location on two different plates or in different locations on two different plates. There was no within-plate replication in the CCLE study. In this investigation, we found high correlation when both replicates were tested in the same location (mean: 0.76; median: 0.77; standard deviation: 0.12) and low correlation when the replicates were tested in two different locations (mean: 0.48; median: 0.54; standard deviation: 0.23; Figure B.7). This indicates strong systematic differences in sensitivity based on location.

We found further evidence of these systematic differences when considering only the drug-cell line combinations that were replicated in the same location. Among these, most drugs were tested in multiple locations on different plates. Specifically, while both replicates of cell line A were tested in a single location, and both replicates of cell line B were tested in a single location, the locations for cell line A and cell line B could be different. For instance, for irinotecan, some cell lines were replicated in rows 2 through 16 of column 33, while others were replicated in rows 17 through 31 of column 22.

As displayed in Figure B.8, these different locations can produce systematically different results. For some drugs, the amount of noise and level of concordance varies between well locations, while for other drugs, the magnitude of AUC values differs. These systematic differences between locations prevent sensitivity estimates for a single drug to be accurately compared across cell lines or for a single cell line to be compared across drugs.

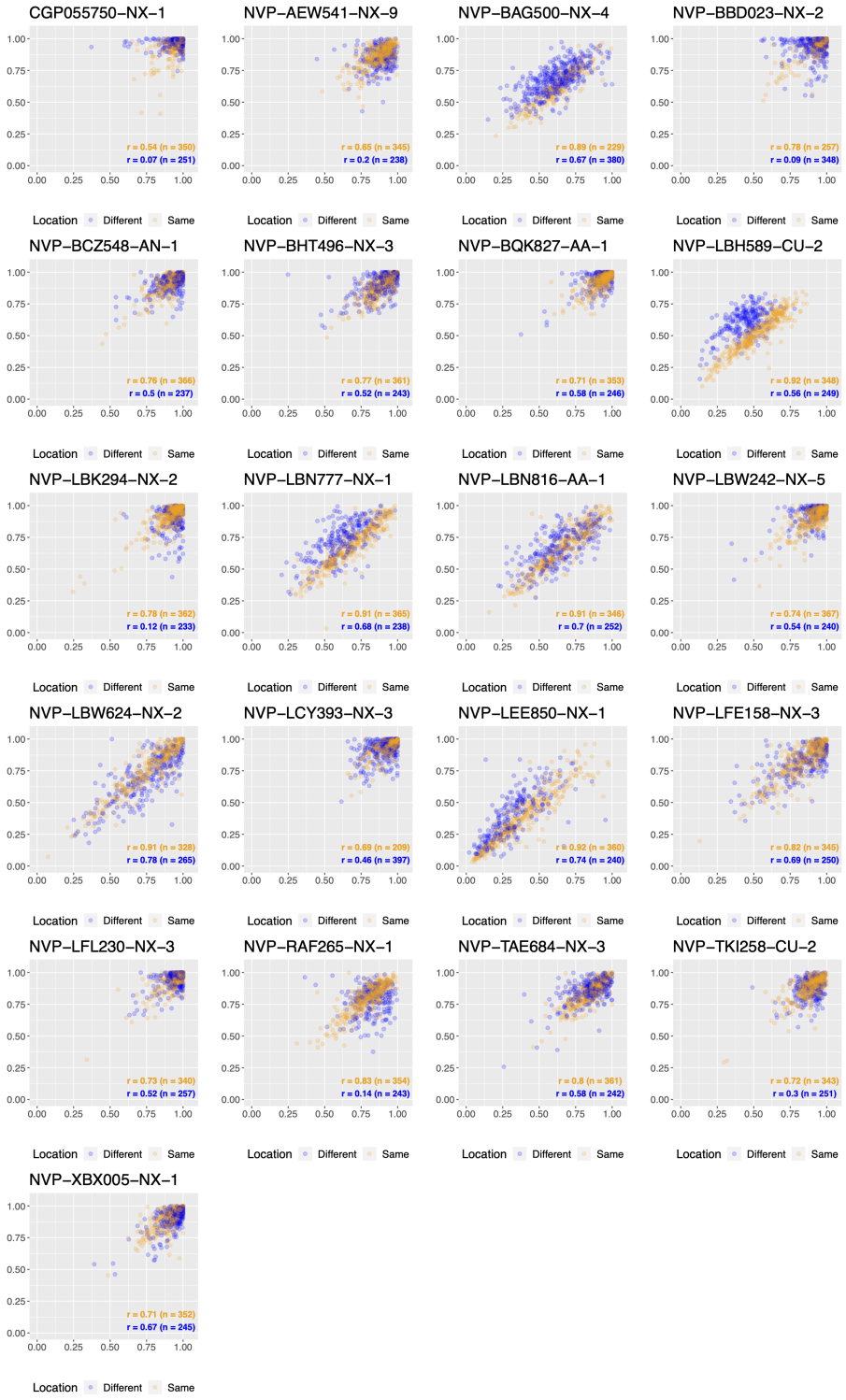

Figure B.7: Location effects for CCLE. AUC estimates for replicated CCLE drug-cell line combinations. The coloring indicates whether the replicates for each cell line were tested in the same location or in different locations across plates. Only drugs that had replicates tested in both the same location and in different locations are shown.

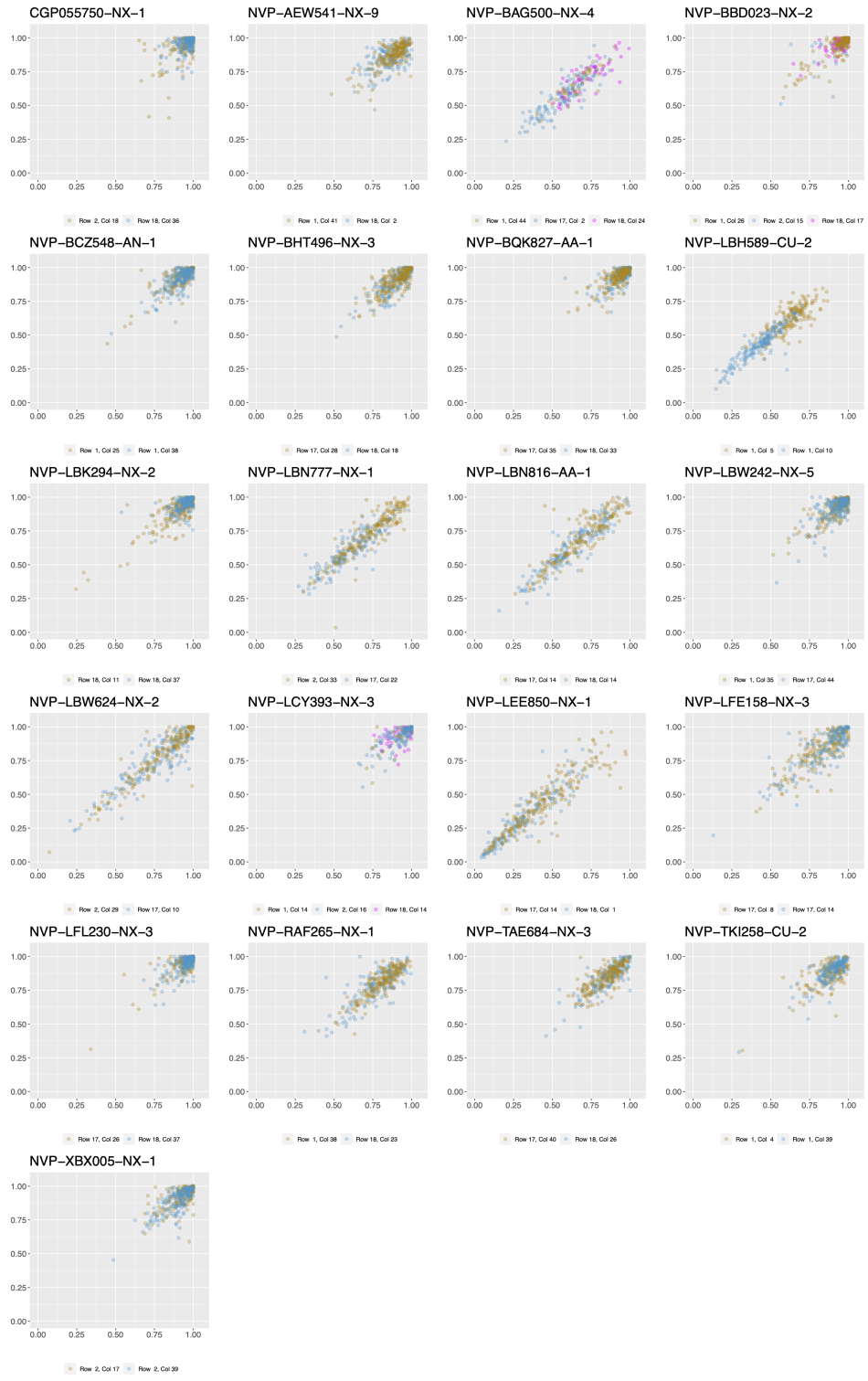

Figure B.8: Location effects for CCLE cell lines tested in the same location across plates. AUC estimates for replicated CCLE drug-cell line combinations that were tested in the same location across plates. Cell lines are colored by the location in which both replicates were tested (row and column of the highest drug dose).

B.9 Challenges for Analytical Methods

Spatial adjustment We applied two traditional spatial adjustment methods to replicates in the GDSC study: linear regression and loess regression. A linear adjustment method is particularly relevant given the layout of GDSC plates and the considerable distance between some drugged wells and the nearest control wells (Figure 2.1). Neither adjustment method, however, substantially improved data quality or agreement between replicates (Figure B.9).

The ability of linear and loess regression to accurately estimate and remove spatial effects is impaired by the confounding between biology (a gradient in well intensities caused by an effective drug) and technical variation. Because of this confounding, these methods can pick up on biological effects and inadvertently introduce spatial bias to previously clean plates (Figure B.10ac). Using an adjustment method based only on the untreated control wells avoids this problem and does not hurt clean plates, but may not necessarily mitigate existing spatial bias. In particular, linear regression adjustments are unable to address non-linear spatial effects (Figure B.10b). Loess handles non-linear spatial effects somewhat better, but because it is a "local" regression, it is hindered by the non-representative locations of the control wells and, in particular, the large distance between some drugged wells and the nearest control wells.

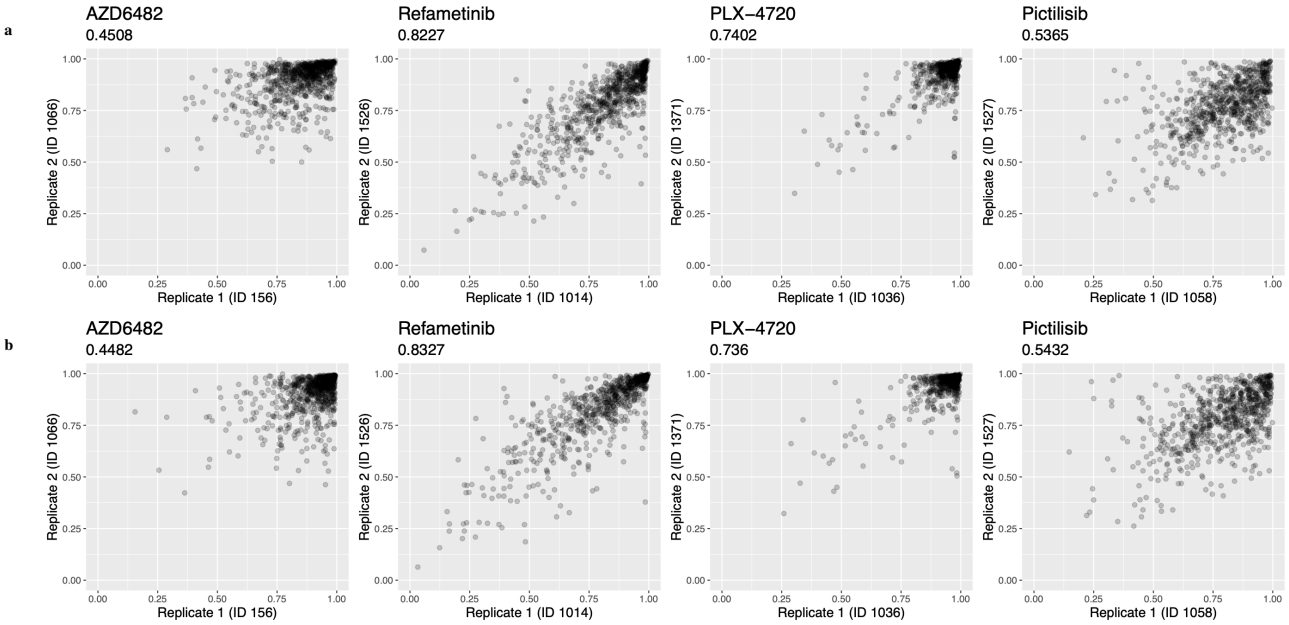

Figure B.9: AUC agreement does not improve with linear regression adjustment. AUC estimates for the four drugs replicated within GDSC. The raw data has been adjusted for spatial effects using (**a**) linear regression on all drugged and untreated control wells and (**b**) linear regression on all untreated control wells only. Each point is one cell line. AUC values were calculated using the GDSC analysis pipeline. Pearson correlation is provided at the top of each plot.

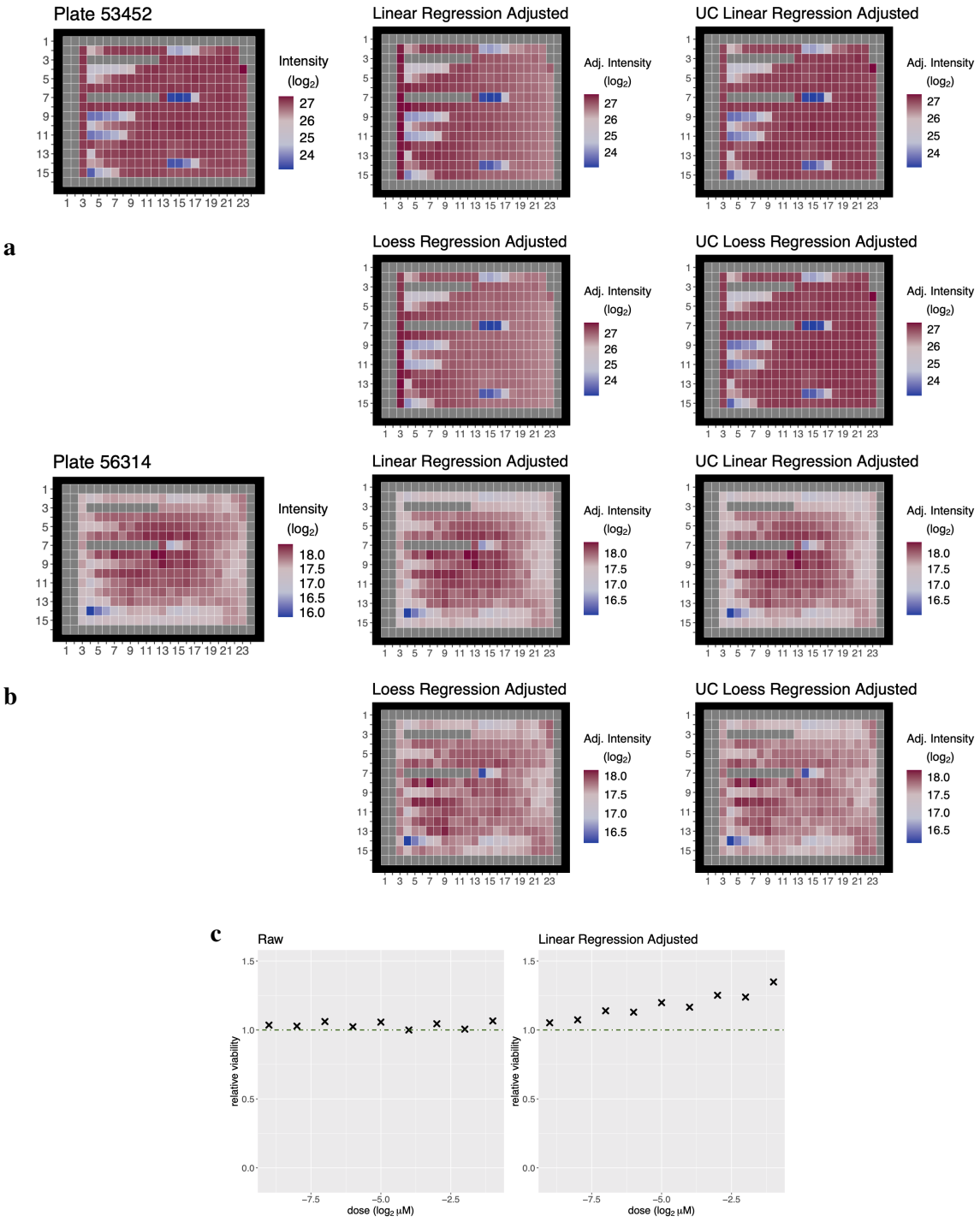

Figure B.10: Technical error is challenging for traditional analytical methods. (**a**) The result of using linear and loess regression to spatially adjust a clean GDSC plate with several effective drugs and (**b**) a GDSC plate with non-linear spatial effects. Column 1 shows unadjusted data; column 2 shows adjusted data where the adjustment is based on drugged wells and untreated control wells; column 3 shows adjusted data where the adjustment is based on untreated control wells only. (**c**) Dose-response curves for an insensitive drug-cell line combination before and after applying a linear regression adjustment. This adjustment technique introduces spatial bias to a previously clean plate.

