Carnegie Mellon University
Dietrich College of Humanities and Social Sciences
Dissertation

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

Title: Robust Inference for Single-Cell CRISPR Screens
Presented by: Timothy Barry
Accepted by: Department of Statistics & Data Science
Readers:

KATHRYN ROEDER, ADVISOR

EUGENE KATSEVICH (WHARTON SCHOOL, UNIVERSITY OF PENNSYLVANIA), ADVISOR

JING LEI

F WILLIAM TOWNES

JIAN MA (SCHOOL OF COMPUTER SCIENCE, CARNEGIE MELLON UNIVERSITY)

Approved by the Committee on Graduate Degrees:

RICHARD SCHEINES, DEAN DATE

Carnegie Mellon University

Robust Inference for Single-Cell CRISPR Screens

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Statistics

by

Timothy Barry

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

Carnegie Mellon University

June 8, 2023

Abstract

CRISPR is a genome engineering technology that has enabled scientists to precisely manipulate and perturb human genomes. Single-cell CRISPR screens combine CRISPR genome engineering and single-cell sequencing to survey the effects of CRISPR perturbations on the molecular phenotypes of individual cells. Single-cell CRISPR screens have generated substantial academic and industrial interest in recent years, promising to accelerate cancer, longevity, and medical genetics research. However, single-cell CRISPR screens pose considerable statistical challenges, currently limiting the reliability of conclusions drawn on the basis of single-cell CRISPR screen experiments. The broad objective of this thesis is to develop statistically rigorous and computationally efficient methods for the analysis of single-cell CRISPR screen data.

To this end we make three main contributions. First, we leverage cluster- and cloud-scale computing to conduct an extensive empirical investigation of a diverse array of single-cell CRISPR screen datasets, identifying the most pressing statistical challenges that the data pose. Next, we develop statistical methods that address these challenges both in theory and practice. An application of the proposed methods to real data indicates considerably improved performance relative to existing methods in most cases. Finally, we implement the proposed methods in efficient and practical software aimed at working biologists. Taken together, these contributions help put single-cell CRISPR screen data analysis onto more solid statistical footing, thereby facilitating the application of single-cell CRISPR screen technology to accelerate biological discovery. The methods that we develop — which primarily focus on assumption- and compute-lean hypothesis testing — may be of independent statistical interest.

Acknowledgments

TO APPEAR. (I thank everyone in the department. I especially thank Kathryn and Gene.)

Contents

One

CRISPR, single-cell sequencing, and single-cell CRISPR screens

This thesis is about two transformative genomic technologies that emerged over the past decade: CRISPR and single-cell sequencing. CRISPR is a genome engineering tool that has enabled scientists to edit and perturb human genomes with unprecedented precision and efficiency. Scientists are leveraging CRISPR to design medicines that "fix" aberrant genes in individuals suffering from genetic disorders; CRISPR-based therapeutics for sickle-cell anemia (Frangoul et al., 2021), cancer (Yin et al., 2019), and muscular dystrophy (Min et al., 2019) are in development. Scientists also routinely apply CRISPR in non-clinical research settings to accelerate biological discovery. CRISPR screens (Bock et al., 2022) are experiments in which CRISPR perturbations are introduced into a pool of cells, with each CRISPR perturbation targeting a region of the genome — typically of unknown function — for deactivation, inhibition, or activation. The effects of the induced CRISPR perturbations are assessed by monitoring the perturbed cells for phenotypic changes. For example, in the extreme case that all cells infected by a given perturbation die, one can infer that the region of the genome targeted by the given perturbation is crucial for cellular function. CRISPR screens have become a key component of drug discovery pipelines.

Single-cell sequencing is an experimental technique in which the molecular profile of a large number of individual cells is measured (Hao et al., 2021). First, a pool of cells is cultured; then, each cell is captured in a droplet containing chemical reagents, and the set of gene transcripts contained within each cell is sequenced. This process yields an $n \times p$ matrix, where n is the number of cells sequenced and p is the number of genes measured. (Typically, n is on the order of several thousand to several million, while p is on the order of $20,000$, the number of protein-coding genes in the human genome.) Each entry in the matrix is a count taking a value in the set of non-negative integers $\{0, 1, 2, \ldots, \}$.

A given entry of the matrix indicates the expression level of a given gene within a given cell. Single-cell sequencing provides a more information-rich readout of the molecular content of a collection of cells than older methods, such as bulk RNA sequencing. Moreover, multimodal single-cell sequencing techniques enable the measurement of cellular modalities beyond gene expression, such as cell-surface protein expression and chromatin accessibility (Zhu et al., 2020).

Single-cell CRISPR screens combine CRISPR genome engineering and single-cell sequencing to survey the effects of CRISPR perturbations on individual cells (Dixit et al., 2016; Mimitou et al., 2019; Papalexi et al., 2021). In such screens, a library of CRISPR perturbations is introduced into a population of cells, following by single-cell sequencing to determine the perturbation (or perturbations) that each cell received and measure a rich molecular phenotype for each cell (typically including gene expressions). Single-cell CRISPR screens have emerged as among the most flexible and powerful methods for linking genetic perturbations to changes in cellular phenotypes, promising to accelerate cancer, longevity, and medical genetics research (Replogle et al., 2022; Bock et al., 2022). However, single-cell CRISPR screens pose considerable statistical challenges, currently limiting the reliability of conclusions drawn on the basis of single-cell CRISPR screen experiments. The primary objective of this thesis is to develop statistically rigorous and computationally efficient methods for the analysis of single-cell CRISPR screen data. The secondary objective of this thesis is to implement these methods in practical software tools aimed at working biologists. The methods that we develop — which focus on robust hypothesis testing and estimation (with confidence) in the presence of measurement error — may be of independent statistical interest.

The mechanics of single-cell CRISPR screens

A single-cell CRISPR experiment involves several steps, which we describe here. First, the scientist constructs a library of CRISPR perturbations, each designed to target a different region of the genome for disruption. The CRISPR perturbations typically target genes (Replogle et al., 2022) or enhancers (Xie et al., 2017, 2019a; Gasperini et al., 2019) with unknown or poorly understood function. (Enhancers are noncoding regulatory elements in the genome that are responsible for increasing or decreasing the expression of one or more nearby genes.) The CRISPR perturbations can knockout (Dixit et al., 2016; Papalexi et al., 2021), inhibit (Schraivogel et al., 2020; Gasperini et al., 2019), or activate (Alda-Catalinas et al., 2020; Chardon et al., 2023) their targets; variants of CRISPR called CRISPR-knockout (CRISPRko), CRISPR-inhibition (CRISPRi), and CRISPR-activation (CRISPRa) implement these different modalities of genetic perturbation. Special perturbations called "non-targeting"

(NT) perturbations are designed to exert no effect whatsoever on the cell and serve as negative controls. The CRISPR library typically consists of tens to tens of thousands of CRISPR perturbations.

Next, the scientist introduces the library of CRISPR of perturbations into a pool of cells; the perturbations assort into the cells randomly. The scientist typically chooses a cell type or tissue relevant to the disease or trait under investigation. (For example, if the scientist is interrogating the immune system, the scientist likely would select immune cells, such as T cells, for perturbation; Frangieh et al. 2021.) The perturbations can be administered either at "high multiplicity-of-infection" (high-MOI; Gasperini et al. 2019; Morris et al. 2023; Yao et al. 2023) or "low multiplicity-of-infection" (low-MOI; Dixit et al. 2016; Replogle et al. 2022). In high-MOI screens many (e.g., 10-30) CRISPR perturbations are inserted into each cell; in low-MOI screens, by contrast, exactly one CRISPR perturbation is inserted into each cell. (Cells that by chance receive two or more perturbations are discarded in low-MOI screens.) The high-MOI design is more common for enhancer-targeting screens (as each perturbation is expected to have a weak effect in that setting), while the low-MOI design is more common for gene-targeting screens.

The scientist then waits several days for the perturbations to take effect and uses single-cell sequencing to measure each cell's transcriptome (i.e., its gene expressions) and the perturbation (or perturbations) that it received. (Each cell's protein expressions and/or chromatin accessibility profile may also be measured; Frangieh et al. 2021; Papalexi et al. 2021; Liscovitch-Brauer et al. 2021.) Finally, the scientist subjects the data to a statistical analysis. The objective of the statistical analysis is to understand how the various perturbations impact the expressions of the genes. There are two broad approaches to analyzing single-cell CRISPR screen data: multivariate and univariate. The multivariate approach (Zhou et al., 2022; Yao et al., 2023) involves fitting a model to the entire set of perturbations and gene expressions and leveraging this fitted model to study the impact of perturbations (or sets of perturbations) on genes (or sets of genes). The univariate approach (Gasperini et al., 2019; Barry et al., 2021a), by contrast, involves testing for association (and estimating the change in expression) for a large set of preselected perturbation-gene pairs. (Each pair is analyzed one at a time). The multivariate and univariate approaches each present statistical and computational advantages and disadvantages. This thesis focuses on the univariate approach, which is simpler (though still quite challenging), more commonplace, and more amenable to computational parallelization.

The univariate approach to analyzing single-cell CRISPR screen data

We describe the univariate approach to single-cell CRISPR screen analysis in greater detail. The analysis procedure differs somewhat between low-MOI and high-MOI settings. To test for association between a given targeting CRISPR perturbation and gene in low-MOI (Figure 1.1a), one first divides the cells into two groups: those that received the targeting perturbation, and those that received a non-targeting (NT) perturbation. (All other cells typically are ignored.) One then tests for differential expression of the given gene across these two groups of cells, yielding a fold change estimate and p-value. (The differential expression test entails testing if the distribution of the gene expression differs across the two groups of cells.) One repeats this procedure for a (typically) large, preselected set of perturbation-gene pairs. Finally, one computes the discovery set by subjecting the tested pairs to a multiplicity correction procedure, typically the BH procedure or a variant thereof (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001). The procedure for analyzing high-MOI single-cell CRISPR screen data is similar. To test for association between a given targeting perturbation and gene, one partitions the cells into two groups: those that received the given targeting perturbation, and those that did *not* receive the given targeting perturbation. One then tests for differential expression of the given gene across these two groups of cells (Figure 1.1b). (One cannot compare against cells containing only NT perturbations, as very few (if any) cells contain only NT perturbations in high-MOI.)

A first-pass statistical framing of the problem

We provide a first-pass, high-level statistical rendering of the problem to make the problem more concrete for a statistical audience. Consider a given targeting-perturbation-gene pair. We observe i.i.d. data $(X_1, Y_1, Z_1), \ldots, (X_n, Y_n, Z_n)$, where $X_i \in \{0, 1\}$ is the "perturbation indicator" of cell i, $Y_i \in \{0, 1, 2, 3, \ldots\}$ is the expression of the gene in cell i, and $Z_i \in \mathbb{R}^p$ is a low-dimensional vector of "technical factors" measured in cell i. X_i is set to 1 if cell i contains the targeting perturbation and 0 otherwise. (In low-MOI, $X_i = 0$ if cell i contains a non-targeting perturbation, while in high-MOI, $X_i = 1$ if cell i does *not* contain the targeting perturbation.) The "technical factor" vector Z_i is a vector of cell-specific covariates that is measured as part of the single-cell sequencing process. The technical factor vector includes covariates such as sequencing depth (i.e., the number of gene transcripts sequenced in the cell) and sequencing batch (i.e., the batch in which the given cell was sequenced). Typically, the dimension of Z_i

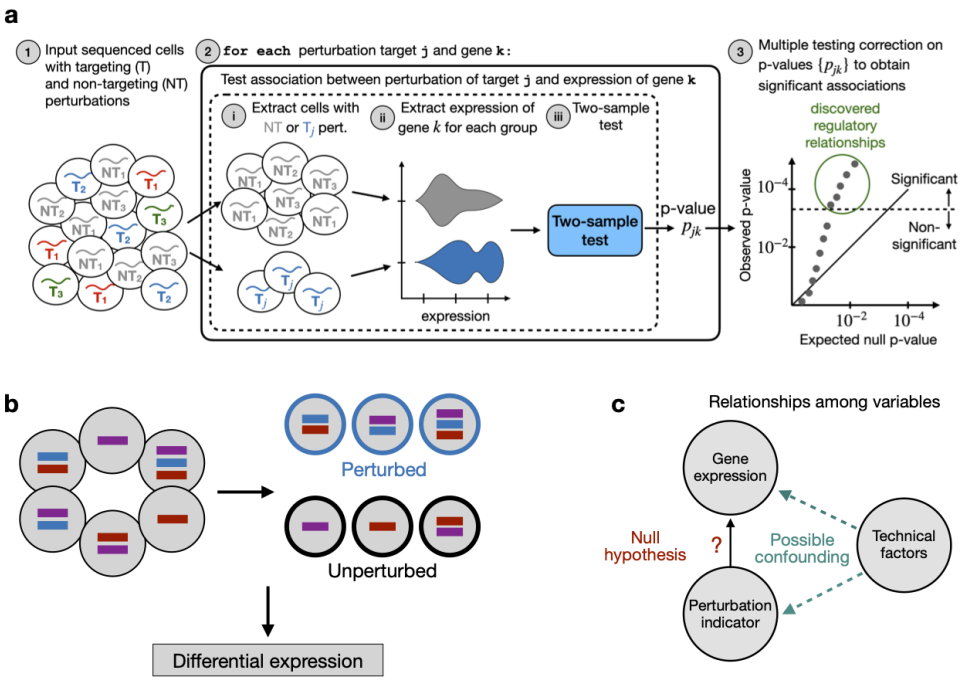


Figure 1.1: **Overview of the statistical analysis protocol for single-cell CRISPR screen data**. **a**, The standard paradigm for association testing on low-MOI single-cell CRISPR screen data. One compares the expression of a given gene across two groups of cells: those containing the given targeting perturbation, and those containing a non-targeting (NT) perturbation. **b**, The standard paradigm for high-MOI data. One compares the expression of the given gene across cells containing the given targeting perturbation and *all other cells*. **c**, The relationship between the variables in the analysis. The technical factors may or may not exert a confounding effect on the perturbation indicator and gene expression.