Capped relative viabilities Capping relative viabilities at 1 can artificially increase agreement in AUC between replicates. This is particularly true for CCLE where the untreated control intensities are too low for many drug-cell line pairs (for 32% of the data, the median intensity of the untreated controls is more than 0.1 $\log_2$ units smaller than the intensity in the wells treated with the lowest drug dose). In these situations, normalizing with the untreated controls will result in many large relative viabilities that will all be flattened to 1 during truncation, eliminating all variability. Replicates that differ greatly before truncation could become identical after truncation. Because of the low untreated control intensities in CCLE, some of the variability in relative viabilities above 1 is biologically meaningful and this information is lost (Figure B.11).

Parametric dose-response curves Drug-cell line combinations that contain a strong checkerboard pattern are not well-suited to the use of a parametric dose-response curve fitting method (Figure B.12).

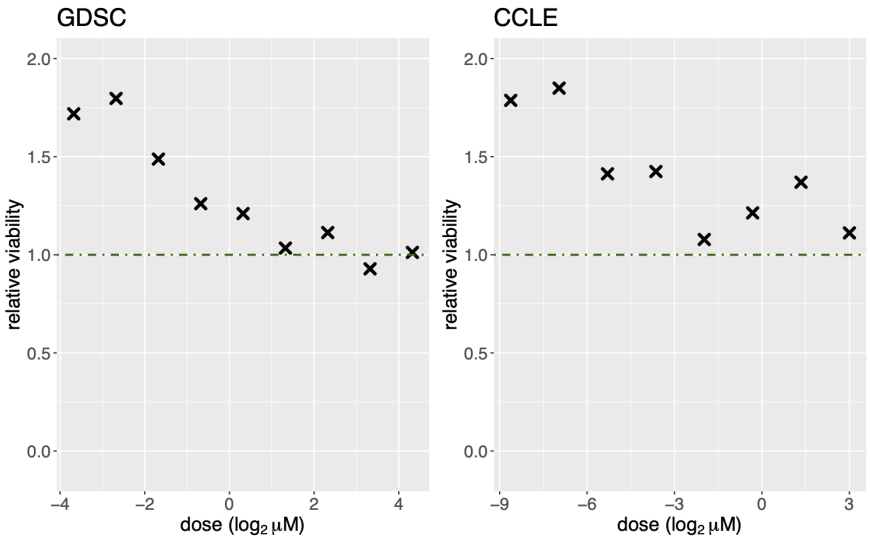

Figure B.11: Capping relative viabilities eliminates biology. Dose-response curves for two apparently sensitive drug-cell line combinations (decreasing cell viability with increasing drug dose). Almost all relative viabilities are larger than 1, so capping at 1 would remove all signal of interest.

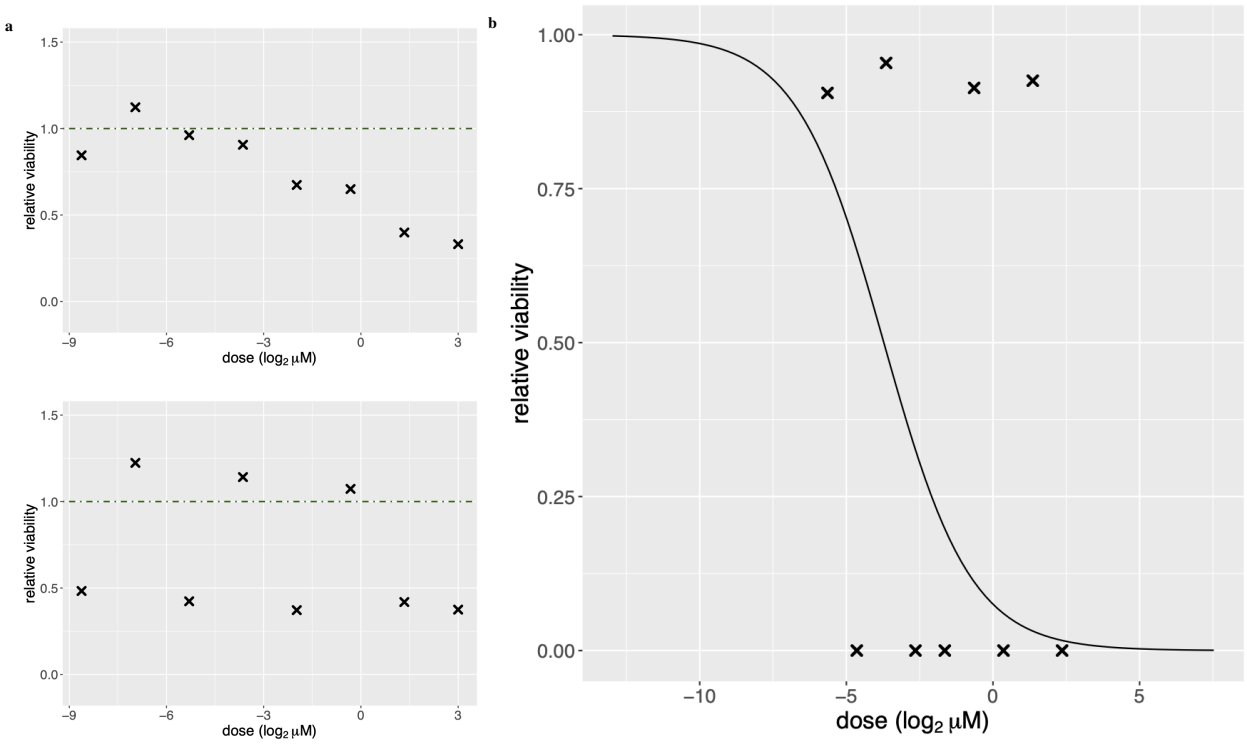

Figure B.12: Dose-response curves with checkerboard pattern. (**a**) Two CCLE replicates. One shows the cell line is sensitive to the drug with a dose-response relationship that could reasonably be modeled by a sigmoid curve. All signal in the other replicate is obscured by checkerboard pattern. (**b**) A sigmoidal dose-response curve cannot handle a severe checkerboard pattern. This curve is fit as in GDSC.

APPENDIX C

Appendices for Flexible and Spatially Varying Normalization for Well-Based Assays

C.1 Data Retrieval

We obtained data from the Genomics of Drug Sensitivity in Cancer (GDSC) Project (*Yang et al.*, 2013). Version 8.2 of the GDSC2 raw drug sensitivity data and annotations for tested compounds, cell lines, and the experimental procedure were retrieved from the GDSC website (https://www.cancerrxgene.org/downloads/bulk_download; November 2020).

C.2 Existing Relative Viability Methods

In Section 3.2.1, we introduced two relative viability normalization approaches. One approach is from the Cancer Cell Line Encyclopedia (CCLE):

$$\mathrm{V}_{ijk}^{uc} = \frac{Y_{ijk}}{\tilde{U}_k},$$

where $\tilde{U}k$ is the median of the untreated control wells on plate k (*Barretina et al.*, 2012). The other relative viability approach is from GDSC:

$$V_{ijk}^{GDSC} = \frac{Y_{ijk} - \bar{B}_k}{\bar{U}_k - \bar{B}_k},$$

where $\bar{B}_k$ is the mean of the blank controls on plate k and $\bar{U}_k$ is the mean of the untreated controls on plate k (*Vis et al.*, 2016). In the main text, we chose to focus on the CCLE normalization (called UC normalization) because it tends to have better performance than the GDSC normalization does.

The main difference between the CCLE and GDSC approaches is the use of the blank control wells in the GDSC calculation. GDSC is using the mean of the intensities in the blank control wells to estimate the intensity associated with zero cell viability. Each plate in a drug screening

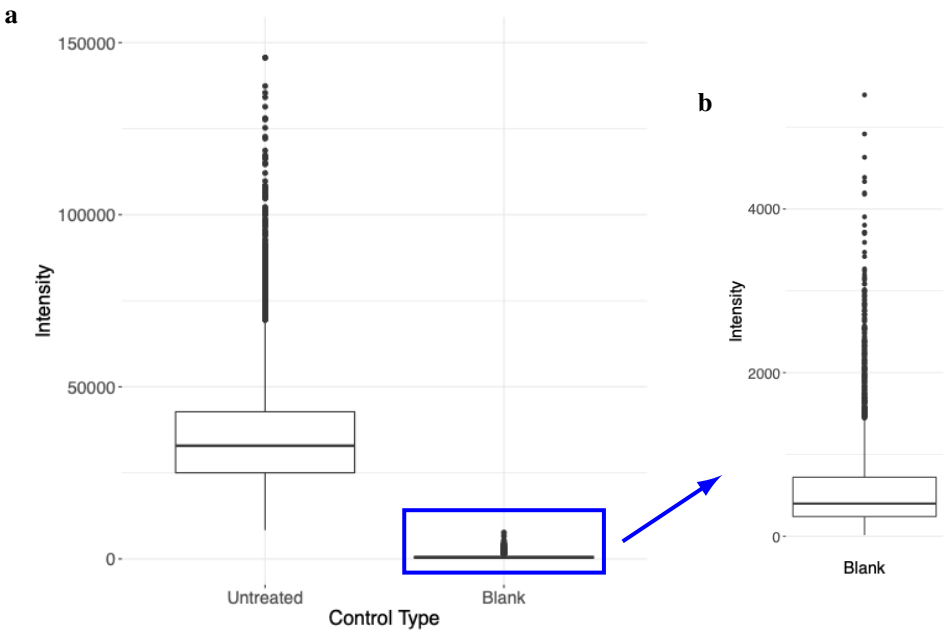

Figure C.1: Mean untreated and blank control well intensities. **(a)** Boxplots of the mean untreated and blank control intensities for all GDSC plates ($n = 7,307$ plates). The mean blank controls are negligible compared to the mean untreated controls. **(b)** A close-up of the mean blank control distribution.

study, however, typically has very few blank control wells. In GDSC, they are all placed along the plate edges. Therefore, blank controls can suffer from edge effects and introduce a large amount of noise into the relative viability calculation (*Mpindi et al.*, 2015).

Instead of relying on the blank control wells, CCLE uses an intensity of 0 to represent a viability of zero. While this may introduce some bias by underestimating the true intensity of zero viability, the amount of bias is likely not meaningful in practice (Figure C.1). Additionally, removing the blank control wells, and their associated noise, from relative viability calculations can decrease variability.

Further, the GDSC normalization summarizes the untreated and blank control intensities using the mean; CCLE uses the median. The median is robust to outliers, which is an important feature for drug screening data that can contain extremely high or low intensity wells. The median is a better summary measure in this setting.

C.3 Outlier Detection

We developed and implemented three methods to detect outliers and perform quality control on the raw data. If a drug-cell line combination was flagged by any of the methods, it was removed from our analysis. In total, we removed 1,232 out of 232,441 drug-cell line combinations (0.5%) before

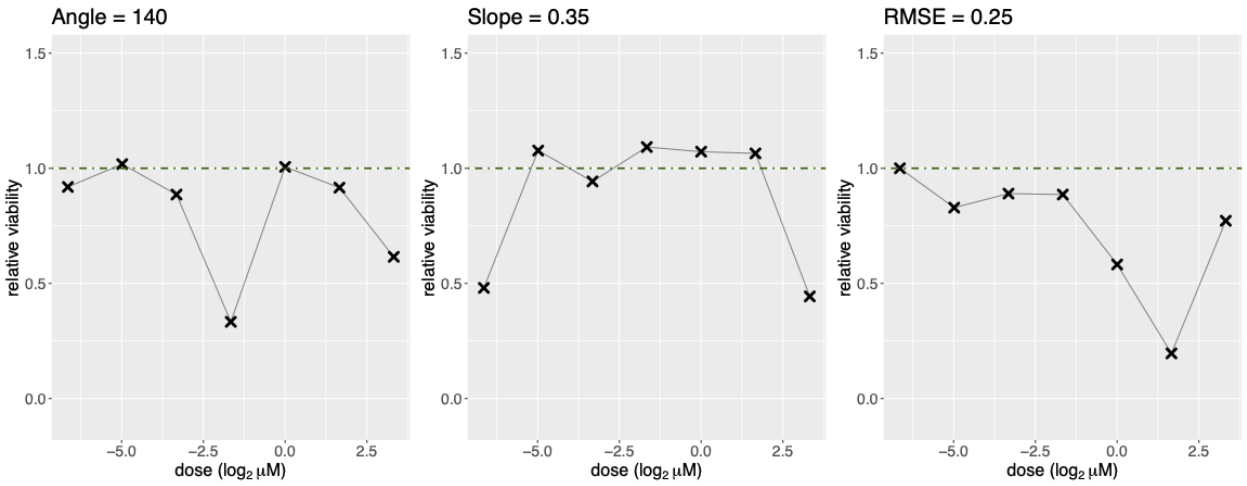

Figure C.2: Example drug-cell line combinations that represent the boundary cases for each outlier detection method. These combinations were flagged and removed before we began our analysis.

doing any further analysis. We describe these quality control techniques below.

Angle-Based Method For each drug-cell line combination, we calculated the angle between each set of three consecutive intensities. If all three intensities are equal, the angle will be 180 degrees. A small angle indicates the presence of an outlier. We flagged all drug-cell line combinations with an angle smaller than 140 degrees ($n = 743$ observations; 0.3%).

Slope-Based Method For each drug-cell line combination, we calculated the slope between each pair of consecutive intensities. If the two intensities are equal, the slope will be 0. A large and positive slope indicates the presence of an outlier. We flagged all drug-cell line combinations with a slope larger than 0.35 ($n = 868$ observations; 0.4%).