ranges from about three to seven. We assume that the technical factor vector Z_i *may* or *may not* exert a confounding effect on the perturbation indicator X_i and gene expression count Y_i. (That is, X_i and Y_i may be dependent or independent of Z_i) Whether or not Z_i exerts a confounding effect can vary across datasets and even across pairs within a given dataset.

Rendered in these terms, our primary objective is to test whether the gene expression count is independent of the perturbation indicator given the technical factor vector (Figure 1.1c). In other words, we seek to test the conditional

independence null hypothesis $H_0 : Y_i \perp\!\!\!\perp X_i | Z_i$. (Note that if Z_i does not exert a confounding effect, then conditional independence $Y_i \perp\!\!\!\perp X_i | Z_i$ and marginal independence $Y_i \perp\!\!\!\perp X_i$ hypotheses are equivalent.) We seek to develop a method for this purpose that is *well-calibrated* and *powerful*. That is, the method should produce uniformly distributed p-values under the null hypothesis of no association between the perturbation indicator and the expression of the gene, and the method should yield small p-values when there *is* an association.

Two methods frequently employed to carry out the association test are negative binomial regression (Gasperini et al., 2019) and the Wilcoxon two-sample test (Mimitou et al., 2019; Papalexi et al., 2021; Wessels et al., 2022). Negative binomial regression models the gene expressions as a function of the perturbation indicator and technical factors:

$$Y_i \sim \mathrm{NB}_\theta(\mu_i); \quad \log(\mu_i) = \gamma X_i + \beta^T Z_i, \tag{1.1}$$

where $NB_\theta(\mu_i)$ denotes the negative binomial distribution with size parameter θ and mean μ_i, and $\gamma \in \mathbb{R}$ and $\beta^T \in \mathbb{R}^p$ are unknown constants. The model (1.1) is fitted to the data, and a Wald test is used to test the hypothesis $H_0 : \gamma = 0$. (If the NB model is correctly specified, then a test of the null hypothesis $H_0 : \gamma = 0$ is equivalent to a test of the conditional independence hypothesis $H_0 : X_i \perp\!\!\!\perp Y_i | Z_i$.) The Wilcoxon two-sample test, meanwhile, nonparametrically tests for a distributional difference across the sets $\{Y_i : X_i = 1\}$ and $\{Y_i : X_i = 0\}$. (The vector Z_i is ignored.) While negative binomial regression and the Wilcoxon test are decent starting points, both can be improved upon considerably via more sophisticated techniques, as we aim to show.

Our secondary goal is to estimate the effect of the perturbation indicator X_i on the gene expression count Y_i while controlling for the technical factor vector Z_i. This secondary goal moves beyond statistical significance and toward biological significance. Within the context of the negative binomial model (1.1), the estimation problem amounts to estimating (with confidence) the parameter γ. Chapters 2 and 4 focus on the hypothesis testing problem, while Chapter 3 focuses on the estimation problem.

A key application of single-cell CRISPR screens: illuminating the function of noncoding GWAS variants

Single-cell CRISPR screens are poised to impact a variety of research areas. One especially important application of single-cell CRISPR screens is to shed light on results obtained from genome-wide association studies (GWAS). The purpose of a GWAS is to identify regions in the genome that are associated with a trait or disease of interest. A GWAS proceeds in several steps (Uffelmann et al.,

2021). First, one identifies a disease of interest (e.g., bipolar disorder). Next, one collects a large sample of subjects (typically numbering in the hundreds of thousands), where some of these subjects are cases (i.e., they have the disease) and the rest are controls (i.e., they are healthy). One then genotypes each individual, which consists of sequencing each individual's DNA at a large number of prespecified locations within the genome (called *variants*). This genotyping step yields an $n \times p$ matrix, where n is the number of subjects in the study, and p is the number of variants sequenced. Each entry of the matrix — coded as 0, 1, or 2 — indicates the version of each variant that each individual harbors. Finally, one regresses the disease status vector onto the matrix of variants, typically proceeding one variant at a time. This process yields a p-value for the test of association between each variant and the disease. Variants with very small p-values are considered to be associated with the disease. One typically plots the p-value of each variant against its genomic position, forming a so-called "Manhattan plot." A Manhattan plot for bipolar disorder is shown in Figure 1.2.

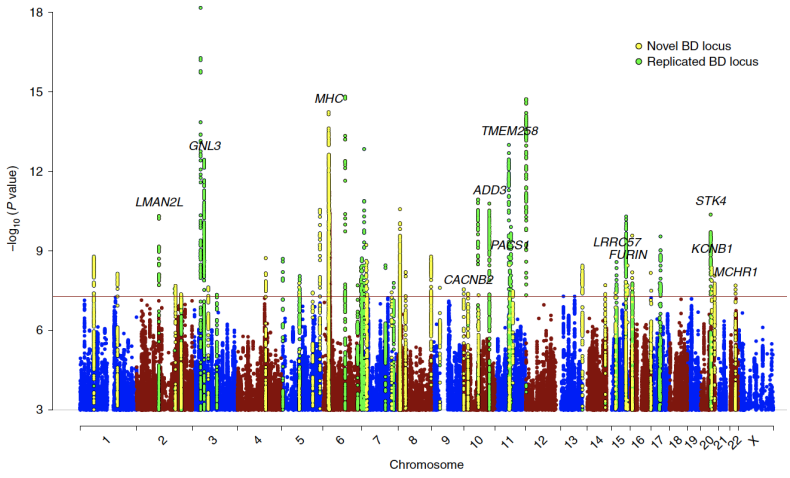


Figure 1.2: **An example GWAS of bipolar disorder**. Each point represents a genomic variant. The x-axis displays the position of the genomic variant in the genome, and the y-axis displays the p-value for a test of association between the genomic variant and the phenotype (i.e., presence of absence of bipolar disorder). The horizontal red line indicates the genome-wide significance level, drawn at $5 \cdot 10^{-8}$. Variants with p-values below the genome-wide significance threshold are considered to be significantly associated with the phenotype. (Figure taken from Mullins et al. 2021).