RMSE-Based Method For each drug-cell line combination, we fit a 4 parameter logistic dose response curve and calculated the root mean square error (RMSE) between the observed intensities and the predicted values. A large RMSE indicates noisy data. We flagged all drug-cell line combinations with RMSE larger than 0.25 ($n = 393$ observations; 0.2%).

C.4 Flexible Normalization for Drug Screening Studies

We develop a relative viability normalization method that estimates the counterfactual intensity for each drugged well. This procedure uses two separate approaches to estimate the counterfactual that we then combine into an ensemble method. We provide technical details for applying both

the basic normalization method introduced in Section 3.3.2 and the extended method introduced in Section 3.3.3 to the GDSC data.

C.4.1 Approach 1

The basic version of Approach 1 uses information from the untreated control wells to estimate the counterfactual for each well on the plate. The extended version of Approach 1 uses information from both the untreated control wells and the wells treated with the two lowest drug concentrations. For the extended method, we label the untreated control wells and the two lowest dose wells "untreated".

C.4.1.1 Estimate of the Counterfactual

With the basic version of Approach 1, we estimate the counterfactual for all treated wells on plate k with the median of the untreated control wells on that plate. We denote this estimate $\tilde{U}_k$ for all wells on plate k.

The extended version of Approach 1 is more involved. We fit a loess regression to the "untreated" wells on each plate. Consider plate k. For each observation on this plate, the loess regression fits a degree 2 polynomial to the nearest 30% of the data using Tukey's bisquare loss. The number of wells used to fit the loess regression varies from plate to plate based on the number and layout of non-missing wells (median: 51 wells; minimum: 36 wells; maximum: 85 wells). The tuning parameter `span = 0.3` was chosen via cross-validation. We then use the fitted loess model to predict the counterfactual, or the untreated intensity, for each well on the plate. We denote this estimate $\tilde{U}_{ijk}$ for well (i, j) on plate k.

C.4.1.2 Estimated Error Distribution

We define ε_{ijk} as the error of $\tilde{U}_k$ (or $\tilde{U}_{ijk}$ for the extended approach) as an estimator for Y^0_{ijk}, and define $\varepsilon_{ijk} = \tilde{U}_k - Y^0_{ijk}$. We estimate the distribution of ε_{ijk} from the observed GDSC data. To do so, we let L_{ijk} (from Approach 2 in Section C.4.2) approximate Y^0_{ijk}. We then estimate ε_{ijk} by calculating the difference between $\tilde{U}_k$ and L_{ijk} for every drugged well. This distribution has a long right tail corresponding to the cell lines that are sensitive to the lowest drug doses (they will have a small value of L_{ijk}). This tail is not relevant to the distribution of ε_{ijk}, however, so we flip the left tail of the observed distribution around the mode to make it symmetric (Figure C.3). To model this error distribution, we fit a t-distribution.

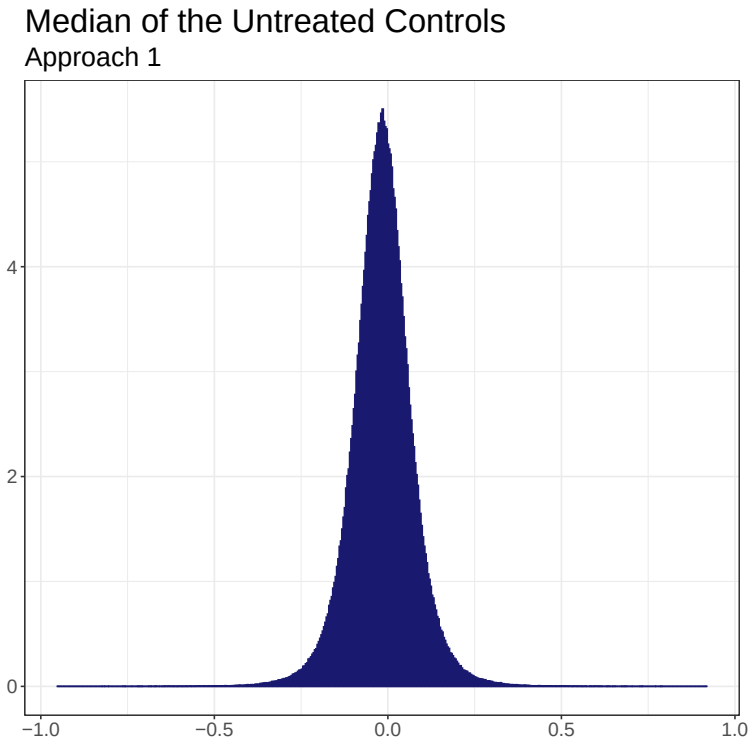

Figure C.3: The estimated error distribution for $\tilde{U}_k$, as in the basic version of Approach 1, as an estimator for Y^0_{ijk}. This is the estimated $f(\varepsilon)$ distribution.

C.4.2 Approach 2

Both the basic and extended approaches use information from the treated wells to estimate the counterfactual. For a given instance of a drug-cell line combination, both approaches produce one counterfactual estimate for all d consecutively drugged wells.

C.4.2.1 Estimate of the Counterfactual

With the basic version of Approach 2, we estimate the counterfactual for all d consecutive wells on plate k treated with a given drug with the intensity in the lowest dose treated well. We denote this estimate L_{ijk} for well (i, j) on plate k.

The extended version of Approach 2 is more complicated. For each instance of each drug-cell line combination, we calculate the UC normalized relative viabilities by taking the seven observed drugged intensities and dividing by the median intensity of the untreated control wells on the plate. We then fit a dose-response curve to the relative viabilities (details in Section C.4.2.2). From this curve, we obtain the fitted values corresponding to the two lowest drug concentrations. We then estimate the counterfactual for this drug-cell line combination with the average of these two fitted values. We denote this estimate L_{ijk}.

C.4.2.2 Dose-Response Curve Fitting

We developed an iterative dose-response curve fitting procedure that is used in extended Approach 2. For a given drug-cell line combination, we iterate three times between fitting

(a) a penalized 4 parameter logistic curve with Huber loss; and

(b) a checkerboard regression with Huber loss.

At each iteration, the response vector is obtained by subtracting the most recent fitted values from the originally observed UC relative viabilities. We use Huber loss in this procedure to minimize the impact of outliers.

In fitting the 4 parameter logistic curve, we use a half-Huber penalty on the slope and lower asymptote parameters. The penalties are defined as follows.

$$\text{slope penalty} = \begin{cases} 0 & \text{slope} \geq -6.5 \\ (\text{slope} + 6.5)^2 & -8.5 \leq \text{slope} < -6.5 \\ 2 \times (|\text{slope} + 6.5| - 1) & \text{slope} < -8.5 \end{cases}$$

$$\text{lower asymptote penalty} = \begin{cases} (\text{asymptote})^2 & \text{asymptote} \leq 0.4 \\ 0.4 \times (|\text{asymptote}| - 0.2) & \text{asymptote} > 0.4 \end{cases}$$

Additionally, if the logistic fit is unable to be initialized or does not converge, we instead fit a constant linear regression with Huber loss. This tends to occur when the dose-response relationship is flat or increasing as drug dose increases.

For the checkerboard regression, we regress the response vector on the checkerboard vector $(-\frac{3}{7}, \frac{4}{7}, -\frac{3}{7}, \frac{4}{7}, -\frac{3}{7}, \frac{4}{7}, -\frac{3}{7})$. We do not include an intercept term in this regression.

After fitting (a) and (b) three times, we obtain the final dose-response curve by fitting a penalized 4 parameter logistic curve with Huber loss to the final residuals (observed - final checkerboard fitted values). This is the dose-response curve used to estimate the counterfactual in Approach 2.

C.4.2.3 Estimated Error Distribution

We define δ_{ijk} as the error of L_{ijk} as an estimator for Y^0_{ijk}, and define $\delta_{ijk} = L_{ijk} - Y^0_{ijk}$. We estimate the distribution of δ_{ijk} using the observed data from GDSC. In particular, we calculate the difference between L_{ijk} values for adjacent (by both row and column) lowest dose wells. We then take the negative absolute value of these differences; this produces the estimated values of δ_{ijk} (Figure C.4a).

As expected, the empirical distribution of these differences has a lot of mass close to 0. This corresponds to the small amounts of noise in any estimator for the counterfactual. Further, this

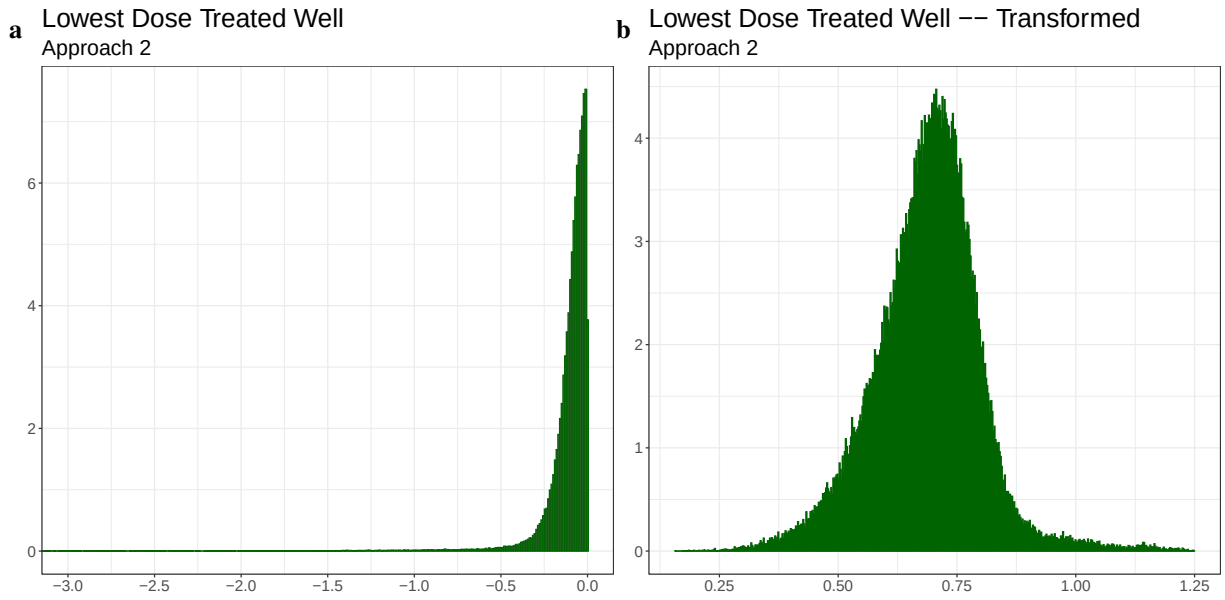

Figure C.4: **(a)** The estimated error distribution for L_{ijk}, as in the basic version of Approach 2, as an estimator for Y^0_{ijk}. This is the estimated $g(\delta)$ distribution. **(b)** The transformed $g(\delta)$ distribution that we use for modeling. We use a Box-Cox transformation, raising the errors to the 0.14 power.

distribution has a small amount of mass far away from 0, corresponding to the cell lines that are sensitive, even at the lowest drug doses. These values are large and negative because $L_{ijk} < Y^0_{ijk}$ when the drug is effective. Finally, there is no mass greater than 0 because L_{ijk} should be no larger than Y^0_{ijk}.

To model this error distribution, we use a Box-Cox transformation. We fit a t-distribution to the optimally transformed differences ($|\text{diff}|^{0.14}$; Figure C.4b).

C.5 Evaluating Normalization Performance

We applied both the UC normalization method and our new normalization technique to the drug screening data from GDSC. We used several methods to compare their performance.

C.5.1 Area Under the Curve (AUC)

We used two different methods to calculate AUC for each instance of each drug-cell line combination. The first approach uses the trapezoid method to calculate the area under the relative viabilities. The second approach uses numerical integration to calculate the area under the fitted dose-response curve, where the curve was fit using the iterative procedure outlined in Appendix C.4.2.1. We discuss results using the numerically integrated AUC in the main text.

C.5.2 Agreement Between Replicates

To evaluate normalization performance, we compared AUC estimates across replicated drug-cell line combinations. Specifically, for each drug, we calculated the absolute difference in AUC for cell lines with repeated measurements. The median of these absolute differences is a measure of agreement across replicates for that drug. A small median absolute difference, and a distribution of differences close to zero, indicates good agreement.

To compare the performance of our normalization method to the existing UC method, we compared median absolute differences across the two normalizations. We let $\tilde{d}_i^{UC}$ and $\tilde{d}_i^{EML}$ be the median absolute difference for drug i for the UC normalization and our normalization, respectively. We defined three performance categories as follows:

(a) our normalization increases agreement if $\tilde{d}_i^{UC} - \tilde{d}_i^{EML} > 0.01$;

(b) our normalization decreases agreement if $\tilde{d}_i^{UC} - \tilde{d}_i^{EML} < -0.01$; and

(c) our normalization does not change agreement if $|\tilde{d}_i^{UC} - \tilde{d}_i^{EML}| \leq 0.01$.

APPENDIX D

Appendices for Computationally Efficient Approximate Cross-Validation for High-Dimensional Linear Discriminant Analysis

D.1 Approximate $\hat{\beta}_1$ for Fast Leave-One-Out Cross-Validated nPC-LDA

Let Z be an $n \times p$ predictor matrix where each row indicates an observation and each column indicates a feature. Let $Z = UDV^\intercal$ be the singular value decomposition of Z, where U is $n \times n$, D is $n \times n$, and V is $p \times n$. Let $W = ZV = UD$ be the projection of Z onto the right singular vectors, V. Let Y be an $n \times d$ class indicator matrix such that $Y_{ik} = 1$ if observation i is in class k and 0 otherwise. In this section, we let $i = 1, \ldots, n$ index observations, and $k = 1, \ldots, d$ index classes.

Let n_k be the number of training observations in class k and $\hat{\pi}_k = \frac{n_k}{n}$ be the proportion of training observations in class k. We let $\hat{\pi}_k$ be the estimated prior probability of an observation belonging to class k and $\hat{\pi} = (\hat{\pi}_1 \ldots \hat{\pi}_d)^\intercal$ be the vector of all estimated prior probabilities such that $\hat{\pi_1} + \cdots + \hat{\pi}_d = 1$. Further, let $\hat{\mu}_k = \frac{1}{n_k} \sum_{i:Y_{ik}=1} W_i$ be the estimated mean vector for class k. We note

$$\begin{bmatrix} \hat{\mu}_1 \\ \vdots \\ \hat{\mu}_d \end{bmatrix} = (Y^\intercal Y)^{-1} Y^\intercal W = O_Y W,$$

where O_Y is the ordinary least squares operator of Y. Further, let

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i:Y_{ik}=1} (W_i - \hat{\mu}_k)^\intercal (W_i - \hat{\mu}_k),$$

and define

$$\hat{\Sigma} = \sum_{k=1}^{d} \frac{n_k}{n} \hat{\Sigma}_k.$$

We can also write

$$\hat{\Sigma} = \frac{1}{n} W^\intercal R_Y W$$

where $R_Y = I - Y(Y^\intercal Y)^{-1} Y^\intercal$ is the residual operator of Y. We note, however, that $\hat{\Sigma}$ has rank $n - d$ and is therefore singular. To create an invertible modified covariance matrix, we replace the d null eigenvalues of $\hat{\Sigma}$ with a small, non-zero value, λ. In practice, we choose λ to be the median of the eigenvalues of $G = ZZ^\intercal$. The resulting regularized covariance matrix is defined as

$$\tilde{\Sigma} = \hat{\Sigma} + \lambda P_{W^{-1}Y},$$

where $P_{W^{-1}Y}$ is the projection operator of $W^{-1}Y$ and is defined as

$$\begin{aligned} P_{W^{-1}Y} &= P_{(UD)^{-1}Y} \\ &= D^{-1}U^{-1}Y(Y^\intercal U^{-\intercal} D^{-\intercal} D^{-1} U^{-1} Y)^{-1} Y^\intercal U^{-\intercal} D^{-\intercal} \\ &= D^{-1}U^{-1}Y(Y^\intercal U D^{-2} U^{-1} Y)^{-1} Y^\intercal U D^{-1}. \end{aligned}$$

With these definitions, we can now construct our nPC-LDA classifier. Let $\tilde{Z}$ be an out-of-sample observation. We assign $\tilde{Z}$ to the class that produces the largest nPC-LDA score; the largest score indicates that $\tilde{Z}$ is closest to the training observations in that class. We obtain the $d \times 1$ nPC-LDA score vector for $\tilde{Z}$ by calculating $\tilde{Z}\hat{\beta}_1 + \hat{\beta}_0$, where $\hat{\beta}_1$ and $\hat{\beta}_0$ are estimated from the in-sample (training) data. Specifically, $\hat{\beta}_1 = V\tilde{\Sigma}^{-1}\hat{\mu}$ and $\hat{\beta}_0 = log(\hat{\pi}) - \frac{1}{2}\hat{\mu}^\intercal \tilde{\Sigma}^{-1}\hat{\mu}$. In this section, we are only interested in the calculation of $\hat{\beta}_1$, and thus note:

$$\begin{aligned} \tilde{Z}\hat{\beta}_1 &= \tilde{Z}V\tilde{\Sigma}^{-1}\hat{\mu} \\ &= \tilde{Z}V[\frac{1}{n}W^\intercal R_Y W + \lambda P_{W^{-1}Y}]^{-1}(O_Y W)^\intercal \\ &= n\tilde{Z}V(DU^\intercal)[(UD)W^\intercal R_Y W(DU^\intercal) + (n\lambda)(UD)P_{W^{-1}Y}(DU^\intercal)]^{-1}(UD)W^\intercal O_Y^\intercal \\ &= n\tilde{Z}Z^\intercal[(ZZ^\intercal)R_Y(ZZ^\intercal) + (n\lambda)UDP_{W^{-1}Y}DU^\intercal]^{-1}(ZZ^\intercal)O_Y^\intercal \\ &= n\tilde{Z}Z^\intercal[(ZZ^\intercal)R_Y(ZZ^\intercal) + \\ &\qquad (n\lambda)UDD^{-1}U^{-1}Y(Y^\intercal U D^{-2} U^{-1} Y)^{-1} Y^\intercal U D^{-1} DU^\intercal]^{-1}(ZZ^\intercal)O_Y^\intercal \end{aligned}$$

$$= n\tilde{Z}Z^\intercal[(ZZ^\intercal)R_Y(ZZ^\intercal) + (n\lambda)Y(Y^\intercal(ZZ^\intercal)^{-1}Y)^{-1}Y^\intercal]^{-1}(ZZ^\intercal)O_Y^\intercal$$
$$= n\underbrace{\tilde{Z}Z^\intercal(ZZ^\intercal)^{-1}}_{P1}\underbrace{[R_Y + (n\lambda)(ZZ^\intercal)^{-1}Y(Y^\intercal(ZZ^\intercal)^{-1}Y)^{-1}Y^\intercal(ZZ^\intercal)^{-1}]^{-1}}_{P2}\underbrace{O_Y^\intercal}_{P3}$$

We can further simplify $P2$ by using the Woodbury matrix inverse identity. Let $H = Y(Y^\intercal Y)^{-1/2}$ and $K = \sqrt{n\lambda}(ZZ^\intercal)^{-1}Y(Y^\intercal(ZZ^\intercal)^{-1}Y)^{-1/2}$ and define

$$A = \begin{bmatrix} -H & K \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} H^\intercal \\ K^\intercal \end{bmatrix}.$$

Then we can write

$$\begin{aligned} P2 &= [I_n - Y(Y^\intercal Y)^{-1}Y^\intercal + (n\lambda)(ZZ^\intercal)^{-1}Y(Y^\intercal(ZZ^\intercal)^{-1}Y)^{-1}Y^\intercal(ZZ^\intercal)^{-1}]^{-1} \\ &= [I_n - HH^\intercal + KK^\intercal]^{-1} \\ &= [I_n + AI_{2d}B]^{-1} \\ &= I_n - A(I_{2d} + BA)^{-1}B \qquad \text{(by Woodbury),} \end{aligned}$$

which gives us

$$\tilde{Z}\hat{\beta}_1 = n\underbrace{\tilde{Z}Z^\intercal(ZZ^\intercal)^{-1}}_{P1}\underbrace{[I_n - A(I_{2d} + BA)^{-1}B]}_{P2}\underbrace{O_Y^\intercal}_{P3}.$$

Now, let us consider the leave-one-out (LOO) setting where the out-of-sample observation $\tilde{Z} = Z_i$ and the training data $Z = Z_{-i}$. In this setting, an approximate nPC-LDA score, $Z_i\hat{\beta}_1$, is easy to calculate. In particular, in Section D.1.1, we show that the $P1$ term $Z_iZ_{-i}^\intercal(Z_{-i}Z_{-i}^\intercal)^{-1}$ simplifies to $\frac{-1}{G_{ii}^{-1}}G_i^{-\intercal}D_i$, where $G = ZZ^\intercal$ and D_i is the $n \times n$ identity matrix with the i^{th}column removed. Further, in Section D.1.2, we show that the matrix $(I_{2d}+BA)^{-1}BO_Y^\intercal$ is stable whether it is calculated in a LOO manner or with the full data. Therefore, we can approximate the calculation of $Z_i\hat{\beta}_1$ in a pseudo-LOO way that improves computation time.

D.1.1 Calculating $Z_iZ_{-i}^\intercal(Z_{-i}Z_{-i}^\intercal)^{-1}$

Recall Z is an $n \times p$ predictor matrix and $G = ZZ^\intercal$. Let g_{0i} be the i^{th} column of G with the i^{th} element set to 0, and define U_i and C as follows:

$$g_{0i} = \begin{bmatrix} g_{i,1} \\ \vdots \\ g_{i,i-1} \\ 0 \\ g_{i,i+1} \\ \vdots \\ g_{i,n} \end{bmatrix} \qquad U_i = \begin{bmatrix} g_{i,1} & 0 \\ \vdots & \vdots \\ g_{i,i-1} & 0 \\ 0 & 1 \\ g_{i,i+1} & 0 \\ \vdots & \vdots \\ g_{i,n} & 0 \end{bmatrix} = \begin{bmatrix} | & | \\ g_{0i} & e_i \\ | & | \end{bmatrix} \qquad C = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}.$$

Further, define D_i as the $n \times n$ identity matrix with the i^{th}column removed; therefore, D_i is an $n \times n-1$ matrix.

Here, we show that $Z_i Z_{-i}^\intercal (Z_{-i} Z_{-i}^\intercal)^{-1} = \frac{-1}{G_{ii}^{-1}} G_i^{-\intercal} D_i$. To begin, we consider the matrix $G + U_i C U_i^\intercal$ and notice it is equivalent to G with the elements in the i^{th} row and column zeroed out (except element (i,i)). It follows that $(G + U_i C U_i^\intercal)^{-1}_{-i-i} = (G_{-i-i})^{-1}$, where the subscript $_{-i-i}$ indicates a matrix with the i^{th} row and column removed. Therefore, we can write:

$$\begin{aligned} Z_i Z_{-i}^\intercal (Z_{-i} Z_{-i}^\intercal)^{-1} &= g_{i,-i}^\intercal (G_{-i-i})^{-1} \\ &= g_{i,-i}^\intercal (G + U_i C U_i^\intercal)^{-1}_{-i-i} \\ &= g_{0i}^\intercal (G + U_i C U_i^\intercal)^{-1} D_i \\ &= g_{0i}^\intercal \left[G^{-1} - G^{-1} U_i (C^{-1} + U_i^\intercal G^{-1} U_i)^{-1} U_i^\intercal G^{-1} \right] D_i, \end{aligned}$$

where $g_{i,-i}$ is g_{0i} with the i^{th} element removed. The last equality follows from the Woodbury matrix inverse identity. To get the desired result, we must further simplify this expression. We start by calculating $(C^{-1} + U_i^\intercal G^{-1} U_i)^{-1}$.

$$G^{-1} g_{0i} = G^{-1} \begin{bmatrix} g_{i1} \\ \vdots \\ g_{ii} \\ \vdots \\ g_{in} \end{bmatrix} - G^{-1} \begin{bmatrix} 0 \\ \vdots \\ g_{ii} \\ \vdots \\ 0 \end{bmatrix} = e_i - G_i^{-1} g_{ii}$$

$$G^{-1} U_i = G^{-1} \begin{bmatrix} | & | \\ g_{0i} & e_i \\ | & | \end{bmatrix} = \begin{bmatrix} | & | \\ e_i - G_i^{-1} g_{ii} & G_i^{-1} \\ | & | \end{bmatrix}$$

$$U_i^\intercal G^{-1} U_i = \begin{bmatrix} g_{i1} & \dots & 0 & \dots & g_{in} \\ 0 & \dots & 1 & \dots & 0 \end{bmatrix} \begin{bmatrix} & | & \\ e_i - G_i^{-1} g_{ii} & | & G_i^{-1} \\ & | & \end{bmatrix}$$

$$= \begin{bmatrix} -g_{0i}{}^\intercal G_i^{-1} g_{ii} & -g_{0i}{}^\intercal G_i^{-1} \\ 1 - G_{ii}^{-1} g_{ii} & G_{ii}^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} -(1 - G_{ii}^{-1} g_{ii}) g_{ii} & 1 - G_{ii}^{-1} g_{ii} \\ 1 - G_{ii}^{-1} g_{ii} & G_{ii}^{-1} \end{bmatrix}$$

The final equality follows because $g_{0i}{}^\intercal G_i^{-1} = g_i G_i^{-1} - g_{ii} G_{ii}^{-1} = 1 - g_{ii} G_{ii}^{-1}$.
We recognize that $C^{-1} = C$ and can write:

$$U_i^\intercal G^{-1} U_i + C^{-1} = \begin{bmatrix} -(1 - G_{ii}^{-1} g_{ii}) g_{ii} & -G_{ii}^{-1} g_{ii} \\ -G_{ii}^{-1} g_{ii} & G_{ii}^{-1} \end{bmatrix}.$$

Next, we calculate the determinant of this matrix and then invert it:

$$|U_i^\intercal G^{-1} U_i + C^{-1}| = G_{ii}^{-1} g_{ii} (G_{ii}^{-1} g_{ii} - 1) - (G_{ii}^{-1} g_{ii})^2 = -G_{ii}^{-1} g_{ii}$$

$$(U_i^\intercal G^{-1} U_i + C^{-1})^{-1} = \frac{-1}{G_{ii}^{-1} g_{ii}} \begin{bmatrix} G_{ii}^{-1} & G_{ii}^{-1} g_{ii} \\ G_{ii}^{-1} g_{ii} & -(1 - G_{ii}^{-1} g_{ii}) g_{ii} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-1}{g_{ii}} & -1 \\ -1 & \frac{(1 - G_{ii}^{-1} g_{ii})}{G_{ii}^{-1}} \end{bmatrix}$$

We have obtained an expression for $(U_i^\intercal G^{-1} U_i + C^{-1})^{-1}$, and we now focus on calculating the remaining expressions needed for the full calculation.

$$U_i^\intercal G^{-1} D_i = \begin{bmatrix} - & e_i^\intercal - G_i^{-\intercal} g_{ii} & - \\ - & G_i^{-\intercal} & - \end{bmatrix} D_i = \begin{bmatrix} -g_{ii} \\ 1 \end{bmatrix} G_i^{-\intercal} D_i$$

The last equality follows because multiplying a matrix by D_i ignores the i^{th} column, which is the only column where e_i is non-zero.

$$(U_i^\intercal G^{-1} U_i + C^{-1})^{-1} U_i^\intercal G^{-1} D_i = \begin{bmatrix} \frac{-1}{g_{ii}} & -1 \\ -1 & \frac{(1-G_{ii}^{-1} g_{ii})}{G_{ii}^{-1}} \end{bmatrix} \begin{bmatrix} -g_{ii} \\ 1 \end{bmatrix} G_i^{-\intercal} D_i$$
$$= \begin{bmatrix} 0 \\ g_{ii} + \frac{(1-G_{ii}^{-1} g_{ii})}{G_{ii}^{-1}} \end{bmatrix} G_i^{-\intercal} D_i$$
$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{G_i^{-\intercal} D_i}{G_{ii}^{-1}}$$

$$g_{0i}^\intercal G^{-1} U_i = \begin{bmatrix} -(1 - g_{ii} G_{ii}^{-1}) g_{ii} & 1 - g_{ii} G_{ii}^{-1} \end{bmatrix}$$

$$g_{0i}^\intercal G^{-1} U_i (U_i^\intercal G^{-1} U_i + C^{-1})^{-1} U_i^\intercal G^{-1} D_i = (1 - g_{ii} G_{ii}^{-1}) \frac{G_i^{-\intercal} D_i}{G_{ii}^{-1}}$$

$$g_{0i}^\intercal G^{-1} D_i = (e_i^\intercal - G_i^{-\intercal} g_{ii}) D_i = -g_{ii} G_i^{-\intercal} D_i$$

$$g_{0i}^\intercal [G^{-1} - G^{-1} U_i (U_i^\intercal G^{-1} U_i + C^{-1})^{-1} U_i^\intercal G^{-1}] D_i = -g_{ii} G_i^{-\intercal} D_i - (1 - g_{ii} G_{ii}^{-1}) \frac{G_i^{-\intercal} D_i}{G_{ii}^{-1}}$$
$$= \frac{-1}{G_{ii}^{-1}} G_i^{-\intercal} D_i$$

Thus, we have successfully shown that $Z_i Z_{-i}^\intercal (Z_{-i} Z_{-i}^\intercal)^{-1} = \frac{-1}{G_{ii}^{-1}} G_i^{-\intercal} D_i$. This result makes it quite fast and simple to obtain $Z_i Z_{-i}^\intercal (Z_{-i} Z_{-i}^\intercal)^{-1}$ at each LOO iteration when the matrix G^{-1} is already known.

D.1.2 Stability of $(I_{2d} + BA)^{-1} B O_Y^\intercal$

As derived above, we can calculate the nPC-LDA scores for an out-of-sample observation $\tilde{Z}$ as follows:

$$\tilde{Z} \hat{\beta}_1 = n \tilde{Z} Z^\intercal (Z Z^\intercal)^{-1} [I_n - A(I_{2d} + BA)^{-1} B] O_Y^\intercal.$$

In this section, we focus on speeding up the calculations of $(I_{2d} + BA)^{-1} B] O_Y^\intercal$ within the LOO loop. We notice that all of the matrices involved in this calculation are fairly low-dimensional: in the full data setting, B is $2d \times n$, A is $n \times 2d$, and O_Y is $d \times n$. The resulting matrix has dimension

$2d \times d$ and does not grow with n. This indicates that the values in $(I_{2d} + BA)^{-1}BO_Y^\intercal$ will not substantially change whether this matrix is calculated with the full data (n samples) or with the full data minus the i^{th} observation ($n-1$ samples). Indeed, we have observed this stability to be true for simulated data. In particular, we let $\hat{\beta}_{1,(-i)}$ be the estimated value of β_1 calculated without observation i in the i^{th} LOO CV iteration. We have found that the values of $Z_i\hat{\beta}_{1,(-i)}$ do not meaningfully differ whether $(I_{2d} + BA)^{-1}BO_Y^\intercal$ is calculated with n or with $n-1$ samples. Therefore, we do not need to separately calculate this term at each CV iteration. Instead, we calculate it once with the full data and insert it into the $Z_i\hat{\beta}_{1,(-i)}$ calculation at each CV iteration for our FAST-CV algorithm.