Two major challenges associated with interpreting GWAS results are as follows (Uffelmann et al., 2021; Tam et al., 2019). First, the vast majority (>98%) of variants associated with diseases are located outside of genes and in noncoding regions (Gallagher and Chen-Plotkin, 2018). These noncoding variants are thought to lie inside of enhancers, contributing to disease by modulating the expression of one or more disease-relevant genes. It is unclear which gene (or genes) most noncoding variants regulate. A major objective in contemporary genetics, therefore, is to map noncoding variants to target genes at genome-wide scale. Second, variants in close physical proximity tend to be extremely highly correlated with one another (due to linkage disequilibrium), making it challenging to identify the causal variant among a set of nearby variants. A second major objective in contemporary genetics, therefore, is to distinguish causal variants from variants that merely are strongly correlated with causal variants.

Single-cell CRISPR screens are among the most promising technologies for resolving both of these challenges (Morris et al., 2023). Several recent, cutting edge studies have leveraged single-cell CRISPR screens to map noncoding GWAS variants (Morris et al., 2023; Tuano et al., 2023). These studies targeted CRISPR perturbations to noncoding GWAS variants in disease-relevant cell types. These GWAS variants were then mapped to target genes by testing for association between the CRISPR perturbations that disrupted these GWAS variants and nearby genes. The former study used CRISPR base editing to disrupt single variants among sets of variants in strong linkage disequilibrium and mapped these individual variants. Altogether, these studies demonstrate the promise of single-cell CRISPR screens to elucidate the function of noncoding GWAS variants.

Organization of the thesis

This thesis contains three content chapters. Chapter 2 studies the association testing problem in the high-MOI setting. We propose SCEPTRE (high-MOI), a method for well-calibrated and powerful association testing based on the conditional randomization test. Next, Chapter 3 examines the estimation problem in the high-MOI setting. We introduce GLM-EIV, a method for estimation (with confidence) in the presence of non-Gaussian measurement error. Finally, Chapter 4 explores the association testing problem in the low-MOI setting. We propose SCEPTRE (low-MOI) a substantial extension of SCEPTRE (high-MOI) geared to the low-MOI setting. SCEPTRE (low-MOI) is based on the novel and statistically principled technique of permuting GLM score statistics. Throughout, we explore and highlight fundamental statistical and computational challenges that single-cell CRISPR screen data pose.

Two

Robust association testing at high multiplicity-of-infection

2.1 Introduction

The noncoding genome plays a crucial role in human development and homeostasis: over 90% of loci implicated by GWAS in diseases lie in regions outside protein-coding exons (Gallagher and Chen-Plotkin, 2018). Enhancers and silencers, segments of DNA that modulate the expression of a gene or genes in *cis*, harbor many or most of these noncoding trait loci. While millions of *cis*-regulatory elements (CREs) have been nominated through biochemical annotations, the functional role of these CREs, including the genes that they target, remain essentially unknown (Gasperini et al., 2020). A central challenge over the coming decade, therefore, is to unravel the *cis*-regulatory landscape of the genome across various cell types and diseases.

Single-cell CRISPR screens — implemented by Perturb-seq (Dixit et al., 2016; Adamson et al., 2016), CROP-seq (Datlinger et al., 2017), ECCITE-seq (Mimitou et al., 2019) and other protocols — are the most promising technology for mapping CREs to their target genes at genome-wide scale. Single-cell CRISPR screens pair CRISPR perturbations with single-cell sequencing to survey the effects of perturbations on cellular phenotypes, including the transcriptome. High multiplicity of infection (MOI) screens deliver dozens perturbations to each cell (Xie et al., 2017, 2019a; Gasperini et al., 2019), enabling the interrogation of hundreds or thousands of CREs in a single experiment. Single-cell screens overcome the limitations of previous technologies for mapping CREs (Gasperini et al., 2019): unlike eQTLs, single-cell screens are high-resolution and can target rare variants, and unlike bulk screens, single-cell

This Chapter is based on Barry et al. (2021a), which is joint work with Xuran Wang, John Morris, Kathyrn Roeder, and Eugene Katsevich. Timothy Barry and Eugene Katsevich both contributed to the text and figures.

screens measure the impact of perturbations on the entire transcriptome.

Despite their promise, high-MOI single cell CRISPR screens pose significant statistical challenges. In particular, researchers have encountered substantial difficulties in calibrating tests of association between a CRISPR perturbation and the expression of a gene. Gasperini et al. (2019) found considerable inflation in their negative binomial regression based p-values for negative control perturbations. Similarly, Xie et al. (2019a) found an excess of false positive hits in their rank-based Virtual FACS analysis. Finally, Yang et al. (2020) found that their permutation-based scMAGeCK-RRA method deems almost all gene-enhancer pairs significant in a reanalysis of the Gasperini et al. data. These works propose ad hoc fixes to improve calibration, but we argue that these adjustments are insufficient to address the issue. Miscalibrated p-values can adversely impact the reliability of data analysis conclusions by creating excesses of false positive and false negative discoveries.

In this work we make two contributions. We (i) elucidate core statistical challenges at play in high-MOI single-cell CRISPR screens and (ii) present a novel analysis methodology to address them. We identify a key challenge that sets single-cell CRISPR screens apart from traditional differential expression experiments: the "treatment"—in this case the presence of a CRISPR perturbation in a given cell—is subject to measurement error (Dixit et al., 2016; Hill et al., 2018; Replogle et al., 2020). In fact, underlying this measurement error are the same technical factors contributing to errors in the measurement of gene expression, including sequencing depth and batch effects. These technical factors therefore act as confounders, invalidating traditional nonparametric calibration approaches. On the other hand, parametric modeling of single-cell expression data is also fraught with unresolved difficulties.

To address these challenges, we propose SCEPTRE (**s**ingle-**ce**ll **p**er**t**urbation screens via **re**sampling; pronounced "scepter"). SCEPTRE is based on the conditional randomization test (Candès et al., 2018b), a powerful and intuitive statistical methodology that, like parametric methods, enables simple confounder adjusment, and like nonparametric methods, is robust to expression model misspecification. We used SCEPTRE to analyze two recent, large-scale, high-MOI single-cell CRISPR screen experiments. SCEPTRE demonstrated excellent calibration and sensitivity on the data and revealed hundreds of new regulatory relationships, validated using a variety of orthogonal functional assays. In the Discussion we describe an independent work conducted in parallel to the current study in which we leveraged biobank-scale GWAS data, single-cell CRISPR screens, and SCEPTRE to dissect the *cis* and *trans* effects of noncoding variants associated with blood diseases (Morris et al., 2023). This work highlights what we see as a primary application of SCEPTRE: dissecting

regulatory mechanisms underlying GWAS associations.