D.2 Approximate $\hat{\beta}_0$ for Fast Leave-One-Out Cross-Validated nPC-LDA

As described in Section D.1, the nPC-LDA score for an out-of-sample observation, $\tilde{Z}$, is calculated as $\tilde{Z}\hat{\beta}_1 + \hat{\beta}_0$. In this section, we are interested in the calculation of $\hat{\beta}_0 = log(\hat{\pi}) - \frac{1}{2}\hat{\mu}^\intercal\tilde{\Sigma}^{-1}\hat{\mu}$, where $\hat{\mu}$ is the $p \times d$ sample mean feature matrix and $\tilde{\Sigma}$ is the $p \times p$ modified sample covariance matrix.

When $\hat{\beta}_0$ is calculated via this formula within the LOO cross-validation loop, however, it can reduce classification accuracy. Consider observation i. Without loss of generality, let this observation belong to class 1. For the i^{th} LOO CV iteration, observation i will be left out of model training. The sample mean of the features for class 1, therefore, will be calculated from the remaining $n_1 - 1$ class 1 observations and will be anti-correlated with observation i. Specifically, if the values in Z_i are large and positive, the values in the sample mean feature vector for class 1 will decrease when Z_i is excluded. This anti-correlation will decrease the cross-validation classification accuracy.

Therefore, instead of estimating β_0 within the LOO loop, we calculate it with the full data. This means that the $\hat{\beta}_0$ used for predictions for out-of-sample observations is the same value used to obtain the LOO scores. Specifically, we calculate

$$\hat{\beta}_0 = log(\hat{\pi}) - \frac{1}{2}O_Y G\hat{b}_1,$$

where $O_Y Z = \hat{\mu}$ and $\hat{b}_1 = n(ZZ^\intercal)^{-1}[I_n - A(I_{2d} + BA)^{-1}B]O_Y^\intercal$ such that $Z^\intercal\hat{b}_1 = \hat{\beta}_1$ as in Section D.1.

D.3 Additional Simulation Results

In our simulations, we need to manage lengthy computation times. Therefore, when assessing CV accuracy rates, we vary the number of replicates with n: 3000 replicates were performed for $n = 6$, 1800 replicates for $n = 10$, 900 for $n = 20$, 300 for $n = 60$, 180 for $n = 100$, 90 for $n = 200$, 36 for $n = 500$, and 18 for $n = 1000$. These numbers of replicates were selected so that $n \times Nrep = 18000$. Despite these precautions, we were unable to perform accuracy simulations for "correct" LOO CV for PC-LDA with $r = 3$ and $r = 50$ principal components due to impractical computation needs.

When using simulations to assess computation time, we take the median time across 10 replicates for all values of n and p. If the median computation time for a classification method, for given values of n and p, exceeds 2,000 seconds, we do not test that method at the next value of n and the same value of p.

In the main text, we present simulation results for $p = 20{,}000$. Here, we present additional results for $p = 100{,}000$ and 500,000; this allows us to evaluate how the performance of nPC-LDA, with both LOO CV and FAST-CV, and PC-LDA with a fixed number of PCs and various CV schemes varies with p in addition to how it varies with n. All tested methods show largely the same behavior for every value of p we examined.

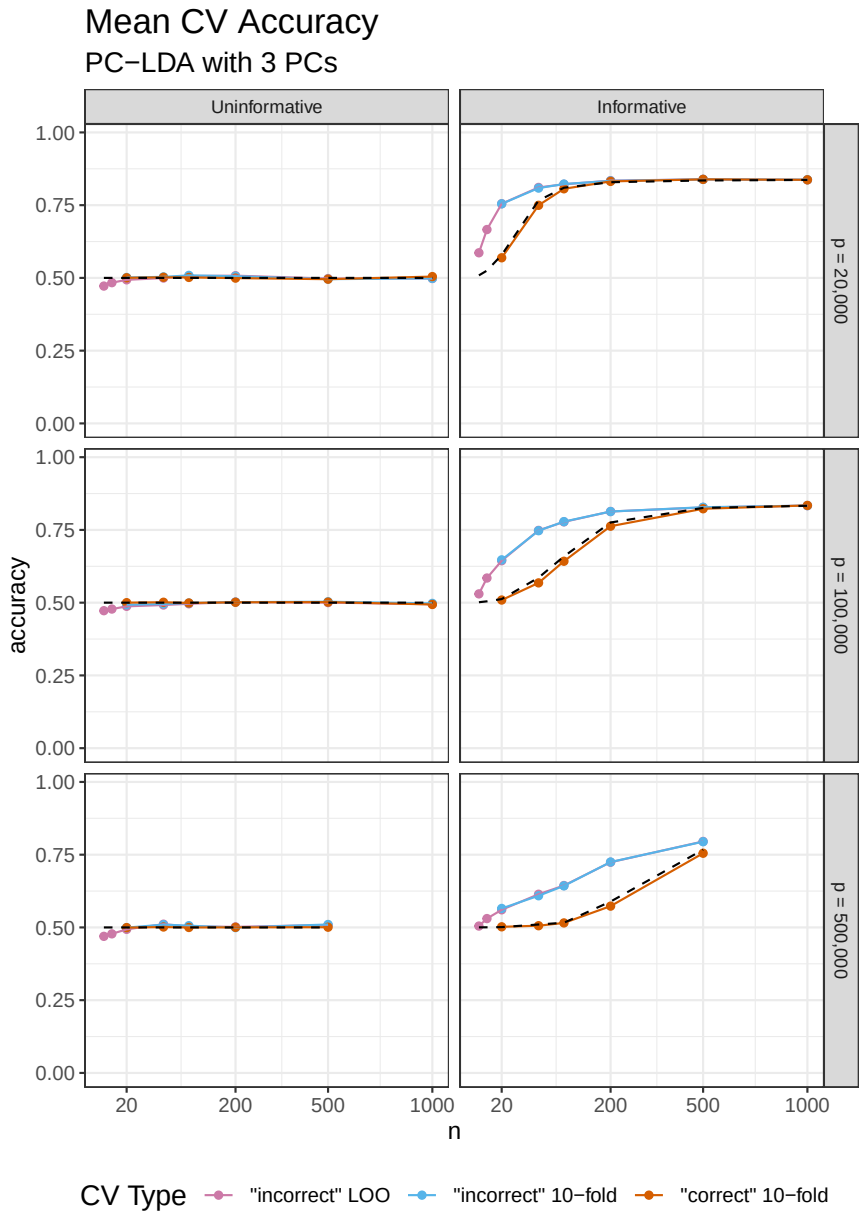

Figure D.1: Mean CV accuracies for a PC-LDA classifier with $r = 3$ and $p = 20{,}000$, $p = 100{,}000$, and $p = 500{,}000$. Sample size (n) is displayed on a square-root scale. The black dashed lines indicate the theoretical classification accuracy rates for a PC-LDA classifier with 3 principal components built on the generated data. "Incorrect" CV involves only calculating the principal components on the full data, while "correct" CV involves calculating the principal components for each individual CV training set. Note: 10-fold CV cannot be run for small sample sizes. Also, the simulations for $n = 1000$ and $p = 500,000$ were too memory-intensive to complete.

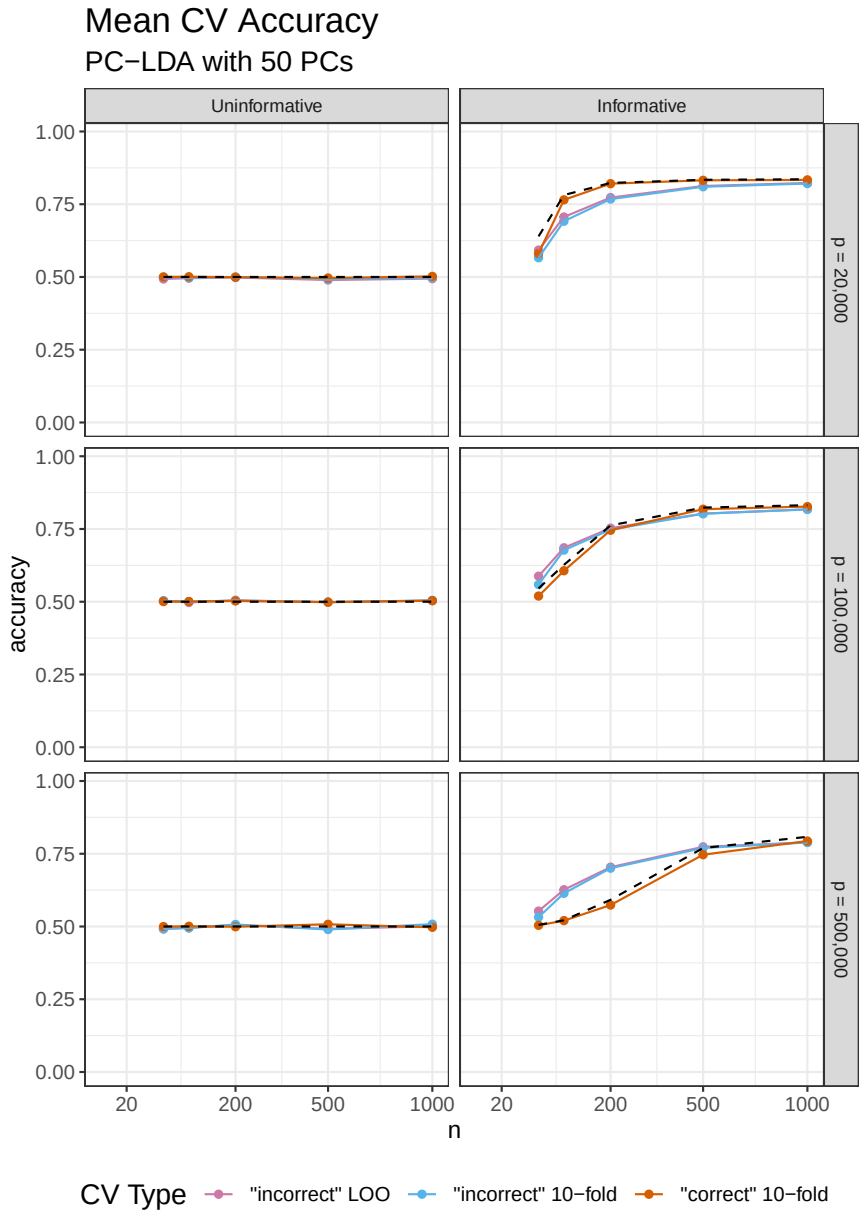

Figure D.2: Mean CV accuracies for a PC-LDA classifier with $r = 50$ and $p = 20{,}000$, $p = 100{,}000$, and $p = 500{,}000$. Sample size (n) is displayed on a square-root scale. The black dashed lines indicate the theoretical classification accuracy rates for a PC-LDA classifier with 50 principal components built on the generated data. "Incorrect" CV involves only calculating the principal components on the full data, while "correct" CV involves calculating the principal components for each individual CV training set. Note: PC-LDA with 50 PCs cannot be run for sample sizes smaller than 50.

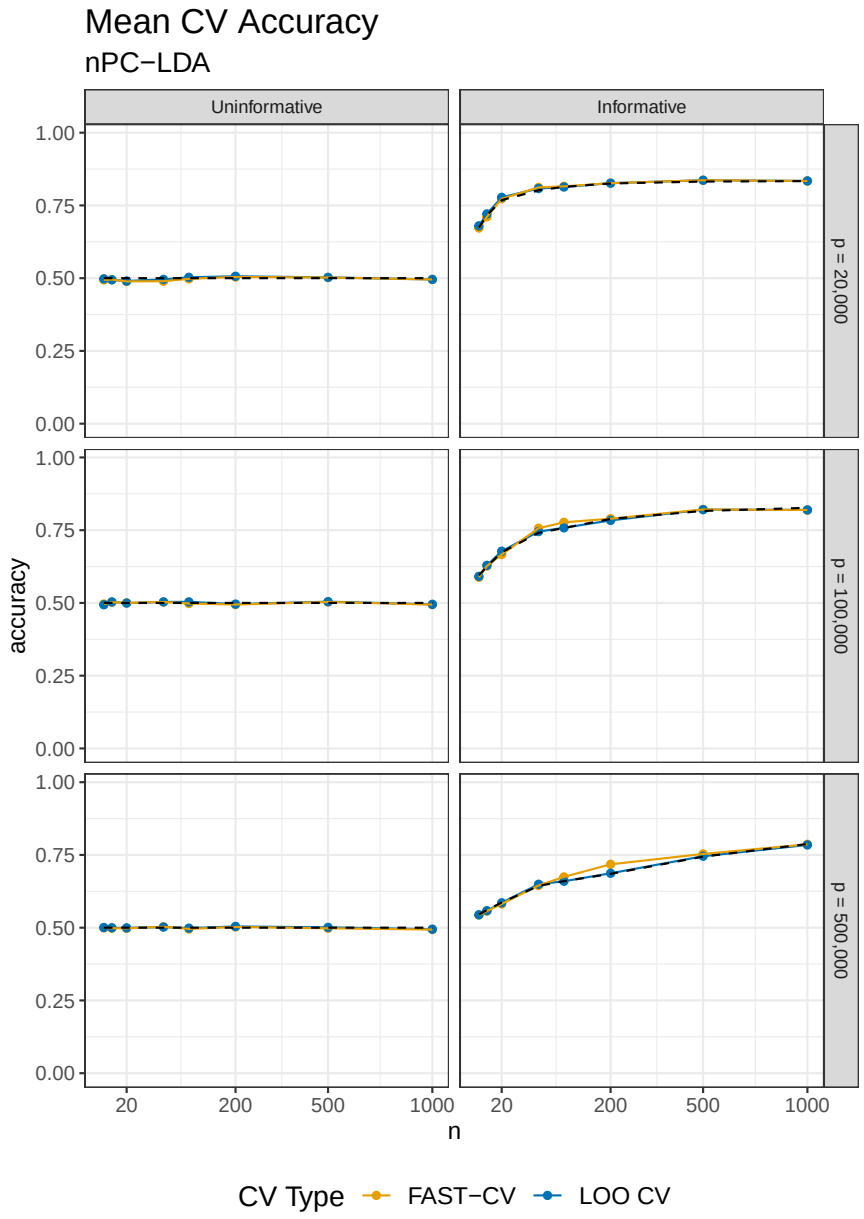

Figure D.3: Mean CV accuracies for nPC-LDA obtained via LOO CV and FAST-CV for $p = 20{,}000$, $p = 100{,}000$, and $p = 500{,}000$. Sample size (n) is displayed on a square-root scale. The black dashed lines indicate the theoretical classification accuracy rates for an nPC-LDA classifier built on the generated data.