2.2 Results

Analysis challenges

We examined two recent single-cell CRISPR screen datasets — one produced by Gasperini et al. (2019) and the other by Xie et al. (2019a) — that exemplify several of the analysis challenges in high-MOI single-cell CRISPR screens. Gasperini et al. and Xie et al. used CRISPRi to perturb putative enhancers at high MOI in K562 cells. They sequenced polyadenylated gRNAs alongside the whole transcriptome and assigned perturbation identities to cells by thresholding the resulting gRNA UMI counts.

Both Gasperini et al. and Xie et al. encountered substantial difficulties in calibrating tests of association between candidate enhancers and genes. Gasperini et al. computed p-values using a DESeq2-inspired negative binomial regression analysis implemented in Monocle2 (Love et al., 2014; Qiu et al., 2017); Xie et al. by contrast, computed p-values using Virtual FACS, a nonparametric method proposed by these authors. Gasperini et al. assessed calibration by pairing each of 50 non-targeting (or negative) control gRNAs with each protein-coding gene. These "null" p-values exhibited inflation, deviating substantially from the expected uniform distribution (Figure 2.1a, red). To assess the calibration of Virtual FACS in a similar manner, we constructed a set of *in silico* negative control pairs of genes and gRNAs on the Xie et al. data (see Methods). The resulting p-values were likewise miscalibrated, with some pairs exhibiting strong conservative bias and others strong liberal bias (Figure 2.1a, gray-green).

A core challenge in the analysis of single-cell CRISPR screens is the presence of confounders, technical factors that impact both gRNA detection probability and gene expression. The total number of gRNAs detected in a cell increases with the total number of mRNA UMIs detected in a cell ($\rho = 0.35, p < 10^{-15}$ in Gasperini et al. data; $\rho = 0.25, p < 10^{-15}$ in Xie et al. data; Figures 2.1b-c). Techical covariates, such as sequencing depth and batch, induce a correlation between gRNA detection probability and gene expression, even in the absence of a regulatory relationship (Figure 2.1d). This confounding effect can lead to severe test miscalibration and is especially problematic for traditional nonparametric approaches, which implicitly (and incorrectly) treat cells symmetrically with respect to confounders.

Parametric regression approaches, like negative binomial regression, are the most straightforward way to adjust for confounders. However, parametric

methods rely heavily on correct model specification, a challenge in single-cell analysis given the heterogeneity and complexity of the count data. We hypothesized that inaccurate estimation of the negative binomial dispersion parameter was (in part) responsible for the p-value inflation observed by Gasperini et al. Monocle2 estimates a raw dispersion for each gene, fits a parametric mean-dispersion relationship across genes, and finally collapses raw dispersion estimates onto this fitted line (Figure 2.1e). We computed the deviation from uniformity of the negative control p-values for each gene using the Kolmogorov-Smirnov (KS) test, represented by the color of each point in Figure 2.1e. Circled genes had significantly miscalibrated p-values based on a Bonferroni correction at level $\alpha = 0.05$. Genes significantly above the curve showed marked signs of p-value inflation, suggesting model misspecification. Analysis challenges are summarized in Table 2.1.

Gasperini et al. and Xie et al. incorporated ad hoc adjustments into their analyses to remedy the observed calibration issues. On closer inspection, however, these efforts were not satisfactory to ensure reliability of the results. Gasperini et al. attempted to calibrate p-values against the distribution negative control p-values instead of the more standard uniform distribution. This adjustment lead to overcorrection for some gene-enhancer pairs (false negatives) and undercorrection for others (false positives) (Figure A.1). Along similar lines Xie et al. compared their Virtual FACS p-values to gene-specific simulated null p-values to produce "significance scores" that were used to determine significance. These significance scores were challenging to interpret and could not be subjected to multiple hypothesis testing correction procedures, as they are not p-values.

Improvements to the negative binomial approach

We attempted to alleviate the miscalibration within the negative binomial regression framework by following the recommendations of Hafemeister and Satija, who recently proposed a strategy for parametric modeling of single-cell RNA-seq data (Hafemeister and Satija, 2019). First, we abandoned the DESeq2-style size factors of Monocle2 and instead corrected for sequencing depth by including it as a covariate in the negative binomial regression model. Second, we adopted a more flexible dispersion estimation procedure: we (i) computed raw dispersion estimates for each gene, (ii) regressed the raw dispersion estimates onto the mean gene expressions via kernel regression, and (iii) projected the raw dispersion estimates onto the fitted nonparametric regression curve.

We reanalyzed the Gasperini et al. and Xie et al. negative control data using the improved negative binomial regression approach. In addition to sequencing depth, we included as covariates in the regression model the total number of

Method class	**Example**	**Robust to expression model misspecification**	**Able to adjust for confounders**
Parametric	Monocle (Qiu et al., 2017)	No	Yes
Nonparametric	Virtual FACS (Xie et al., 2019a)	Yes	No
Conditional resampling	SCEPTRE	Yes	Yes

Table 2.1: Statistical methods employed in single-cell CRISPR screen analysis. Parametric methods are non-robust to misspecified gene expression distributions, and classical nonparametric methods cannot adjust for confounders. Conditional resampling (implemented in this work as SCEPTRE) addresses both challenges.

expressed genes per cell and the technical factors accounted for in the original analysis (total number of gRNAs detected per cell, percentage of transcripts mapped to mitochondrial genes, and sequencing batch). Improved negative binomial regression exhibited better calibration than Monocle regression on both Gasperini et al. and Xie et al. datasets. Still, improved negative binomial regression demonstrated clear p-value inflation. We concluded that parametric count models likely are challenging to calibrate to high-MOI single-cell CRISPR screen data.

SCEPTRE: Analysis of single-cell perturbation screens via conditional resampling

To address the challenges identified above, we propose SCEPTRE, a methodology for single-cell CRISPR screen analysis based on the simple and powerful conditional randomization test (Candès et al., 2018b) (Figure 4.3). To test the association between a given gRNA and gene, we first fit the improved negative binomial statistic described above. This yields a z-value, which typically would be compared to a standard normal null distribution based on the parametric negative binomial model. Instead, we build a null distribution for this statistic via conditional resampling. First, we estimate the probability that the gRNA will be detected in a given cell based on the cell's technical factors, such as sequencing depth and batch. Next, we resample a large number of "null" datasets, holding gene expression and technical factors constant while redrawing gRNA

assignment independently for each cell based on its fitted probability. We compute a negative binomial z-value for each resampled dataset, resulting in an empirical null distribution (gray histogram in Figure 4.3). Finally, we compute a left-, right-, or two-tailed probability of the original z-value under the empirical null distribution, yielding a well-calibrated p-value. This p-value can deviate substantially from that obtained based on the standard normal (Figure 4.3, Figure A.2). While we used a negative binomial regression test statistic for this work, SCEPTRE in principle is compatible with any test statistic that reasonably tracks the expression data, including, for example, statistics based on machine learning algorithms.

We leverage several computational accelerations to enable SCEPTRE to scale to large single-cell CRISPR screen datasets. First, we approximate the null histogram of the resampled test statistics using a skew-t distribution to obtain precise p-values based on a limited number of resamples (500 in the current implementation). Second, we employ statistical shortcuts that reduce the cost of each resample by a factor of about 100 (see Methods). Finally, we implement the method so that it can run in parallel on hundreds or thousands of processors on a computer cluster. (We used this approach in our independent study of noncoding blood trait GWAS loci; Morris et al. 2023.) We estimate that SCEPTRE can analyze 2.5 million gene-gRNA pairs on a dataset of 200,000 cells in a single day using 500 processors.

SCEPTRE demonstrates improved calibration and sensitivity on real and simulated data

First, we investigated the calibration of SCEPTRE in a small, proof-of-concept simulation study (Figure 2.3a). We considered a class of negative binomial regression models with fixed dispersion and two technical covariates (sequencing depth and batch). We simulated expression data for a single gene in 1000 cells using four models selected from this class: the first with dispersion = 1, the second with dispersion = 0.2, the third with dispersion = 5, and the last with dispersion = 1, but with 25% zero-inflation. We also simulated negative control gRNA data using a logistic regression model with the same covariates as the gene expression model. We assessed the calibration of SCEPTRE and negative binomial regression across the four simulated datasets. To explore the impact of model misspecification on SCEPTRE and the negative binomial method (on which SCEPTRE relies), we fixed the dispersion of the negative binomial method to 1. The negative binomial method worked as expected when the model was correctly specified. However, negative binomial regression broke down in all three cases of model misspecification. SCEPTRE demonstrated good calibration in all settings.

Next, to assess the calibration of SCEPTRE on real data, we applied SCEPTRE to test the association between negative control gRNAs and genes in the Gasperini et al. data (Figure 2.3b) and Xie et al. data (Figure 2.3c). We compared SCEPTRE to Monocle regression and the improved negative binomial method. For the Xie et al. data, we also compared to Virtual FACS, the method originally applied to the data. SCEPTRE showed good calibration on both datasets; by contrast, Monocle regression and improved negative binomial regression demonstrated signs of severe p-value inflation, while Virtual FACS exhibited a bimodal p-value distribution peaked at 0 and 1.

SCEPTRE demonstrated modestly better calibration on the Gasperini et al. data than on the Xie et al. data. This likely is because the Gasperini et al. negative control pairs — which consisted of real, non-targeting gRNAs — were higher-quality than the Xie et al. negative control pairs — which were constructed *in-silico* using enhancer-targeting gRNAs (see Methods). We reasoned that the Xie et al. negative controls carried mild regulatory signal, resulting in slight inflation of the SCEPTRE p-values on these data relative to the Gasperini et al. data.

To assess the sensitivity of SCEPTRE, we applied SCEPTRE to test the 381 positive control pairs of genes and TSS-targeting gRNAs assayed by Gasperini et al. (Figure 2.3d). Allowing for the fact that the empirical correction employed by Gasperini et al. limited the accuracy of p-values to about 10^{-6}, the SCEPTRE p-values for the positive controls were highly significant, and in particular, almost always more significant than the original empirical p-values, indicating greater sensitivity. Finally, we assessed the sensitivity of SCEPTRE on the Xie et al. data. Xie et al. conducted an arrayed CRISPR screen with bulk RNA-seq readout of ARL15-enh, a putative enhancer of gene *ARL15*. Both SCEPTRE and the bulk RNA-seq differential expression analysis rejected *ARL15* at an FDR of 0.1 after a Benjamini-Hochberg (BH) correction, increasing our confidence in the calibration and sensitivity of SCEPTRE (Figure 2.3e).

Analysis of candidate *cis*-regulatory pairs

We applied SCEPTRE to test all candidate *cis*-regulatory pairs in the Gasperini et al. ($n = 84,595$) and Xie et al. ($n = 5,209$) data. A given gene and gRNA were considered a "candidate pair" if the gRNA targeted a site within one Mb the gene's TSS. SCEPTRE discovered 563 and 139 gene-enhancer links at an FDR of 0.1 on the Gasperini et al. and Xie et al. data, respectively. We used several orthogonal assays to quantify the enrichment of SCEPTRE's discovery set for regulatory biological signals, and we compared the SCEPTRE results to those of other methods.

SCEPTRE's discovery set on the Gasperini et al. data was highly biologically plausible, and in particular, more enriched for biological signals of regulation than the original discovery set. Gasperini et al. discovered 470 gene-gRNA pairs at a reported FDR of 0.1. The SCEPTRE p-values and original empirical p-values diverged substantially: of the 670 gene-enhancer pairs discovered by either method, SCEPTRE and the original method agreed on only 363, or 54% (Figure 2.4a). Gene-enhancer pairs discovered by SCEPTRE were physically closer (mean distance = 65 kb) to each other than those discovered by the original method (mean distance = 81 kb; Figure 2.4b). Furthermore, SCEPTRE's gene-enhancer pairs fell within the same topologically associating domain (TAD) at a higher frequency (74%) than the original pairs (71%). Pairs within the same TAD showed similar levels of HI-C interaction frequency across methods, despite the fact that SCEPTRE discovered 85 more same-TAD pairs (Figure 2.4c). Finally, enhancers discovered by SCEPTRE showed improved enrichment across all eight cell-type relevant ChIP-seq targets reported by Gasperini et al. (Figure 2.4d, Figure A.5a).

When we compared discoveries unique to SCEPTRE ($n = 200$) against those unique to the original method ($n = 107$), the disparities became more extreme (Figure A.3). For example, only 57% of pairs unique to the original method fell within the same TAD, compared to 73% unique to SCEPTRE. We concluded that many pairs in the Gasperini et al. discovery set likely were false positives. Finally, when we compared SCEPTRE to the improved negative binomial method ($n = 824$ discoveries), we observed even greater differences in discovery set quality in favor of SCEPTRE (Figures 2.4b - 2.4d).