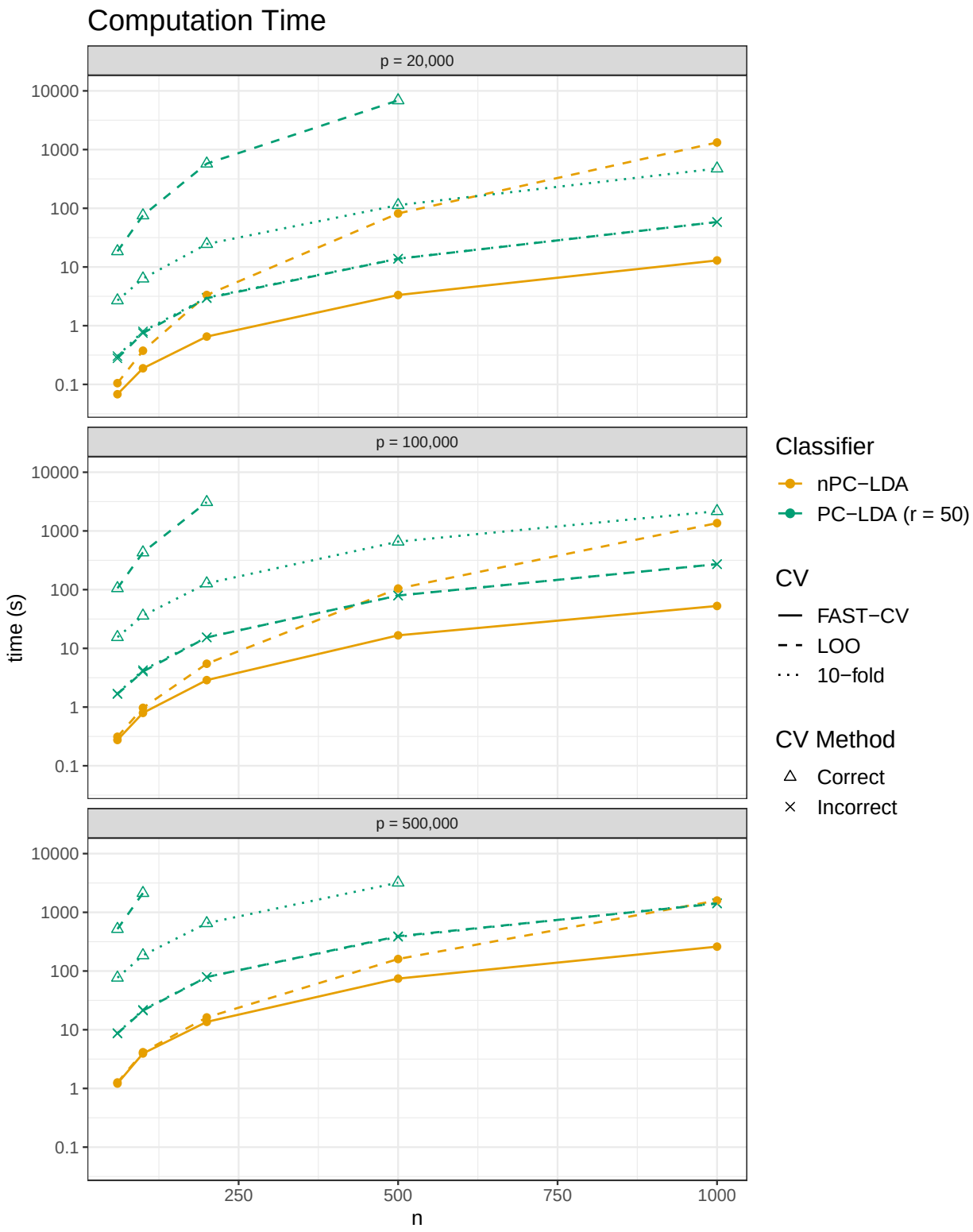

Figure D.4: Median computation times for $p = 20{,}000$, $p = 100{,}000$, and $p = 500{,}000$. Computation time (in seconds) is displayed on the $\log_{10}$ scale. The most computationally intensive algorithms were only run at small sample sizes.

D.4 Additional Pharmacogenomic Data Results

We obtained all real data from the Genomics of Drug Sensitivity in Cancer (GDSC) project (*Yang et al.*, 2013). For this application, we used version 8.2 of the GDSC2 data. This includes summarized drug efficacy values, gene expression levels, methylation values, and cell line tissue type and histology data, all retrieved from the GDSC website (https://www.cancerrxgene.org/downloads/bulk_download; March 2021).

D.4.1 Multiclass Tissue Type Classification

To test the performance of nPC-LDA with FAST-CV on multiclass data, we predicted two aspects of cancer cell line tumor type from gene expression data. We considered a tissue type label with 19 classes: non-small cell lung cancer (n = 109 cell lines), urogenital system (n = 102), leukemia (n = 82), aero-digestive tract (n = 79), lymphoma (n = 66), small cell lung cancer (n = 61), nervous system (n = 56), skin (n = 55), digestive system (n = 51), breast (n = 50), large intestine (n = 49), bone (n = 38), kidney (n = 33), pancreas (n = 32), neuroblastoma (n = 29), lung (n = 22), soft tissue (n = 21), myeloma (n = 18), and thyroid (n = 15). We also considered a tumor histology label with 11 classes: carcinoma (n = 566 cell lines), lymphoid neoplasm (n = 125), glioma (n = 52), malignant melanoma (n = 52), haematopoietic neoplasm (n = 40), neuroblastoma (n = 31), Ewing's sarcoma/peripheral primitive neuroectodermal tumour (n = 22), mesothelioma (n = 21), osteosarcoma (n = 10), rhabdomyosarcoma (n = 8), and chondrosarcoma (n = 5). We removed 18 histologies with fewer than 5 cell lines each.

The nPC-LDA algorithm produces similar estimates of model performance metrics whether implemented with classical LOO CV or with FAST-CV. This is true for both tissue type (estimated confusion matrix in Figure D.1) and histology (estimated confusion matrix in Figure D.2). The FAST-CV method achieves this model performance with substantially shorter computation times than the other tested methods (Table D.3).

Overall, nPC-LDA with FAST-CV performs well in settings with large numbers of classes and with large amounts of class imbalance, in terms of both classifier performance and computation times.

(a)

Observed \ Predicted	aerodigestive tract	bone	breast	digestive system	kidney	large intestine	leukemia	lung	NSCLC	SCLC	lymphoma	myeloma	nervous system	neuroblastoma	pancreas	skin	soft tissue	thyroid	urogenital system
aerodigestive tract	**66**	0	0	3	0	1	0	0	6	0	0	0	0	0	2	0	0	0	1
bone	0	**33**	0	0	0	0	0	0	1	0	0	0	2	0	0	0	1	0	1
breast	2	1	**38**	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	6
digestive system	0	0	0	**26**	0	4	0	0	8	1	1	0	0	0	4	0	0	0	7
kidney	0	0	0	0	**28**	0	0	0	1	0	0	0	0	0	0	0	0	0	4
large intestine	1	0	0	5	0	**39**	0	0	1	0	0	0	0	0	0	0	1	0	2
leukemia	0	0	0	0	0	0	**65**	0	0	0	17	0	0	0	0	0	0	0	0
lung	0	0	0	0	1	0	0	**17**	1	1	0	0	0	0	0	0	0	0	2
NSCLC	6	0	1	0	0	0	0	1	**89**	5	0	0	0	0	3	0	0	0	4
SCLC	0	0	0	0	0	0	0	0	6	**50**	0	0	1	0	0	0	0	0	4
lymphoma	0	0	0	0	0	0	3	0	1	0	**60**	2	0	0	0	0	0	0	0
myeloma	0	0	0	0	0	0	0	0	0	0	3	**15**	0	0	0	0	0	0	0
nervous system	1	0	0	0	0	0	0	0	0	0	0	0	**50**	0	0	1	2	0	2
neuroblastoma	0	0	0	0	0	0	0	0	0	0	0	0	1	**27**	0	0	0	0	1
pancreas	0	0	0	1	0	0	0	0	3	0	0	0	0	0	**27**	0	0	0	1
skin	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**47**	2	1	2
soft tissue	0	2	0	1	1	0	0	0	0	0	0	0	2	0	0	0	**9**	1	5
thyroid	0	0	0	0	0	0	0	0	4	1	0	0	1	0	0	0	0	**5**	4
urogenital system	9	0	0	2	1	1	0	1	5	2	0	0	0	0	1	0	0	2	**78**

(b)

Observed \ Predicted	aerodigestive tract	bone	breast	digestive system	kidney	large intestine	leukemia	lung	NSCLC	SCLC	lymphoma	myeloma	nervous system	neuroblastoma	pancreas	skin	soft tissue	thyroid	urogenital system
aerodigestive tract	**65**	0	0	3	0	1	0	0	6	0	0	0	0	0	1	0	0	0	3
bone	0	**33**	0	0	0	0	0	0	1	0	0	0	2	0	0	0	1	0	1
breast	2	1	**40**	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	5
digestive system	1	0	0	**23**	0	5	1	0	9	1	0	0	0	0	5	0	0	0	6
kidney	0	0	0	0	**29**	0	0	0	1	0	0	0	0	0	0	0	0	0	3
large intestine	1	0	0	5	0	**39**	0	0	1	0	0	0	0	0	0	0	1	0	2
leukemia	0	0	0	0	0	0	**65**	0	0	0	17	0	0	0	0	0	0	0	0
lung	0	0	0	0	1	0	0	**19**	1	1	0	0	0	0	0	0	0	0	0
NSCLC	6	0	1	0	0	0	0	1	**92**	5	0	0	0	0	2	0	0	0	2
SCLC	0	0	0	0	0	0	0	0	6	**51**	0	0	1	0	0	0	0	0	3
lymphoma	0	0	0	0	0	0	4	0	1	0	**59**	2	0	0	0	0	0	0	0
myeloma	0	0	0	0	0	0	0	0	0	0	4	**14**	0	0	0	0	0	0	0
nervous system	1	0	0	0	0	0	0	0	0	0	0	0	**51**	0	0	1	2	0	1
neuroblastoma	0	0	0	0	0	0	0	0	0	0	0	0	1	**27**	0	0	0	0	1
pancreas	0	0	0	1	0	0	0	0	3	0	0	0	0	0	**28**	0	0	0	0
skin	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**47**	2	1	2
soft tissue	0	3	0	1	1	0	0	0	0	0	0	0	1	0	0	0	**7**	2	6
thyroid	0	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	1	**7**	3
urogenital system	7	0	0	2	1	1	0	1	5	2	0	0	1	0	1	0	1	2	**78**

Table D.1: Estimated confusion matrix for classification via nPC-LDA with **(a)** FAST-CV and **(b)** LOO CV to predict tumor tissue type ($k = 19$ classes) from gene expression levels in the GDSC study ($n = 968$ cell lines). The nPC-LDA algorithm performs well, even when there are a large number of classes.

	Predicted	carcinoma	chondrosarcoma	Ewings sarcoma/pPNET	glioma	haematopoietic neoplasm	lymphoid neoplasm	malignant melanoma	mesothelioma	neuroblastoma	osteosarcoma	rhabdomyosarcoma
(a)	carcinoma	**561**	1	0	0	0	2	1	1	0	0	0
	chondrosarcoma	2	**2**	0	1	0	0	0	0	0	0	0
	Ewings sarcoma/pPNET	0	0	**22**	0	0	0	0	0	0	0	0
	glioma	2	0	0	**49**	0	0	1	0	0	0	0
	haematopoietic neoplasm	0	0	0	0	**34**	6	0	0	0	0	0
	lymphoid neoplasm	1	0	0	0	1	**123**	0	0	0	0	0
	malignant melanoma	5	0	0	0	0	0	**47**	0	0	0	0
	mesothelioma	4	0	0	0	0	0	0	**17**	0	0	0
	neuroblastoma	1	0	1	1	0	0	1	0	**27**	0	0
	osteosarcoma	4	0	0	0	0	0	0	0	0	**6**	0
	rhabdomyosarcoma	3	0	1	0	0	0	0	0	0	0	**4**
(b)	carcinoma	**561**	1	0	0	0	2	1	1	0	0	0
	chondrosarcoma	1	**2**	0	1	0	0	0	0	0	0	1
	Ewings sarcoma/pPNET	0	0	**22**	0	0	0	0	0	0	0	0
	glioma	1	0	0	**50**	0	0	1	0	0	0	0
	haematopoietic neoplasm	0	0	0	0	**34**	6	0	0	0	0	0
	lymphoid neoplasm	1	0	0	0	1	**123**	0	0	0	0	0
	malignant melanoma	5	0	0	0	0	0	**47**	0	0	0	0
	mesothelioma	3	0	0	0	0	0	0	**18**	0	0	0
	neuroblastoma	1	0	1	1	0	0	1	0	**27**	0	0
	osteosarcoma	4	0	0	0	0	0	0	0	0	**6**	0
	rhabdomyosarcoma	3	0	1	0	0	0	0	0	0	0	**4**

Table D.2: Estimated confusion matrix for classification via nPC-LDA with **(a)** FAST-CV and **(b)** LOO CV to predict tumor histology ($k = 11$ classes) from gene expression levels in the GDSC study ($n = 932$ cell lines). The nPC-LDA algorithm performs well even when classes are seriously imbalanced. Histologies with fewer than 5 observations were not included in this analysis. pPNET is a peripheral primitive neuroectodermal tumour.

Model	Performance Estimation	Time (s) Tissue Type	Histology
nPC-LDA	FAST-CV	16.5	15.2
nPC-LDA	LOO CV	1244.4	1097.7
Random Forest	OOB voting	458.9	343.7
`MASS::lda`	LOO CV	724.0	710.6
MLDA	10-fold CV	675.2	598.2

Table D.3: Computation speed for predicting cell line tissue type and histology from gene expression levels. The nPC-LDA algorithm with FAST-CV has substantially shorter computation times than the other tested methods.

D.4.2 Drug Efficacy Classification

In our drug efficacy classification analysis, we discretize the AUC estimates provided by GDSC into calls of "sensitive" and "insensitive" using a cutoff of 0.85. Drug-cell line combinations with an AUC ≥ 0.85 are labeled "insensitive" and drug-cell line combinations with an AUC < 0.85 are labeled "sensitive" (Figure D.5). This discretization causes many drugs to have seriously imbalanced classes. This makes it difficult to evaluate model performance, particularly when the features are not highly predictive. To combat this, for each drug, we down-sample the majority class in the training data to create a subset with balanced classes. We then use this subset to train the nPC-LDA algorithm and random forests. In our analysis, we considered the 147 drugs with at least 10 observations per class, after down-sampling.

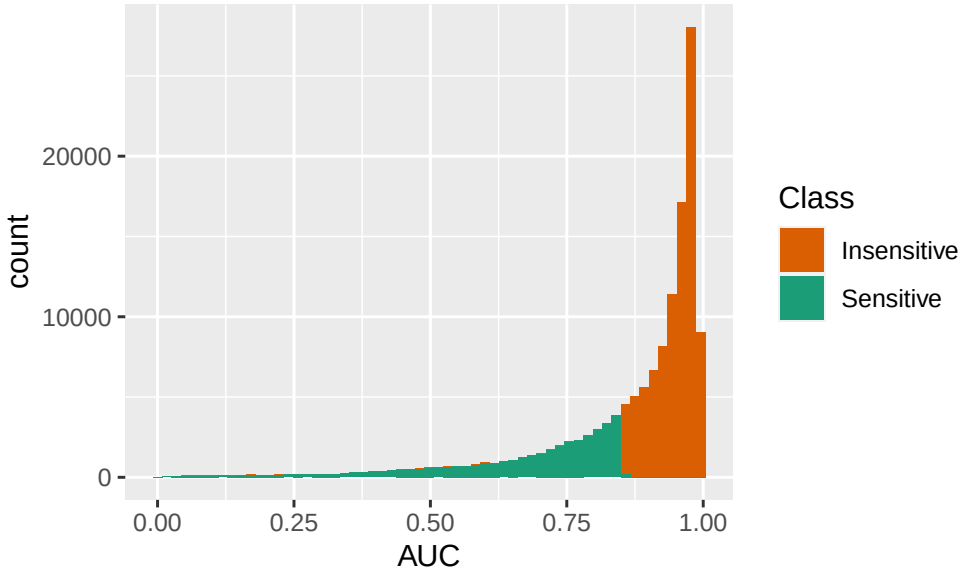

Figure D.5: Histogram of all GDSC-provided AUC estimates colored based on the discretized "sensitive" and "insensitive" classes.

D.4.3 Expanded Drug Efficacy Classification

The speed of nPC-LDA with FAST-CV lets us investigate how classifier performance varies with different response data and different predictor data.

Simple Binary Response First, we consider how varying the simple binary class threshold m affects classifier performance. We vary this threshold from 0.4 to 0.9 and label cell lines with AUCs smaller than m as "sensitive" and cell lines with AUCs larger than m as "insensitive". Table D.4 shows the median model performance for each threshold. In this analysis, the median is taken across the $n = 32$ drugs for which there were sufficient data at all tested thresholds, defined as at least 10 cell lines in each class. When the classes are imbalanced, we down-sample from the majority class to impose balance and improve the interpretability of model performance estimates. Figure D.6 shows boxplots of model performance at each tested threshold.

Binary Response without Moderate AUCs Next, we perform binary classification where cell lines with moderate AUC values are dropped from the analysis. Specifically, we define our binary classes as follows: cell lines are labeled "sensitive" if AUC $< m_1$ and "insensitive" if AUC $> m_2$, such that $m_1 < m_2$. Again, we focus on the $n = 32$ drugs with at least 10 cell lines in each class for all tested thresholds. We further down-sample from the majority class to impose balance. This technique tends to achieve better median model performance than simple binary classification (Figure D.7).

Methylation Data Finally, we predict drug efficacy (as binary response without moderate AUCs) from methylation data and from a concatenation of gene expression and methylation data (Table D.5, Figures D.8 and D.9). The combination of data sources does not perform better than gene expression data on its own.

Threshold	Accuracy	F_1-score	MCC
0.4	0.818	0.820	0.640
0.45	0.819	0.822	0.638
0.5	0.809	0.805	0.623
0.55	0.789	0.788	0.578
0.6	0.772	0.763	0.546
0.65	0.730	0.726	0.460
0.7	0.722	0.728	0.446
0.75	0.698	0.690	0.396
0.8	0.676	0.671	0.353
0.85	0.679	0.673	0.360
0.9	0.658	0.642	0.320

Table D.4: Median model performance estimated via FAST-CV for predicting binary drug efficacy from gene expression data with nPC-LDA. The binary class labels were assigned based on several AUC thresholds between 0.4 and 0.9. The stated model performance is the median across the $n = 32$ drugs for which there were sufficient data for all tested thresholds.

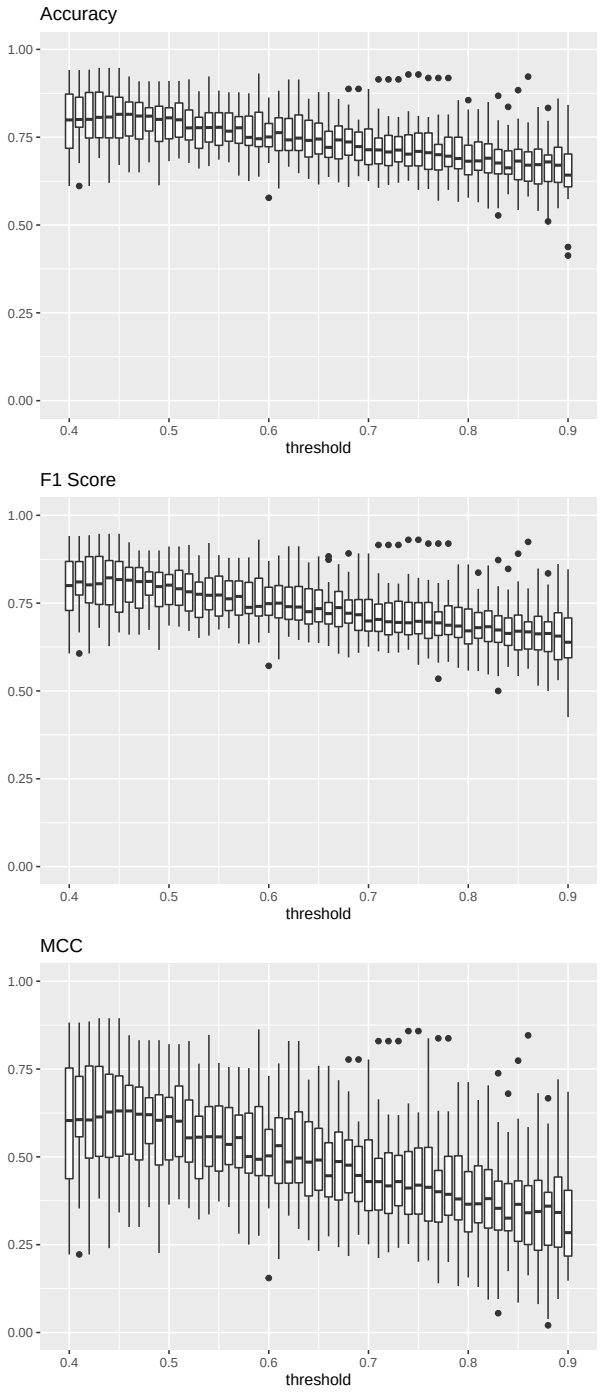

Figure D.6: Model performance estimated via FAST-CV for predicting binary drug efficacy from gene expression data with nPC-LDA. The binary class labels were assigned based on several AUC thresholds between 0.4 and 0.9. Model performance tends to increase as the class threshold decreases. The boxplots contain data for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds.

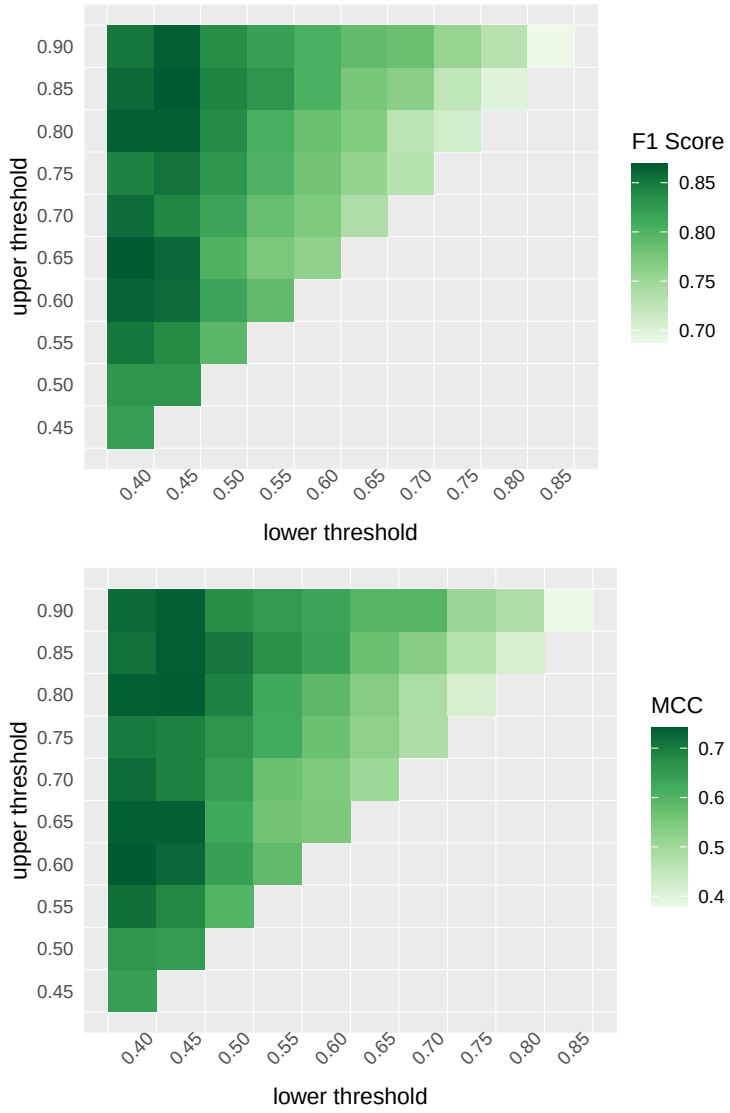

Figure D.7: Median nPC-LDA F_1-score and MCC estimated via FAST-CV for binary classification based on gene expression data. AUC values between the lower threshold (m_1) and the upper threshold (m_2) are discarded. Both measures of model performance tend to be better for smaller values of m_1. At each set of thresholds, we performed classification for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds. The median is taken across these drugs.

		Gene Expression			Methylation			Both		
m_1	m_2	Accuracy	F_1-score	MCC	Accuracy	F_1-score	MCC	Accuracy	F_1-score	MCC
0.40	0.45	0.820	0.820	0.643	0.800	0.800	0.600	0.821	0.827	0.643
0.40	0.50	0.828	0.829	0.658	0.812	0.808	0.625	0.824	0.828	0.655
0.40	0.55	0.857	0.851	0.715	0.808	0.805	0.616	0.812	0.812	0.630
0.40	0.60	0.868	0.863	0.742	0.833	0.828	0.668	0.836	0.843	0.673
0.40	0.65	0.866	0.870	0.734	0.844	0.839	0.689	0.844	0.839	0.689
0.40	0.70	0.855	0.857	0.719	0.850	0.851	0.704	0.875	0.875	0.750
0.40	0.75	0.849	0.845	0.702	0.857	0.849	0.722	0.844	0.839	0.689
0.40	0.80	0.862	0.866	0.735	0.852	0.848	0.707	0.857	0.857	0.742
0.40	0.85	0.856	0.859	0.714	0.833	0.837	0.676	0.861	0.857	0.728
0.40	0.90	0.858	0.851	0.722	0.833	0.833	0.671	0.875	0.875	0.750
0.45	0.50	0.824	0.830	0.650	0.824	0.824	0.647	0.825	0.829	0.651
0.45	0.55	0.841	0.839	0.684	0.825	0.825	0.656	0.828	0.844	0.670
0.45	0.60	0.862	0.859	0.725	0.845	0.849	0.695	0.840	0.846	0.682
0.45	0.65	0.864	0.861	0.734	0.824	0.816	0.647	0.826	0.829	0.659
0.45	0.70	0.847	0.841	0.695	0.849	0.850	0.699	0.833	0.837	0.681
0.45	0.75	0.843	0.854	0.694	0.853	0.848	0.707	0.845	0.844	0.691
0.45	0.80	0.866	0.866	0.738	0.845	0.843	0.691	0.844	0.840	0.700
0.45	0.85	0.868	0.869	0.737	0.826	0.820	0.651	0.845	0.840	0.693
0.45	0.90	0.865	0.867	0.734	0.843	0.851	0.688	0.845	0.833	0.693
0.50	0.55	0.798	0.794	0.600	0.810	0.800	0.622	0.829	0.821	0.660
0.50	0.60	0.819	0.815	0.642	0.828	0.819	0.656	0.818	0.816	0.640
0.50	0.65	0.810	0.800	0.623	0.811	0.803	0.623	0.810	0.811	0.624
0.50	0.70	0.818	0.816	0.645	0.824	0.822	0.651	0.833	0.833	0.667
0.50	0.75	0.831	0.829	0.665	0.817	0.821	0.646	0.816	0.821	0.646
0.50	0.80	0.845	0.838	0.691	0.803	0.800	0.616	0.816	0.808	0.640
0.50	0.85	0.852	0.843	0.707	0.833	0.836	0.668	0.843	0.841	0.685
0.50	0.90	0.837	0.836	0.675	0.828	0.821	0.657	0.853	0.841	0.712
0.55	0.60	0.791	0.790	0.586	0.775	0.776	0.564	0.778	0.785	0.561
0.55	0.65	0.780	0.774	0.562	0.786	0.780	0.573	0.800	0.795	0.601
0.55	0.70	0.784	0.786	0.570	0.786	0.776	0.574	0.788	0.782	0.592
0.55	0.75	0.809	0.801	0.621	0.804	0.798	0.611	0.804	0.792	0.625
0.55	0.80	0.812	0.805	0.625	0.807	0.795	0.621	0.811	0.797	0.622
0.55	0.85	0.835	0.830	0.673	0.821	0.808	0.647	0.830	0.820	0.660
0.55	0.90	0.826	0.819	0.654	0.817	0.804	0.638	0.826	0.820	0.666
0.60	0.65	0.772	0.760	0.549	0.750	0.744	0.511	0.772	0.752	0.544
0.60	0.70	0.774	0.770	0.548	0.769	0.760	0.548	0.781	0.767	0.564
0.60	0.75	0.785	0.779	0.572	0.783	0.768	0.569	0.785	0.776	0.571
0.60	0.80	0.794	0.785	0.590	0.781	0.768	0.568	0.783	0.774	0.569
0.60	0.85	0.816	0.803	0.637	0.806	0.792	0.614	0.809	0.800	0.619
0.60	0.90	0.815	0.804	0.634	0.793	0.787	0.588	0.820	0.812	0.640
0.65	0.70	0.751	0.738	0.505	0.741	0.723	0.488	0.753	0.739	0.508
0.65	0.75	0.762	0.757	0.525	0.745	0.734	0.495	0.756	0.746	0.514
0.65	0.80	0.768	0.770	0.537	0.764	0.753	0.532	0.780	0.774	0.567
0.65	0.85	0.782	0.775	0.571	0.762	0.758	0.524	0.765	0.763	0.529
0.65	0.90	0.798	0.790	0.598	0.761	0.751	0.527	0.775	0.762	0.554
0.70	0.75	0.741	0.731	0.484	0.727	0.719	0.453	0.740	0.728	0.481
0.70	0.80	0.741	0.727	0.486	0.737	0.720	0.477	0.746	0.738	0.492
0.70	0.85	0.767	0.764	0.535	0.744	0.740	0.493	0.761	0.753	0.530
0.70	0.90	0.797	0.785	0.599	0.766	0.757	0.533	0.759	0.762	0.519
0.75	0.80	0.708	0.711	0.418	0.700	0.681	0.402	0.713	0.699	0.427
0.75	0.85	0.733	0.724	0.467	0.712	0.701	0.426	0.727	0.727	0.460
0.75	0.90	0.753	0.755	0.508	0.724	0.713	0.453	0.747	0.730	0.497
0.80	0.85	0.706	0.699	0.413	0.683	0.668	0.367	0.702	0.693	0.413
0.80	0.90	0.739	0.731	0.482	0.714	0.696	0.432	0.717	0.706	0.435
0.85	0.90	0.689	0.687	0.380	0.685	0.667	0.370	0.696	0.685	0.391

Table D.5: Median nPC-LDA performance estimated via FAST-CV for binary classification based on gene expression data, methylation data, and the concatenation of both datasets. AUC values between the lower threshold (m_1) and the upper threshold (m_2) are discarded. At each set of thresholds, we performed classification for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds. The median is taken across these drugs.