We highlight several especially interesting gene-enhancer pairs discovered by SCEPTRE. Five discoveries (Figure 2.4a labels 1-5; Figure 2.4e) were nominated as probable gene-enhancer links by eQTL (Ardlie et al., 2015) and eRNA (Andersson et al., 2014) p-values in relevant tissue types. (eQTLs are genomic loci associated with the differential expression of some typically nearby gene. eRNAs, by contrast, are non-coding RNA molecules transcribed from the primary DNA sequence of an *enhancer*. Both provide evidence of enhancer activity.) The SCEPTRE p-values for these pairs were 1-2 orders of magnitude smaller than the original empirical p-values, hinting at SCEPTRE's greater sensitivity. Additionally, six pairs (Figure 2.4a, blue triangles) were discovered by SCEPTRE but discarded as outliers by the original analysis, underscoring SCEPTRE's ability to handle genes with arbitrary expression distributions.

We repeated the same orthogonal analyses for the SCEPTRE discoveries on the Xie et al. data, comparing SCEPTRE's results to those of Xie et al. Xie et al.'s analysis method, Virtual FACS, outputted "significance scores" rather than p-values (see section "Analysis challenges"). Because significance scores cannot

be subjected to multiple hypothesis testing correction procedures (like BH), we compared the top 139 Virtual FACS pairs (ranked by significance score) against the set of 139 (FDR = 0.1) SCEPTRE discoveries (Figure 2.5a; see Methods). Of the 195 pairs in either set, SCEPTRE and Virtual FACS agreed on only 83, or 43%. The SCEPTRE discoveries were more biologically plausible: compared to the Virtual FACS pairs, the SCEPTRE pairs were (i) physically closer (Figure 2.5b), (ii) more likely to fall within the same TAD (Figure 2.5c), (iii) more likely to interact when in the same TAD (Figure 2.5c), and (iv) more enriched for all eight cell-type relevant ChIP-seq targets (Figure 2.5d, Figure A.5b). When we examined the symmetric difference of the discovery sets, these differences became more pronounced (Figure A.4).

We additionally compared SCEPTRE to Monocle regression ($n = 180$ discoveries) and improved negative binomial regression ($n = 156$ discoveries) on the Xie et al. data. SCEPTRE uniformly dominated Monocle: SCEPTRE pairs were physically closer to one another (median distance = 44kb versus 110kb; Figure 2.5b); SCEPTRE pairs interacted more frequently and were more likely to fall within the same TAD (68% versus 61%; Figure 2.5c); and SCEPTRE pairs were more enriched for seven of eight cell type-relevant ChIP-seq targets (one target, DP22, was a tie; Figure 2.5d). Improved negative binomial regression was more competitive than Monocle across metrics (Figure 2.5b-d). However, as noted earlier, improved negative binomial regression exhibited severe miscalibration on the negative control pairs (Figure 2.3d), rendering its discovery set less reliable than that of SCEPTRE.

Gene expression level and sensitivity

To better understand when SCEPTRE works best, we investigated the impact of gene expression level on the sensitivity of SCEPTRE. We binned candidate gene-enhancer pairs into non-overlapping categories based on mean expression level of the gene. On both the Gasperini et al. and Xie et al. data, we found that candidate pairs containing a highly-expressed gene were more likely to be rejected than candidate pairs containing a lowly-expressed gene (Table A.1,A.2), indicating SCEPTRE's greater sensitivity for highly expressed genes. We observed similar trends for the other methods (not shown), consistent with the intuition that highly-expressed genes carry more information.

2.3 Discussion

In this work we illustrated a variety of statistical challenges arising in the analysis of high-MOI single-cell CRISPR screens, leaving existing methods (based on parametric expression models, permutations, or negative control

data) vulnerable to miscalibration. To address these challenges, we developed SCEPTRE, a resampling method based on modeling the probability a given gRNA will be detected in a given cell, based on that cell's technical factors. We found that SCEPTRE exhibited excellent calibration despite the presence of confounding technical factors and misspecification of single-cell gene expression models. We implemented computational accelerations to bring the cost of the resampling-based methodology down to well within an order of magnitude of the traditional negative binomial parametric approach, making it quite feasible to apply for large-scale data. We used SCEPTRE to reanalyze the Gasperini et al. and Xie et al. data. While our analysis replicated many of their findings, it also clarified other relationships, removing a large set ($> 20\%$ for Gasperini) of pairs that exhibited a weak relationship and adding an even larger set ($> 40\%$ for Gasperini) of new, biologically plausible gene-enhancer relationships. These links were supported by orthogonal evidence from eQTL, enhancer RNA co-expression, ChIP-seq, and HI-C data.

As an example application of SCEPTRE, we highlight *STING-seq*, a platform that we developed in parallel to the current work in an independent study (Morris et al., 2023). STING-seq leverages biobank-scale GWAS data and single-cell CRISPR screens to map noncoding, disease-associated variants at scale. First, we used statistical fine-mapping to identify a set of 88 putatively causal variants from 56 loci associated with quantitative blood traits. We perturbed the selected variants at high MOI in K562 cells using an improved CRISPRi platform and sequenced gRNAs and transcriptomes in individual cells using ECCITE-seq (Mimitou et al., 2019), a protocol that enables the profiling of multiple modalities and the direct capture of gRNAs.

We then applied SCEPTRE to quantify associations between perturbations and changes in gene expression in *cis* (within 500 kb) and *trans*. SCEPTRE confidently mapped 37 noncoding variants to their *cis* target genes, in some cases identifying a causal variant among a set of candidate variants in strong LD. Nine variants were found to regulate a gene other than the closest gene, and four variants were found to regulate multiple genes, an apparent example of pleiotropy. Several perturbations lead to widespread changes in gene expression, illuminating *trans*-effects networks. For example, two variants that were found to regulate the transcription factor *GFI1B* in *cis* altered the expression of hundreds of genes in *trans* upon perturbation; these differentially expressed genes were strongly enriched for GFI1B binding sites and blood disease GWAS hits. We concluded on the basis of this study SCEPTRE can power the systematic dissection regulatory networks underlying GWAS associations.

Despite these exciting results, key challenges remain in the analysis of single-cell CRISPR screens. Currently, SCEPTRE does not estimate the effect sizes

of perturbations, disentangle interactions among perturbed regulatory elements (Zamanighomi et al., 2019; Norman et al., 2019), or leverage information across gRNAs to improve power. Such extensions could be implemented by harnessing more sophisticated, multivariate models of gRNA detection or applying methods for estimating variable importance in the presence of possibly misspecified models (Zhang and Janson, 2020). The statistical challenges that we identified in this study — specifying an accurate expression model, accounting for technical factors — and the solutions that we proposed — conditional resampling, massively parallel association testing — will help guide the development of future versions of SCEPTRE.