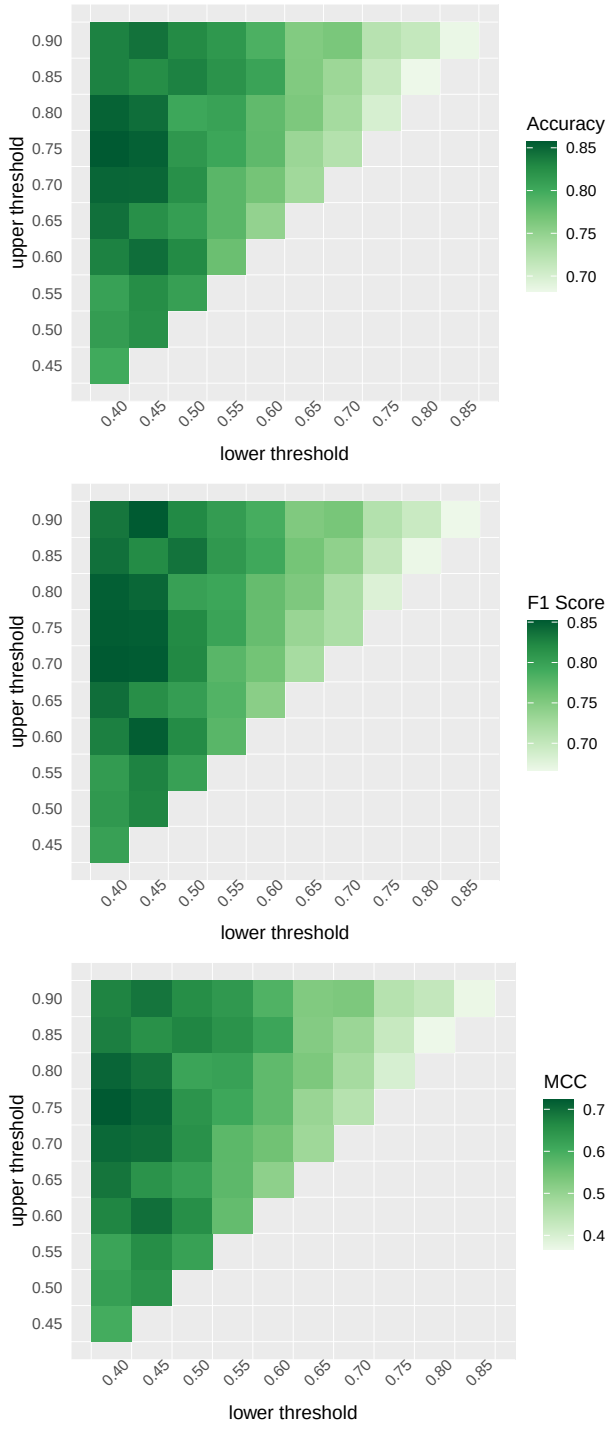

Figure D.8: Median nPC-LDA performance estimated via FAST-CV for binary classification based on methylation data. AUC values between the lower threshold (m_1) and the upper threshold (m_2) are discarded. All three measures of model performance tend to be better for smaller values of m_1. At each set of thresholds, we performed classification for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds. The median is taken across these drugs.

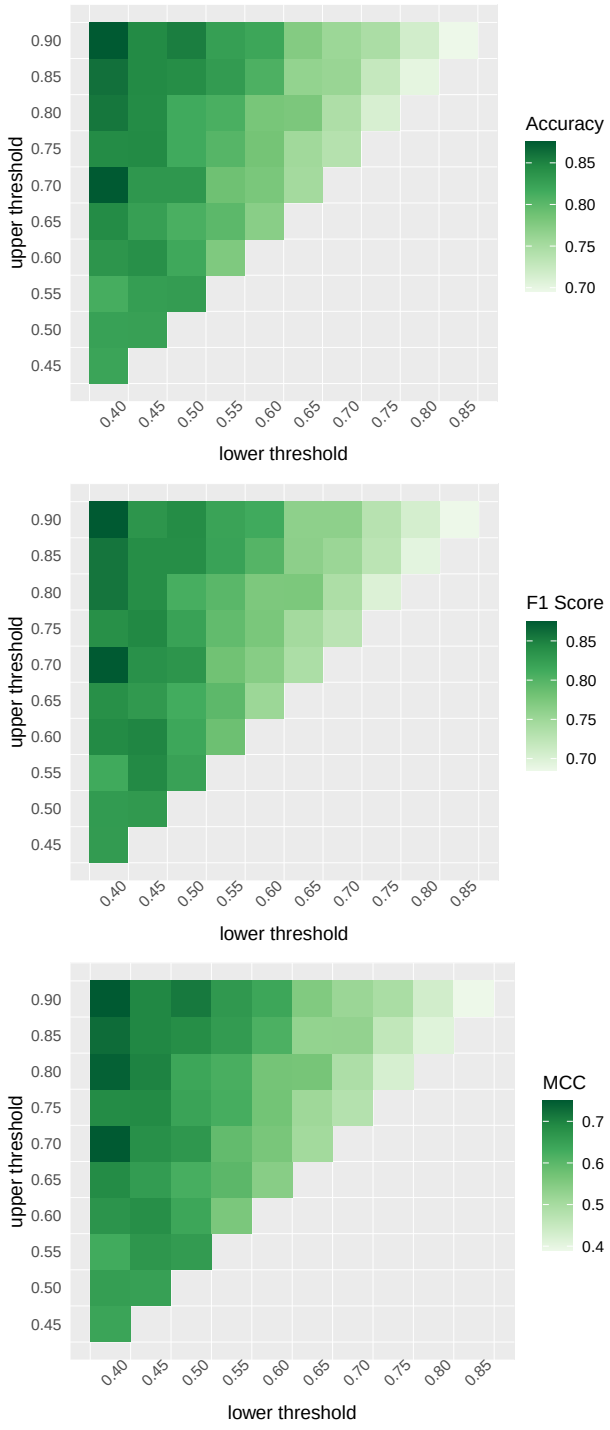

Figure D.9: Median nPC-LDA performance estimated via FAST-CV for binary classification based on both methylation and gene expression data. AUC values between the lower threshold (m_1) and the upper threshold (m_2) are discarded. All measures of model performance tend to be better for smaller values of m_1. At each set of thresholds, we performed classification for the $n = 32$ drugs that have sufficient cell lines in both classes at all tested thresholds. The median is taken across these drugs.

BIBLIOGRAPHY

Barretina, J., et al. (2012), The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity, *Nature*, *483*, 603–607.

Barretina, J., et al. (2019), Addendum: The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity, *Nature*, *565*, E5–E6.

Ben-David, U., et al. (2018), Genetic and transcriptional evolution alters cancer cell line drug response, *Nature*, *560*, 325–330.

Bickel, P. J., and E. Levina (2004), Some theory for Fisher's linear discriminant function, 'naive Bayes', and some alternatives when there are many more variables than observations, *Bernoulli*, *10*(6), 989 – 1010.

Bouhaddou, M., et al. (2016), Drug response consistency in ccle and cgp, *Nature*, *540*, E9–E10.

Brideau, C., B. Gunter, B. Pikounis, and A. Liaw (2003), Improved statistical methods for hit selection in high-throughput screening, *Journal of Biomolecular Screening*, *8*, 634–647.

Cai, T., and W. Liu (2011), A direct estimation approach to sparse linear discriminant analysis, *Journal of the American Statistical Association*, *106*(496), 1566–1577.

Caraus, I., A. A. Alsuwailem, R. Nadon, and V. Makarenkov (2015), Detecting and overcoming systematic bias in high-throughput screening technologies: a comprehensive review of practical issues and methodological solutions, *Briefings in Bioinformatics*, *16*(6), 974–986, doi:10.1093/bib/bbv004.

Cawley, G. C., and N. L. C. Talbot (2003), Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers, *Pattern Recognition*, *36*, 2585–2592.

Cleveland, W. S., E. Grosse, and W. M. Shyu (1992), Local regression models, in *Statistical Models in S*, chap. 8, Wadsworth & Brooks/Cole.

Ding, K.-F., et al. (2017), Analysis of variability in high throughput screening data: applications to melanoma cell lines and drug responses, *Oncotarget*, *8*(17).

Geeleher, P., E. R. Gamazon, C. Seoighe, N. J. Cox, and R. S. Huang (2016), Consistency in large pharmacogenomic studies, *Nature*, *540*, E1–E2.

Genomics of Drug Sensitivity in Cancer (), Help and documentation: Publications, accessed: 2019-02-07.

Guo, Y., T. Hastie, and R. Tibshirani (2006), Regularized linear discriminant analysis and its application in microarrays, *Biostatistics*, *8*(1), 86–100.

Hafner, M., M. Niepel, M. Chung, and P. K. Sorger (2016), Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs, *Nature Methods*, *13*, 521–527.

Haibe-Kains, B., N. El-Hachem, N. J. Birkbak, A. C. Jin, A. H. Beck, H. J. W. L. Aerts, and J. Quackenbush (2013), Inconsistency in large pharmacogenomic studies, *Nature*, *504*, 389–393.

Hastie, T., and R. Tibshirani (2004), Efficient quadratic regularization for expression arrays, *Biostatistics*, *5*(3), 329–340.

Hastie, T., A. Montanari, S. Rosset, and R. J. Tibshirani (2019), Surprises in high-dimensional ridgeless least squares interpolation, *arXiv preprint arXiv:1903.08560*.

Hatzis, C., et al. (2014), Enhancing reproducibility in cancer drug screening: How do we move forward?, *Cancer Research*, *74*(15), 4016–4023.

Haverty, P. M., et al. (2016), Reproducible pharmacogenomic profiling of cancer cell line panels, *Nature*, *533*, 333–337.

Hu, Z. T., Y. Ye, P. A. Newbury, H. Huang, and B. Chen (), *AICM: A Genuine Framework for Correcting Inconsistency Between Large Pharmacogenomics Datasets*, pp. 248–259, doi:10.1142/9789813279827_0023.

Krzanowski, W. J., P. Jonathan, W. V. McCarthy, and M. R. Thomas (1995), Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *44*(1), 101–115.

Larsson, P., H. Engqvist, J. Biermann, E. Werner Rönnerman, E. Forssell-Aronsson, A. Kovács, P. Karlsson, K. Helou, and T. Z. Parris (2020), Optimization of cell viability assays to improve replicability and reproducibility of cancer drug sensitivity screens, *Scientific Reports*, *10*.

Makarenkov, V., P. Zentilli, D. Kevorkov, A. Gagarin, N. Malo, and R. Nadon (2007), An efficient method for the detection and elimination of systematic error in high-throughput screening, *Bioinformatics*, *23*(13), 1648–1657, doi:10.1093/bioinformatics/btm145.

Meijer, R. J., and J. J. Goeman (2013), Efficient approximate k-fold and leave-one-out cross-validation for ridge regression, *Biometrical Journal*, *55*(2), 141–155.

Mpindi, J.-P., P. Swapnil, B. Dmitrii, S. Jani, K. Saeed, K. Wennerberg, T. Aittokallio, P. Östling, and O. Kallioniemi (2015), Impact of normalization methods on high-throughput screening data with high hit rates and drug testing with dose–response data, *Bioinformatics*, *31*(23), 3815–3821, doi:10.1093/bioinformatics/btv455.

Mpindi, J. P., et al. (2016), Consistency in drug response profiling, *Nature*, *540*, E5–E6.

Murie, C., C. Barette, L. Lafanechere, and R. Nadon (2014), Control-plate regression (cpr) normalization for high-throughput screens with many active features, *Journal of Biomolecular Screening*, *19*, 661–671.

Niepel, M., et al. (2019), A multi-center study on the reproducibility of drug-response assays in mammalian cell lines, *Cell Systems*, *9*(1), 35 – 48, doi:https://doi.org/10.1016/j.cels.2019.06.005.

Payne, N. Y., and J. A. Gagnon-Bartsch (2022), Separating and reintegrating latent variables to improve classification of genomic data, *Biostatistics*.

Peck, R., and J. Van Ness (1982), The use of shrinkage estimators in linear discriminant analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *4*(5), 530–537.

Pozdeyev, N., M. You, R. Mackie, R. E. Schweppe, A. C. Tan, and B. R. Haugen (2016), Integrating heterogenous drug sensitivity data from cancer pharmacogenomic studies, *Oncotarget*, *7*(32), 51,619–51,625.

Rahman, R., S. R. Dhruba, K. Matlock, C. De-Niz, S. Ghosh, and R. Pal (2018), Evaluating the consistency of large-scale pharmacogenomic studies, *Briefings in Bioinformatics*, pp. 1–20.

Ramey, J. A., C. K. Stein, P. D. Young, and D. M. Young (2017), High-dimensional regularized discriminant analysis, *arXiv:1602.01182v2*.

Rehnberg, Z. L., A. Rao, and J. A. Gagnon-Bartsch (In Preparation), Technical variation in drug screening studies.

Rubin, D. B. (1974), Estimating causal effects of treatments in randomized and nonrandomized studies, *Journal of Educational Psychology*, *66*, 688–701.

Safikhani, Z., et al. (2016a), Assessment of pharmacogenomic agreement, *F1000Research*, *5*(825).

Safikhani, Z., et al. (2016b), Revisiting inconsistency in large pharmacogenomic studies, *F1000Research*, *5*(2333).

Splawa-Neyman, J., D. M. Dabrowska, and T. P. Speed (1990), On the application of probability theory to agricultural experiments. essay on principles. section 9., *Statistical Science*, *5*, 465–480.

The Cancer Cell Line Encyclopedia and Genomics of Drug Sensitivity in Cancer Investigators (2015), Pharmacogenomic agreement between two cancer cell line data sets, *Nature*, *528*, 84–87.

Thomaz, C. E., E. C. Kitani, and D. F. Gillies (2006), A maximum uncertainty lda-based approach for limited sample size problems — with application to face recognition, *Journal of the Brazilian Computer Society*, *12*, 7–18.

Treder, M. S. (2018), Cross-validation in high-dimensional spaces: a lifeline for least-squares models and multi-class lda, *arXiv:1803.10016*.

van de Wiel, M. A., M. M. van Nee, and A. Rauschenberger (2021), Fast cross-validation for multi-penalty high-dimensional ridge regression, *Journal of Computational and Graphical Statistics*, *30*(4), 835–847.

Vis, D. J., L. Bombardelli, H. Lightfoot, F. Iorio, M. J. Garnett, and L. F. A. Wessels (2016), Multilevel models improve precision and speed of ic50 estimates, *Pharmacogenomics*, *17*(7), 691–700.

Wang, D., J. Hensman, G. Kutkaite, T. S. Toh, J. R. Dry, J. Saez-Rodriguez, M. J. Garnett, M. P. Menden, and F. Dondelinger (2020), A statistical framework for assessing pharmacological response and biomarkers with confidence, *Preprint at https://www.biorxiv.org/content/10.1101/2020.05.01.072983v1*.

Weinstein, J. N., and P. L. Lorenzi (2013), Discrepancies in drug sensitivity, *Nature*, *504*, 381–383.

Xu, P., G. N. Brock, and R. S. Parrish (2009), Modified linear discriminant analysis approaches for classification of high-dimensional microarray data, *Computational Statistics and Data Analysis*, *53*(5), 1674–1687.

Yang, W., et al. (2013), Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells, *Nucleic Acids Research*, *41*, D955–D961.

Ye, J., and T. Wang (2006), Regularized discriminant analysis for high dimensional, low sample size data, in *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.