Single-cell CRISPR screens will play a key role in unraveling the regulatory architecture of the noncoding genome (Przybyla and Gilbert, 2021). Technological improvements and methodological innovations will increase the scope, scale, and variety of theses screens over the coming years. For example, screens of candidate CREs could be extended to different, disease-relevant cell types and tissues (although this remains a challenge); new combinatorial indexing strategies, such as scifi-RNA-seq, could enable the scaling-up of such screens to millions of cells (Datlinger et al., 2021); different CRISPR technologies, such as CRISPRa, could enable the activation, rather than repression, of candidate CREs, yielding new insights; and information-rich, multimodal single-cell readouts could strengthen conclusions drawn about regulatory relationships (Pierce et al., 2020). SCEPTRE is a flexible, robust, and efficient method: it has now successfully been applied to three single-cell CRISPR screen datasets, across two technologies (CROP-seq and ECCITE-seq), to map regulatory relationships both in *cis* and in *trans*. We expect SCEPTRE to facilitate the analysis of future single-cell screens of the noncoding genome, advancing understading of CREs and enabling the detailed interpretation of GWAS results.

2.4 Figures

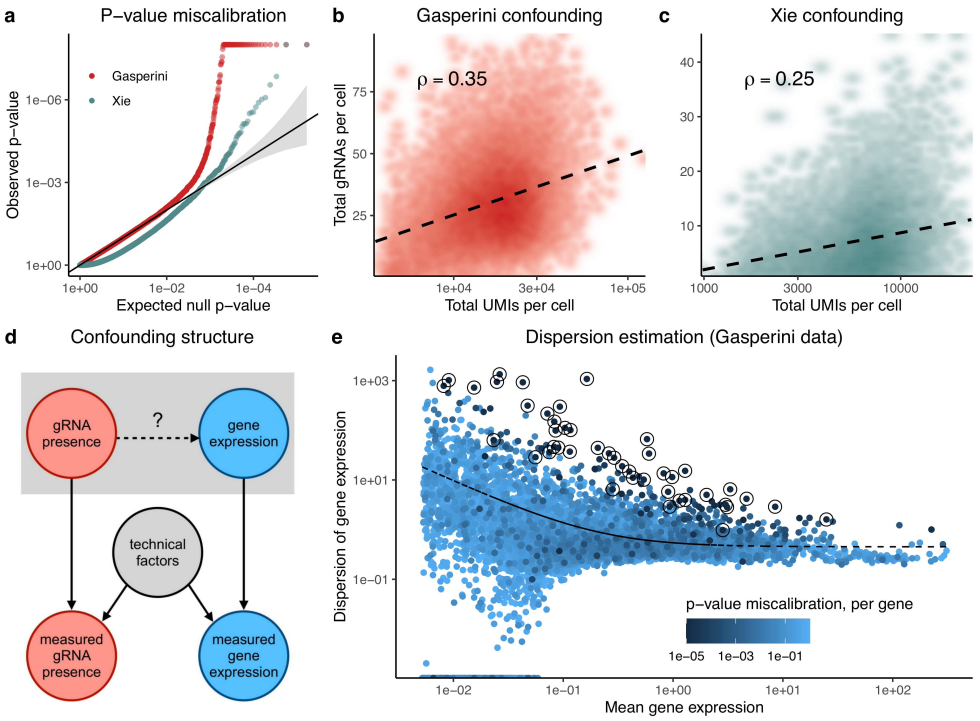


Figure 2.1: **CRISPR screen analysis challenges can lead to false positives and false negatives**. **a**, QQ-plot of negative control p-values produced by Gasperini et al. (red; downsampled for visualization) and Xie et al. (gray-green). These p-values deviate substantially from the expected uniform distribution, indicating test miscalibration. **b-d**, Technical factors, such as sequencing depth and batch, impact gRNA detection probability and observed gene expression levels in both Gasperini et al. (b) and Xie et al. (c) data. Thus, technical factors act as confounders (d), differentiating CRISPR screens from traditional differential expression applications. **e**, Monocle2 estimates the dispersion of each gene by projecting each gene's raw dispersion estimate onto the fitted raw dispersion-mean expression curve. This estimation procedure leads to miscalibration for high-dispersion genes.

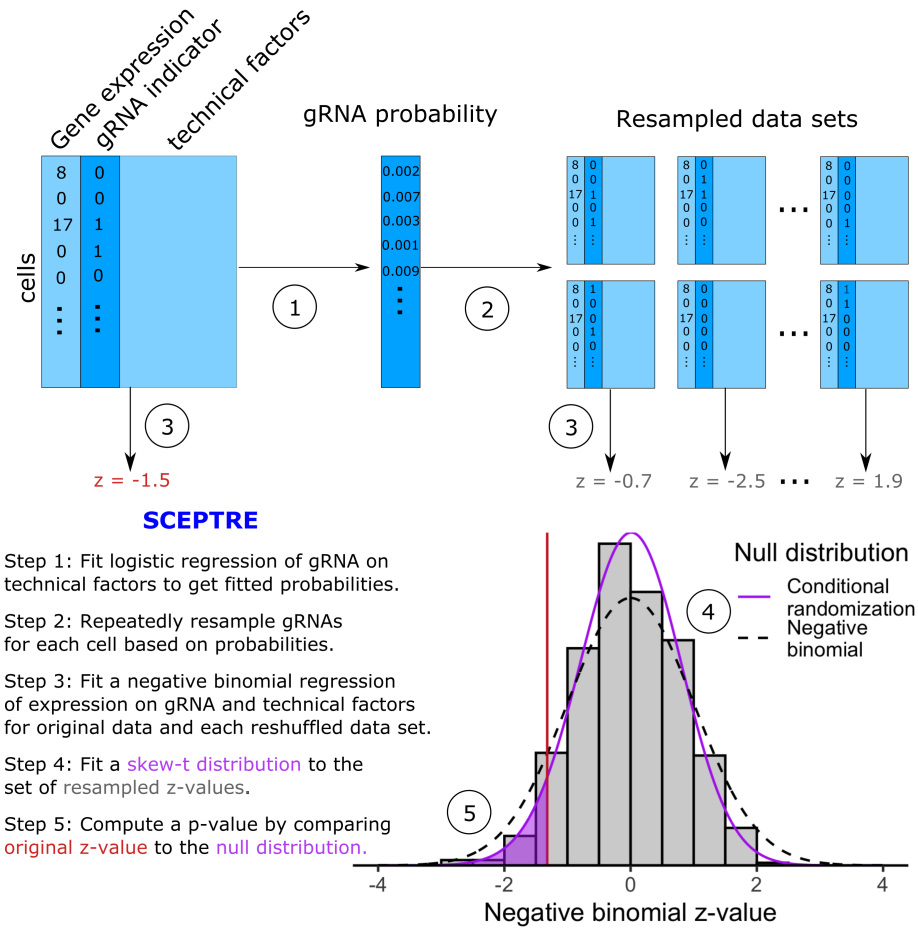


Figure 2.2: **SCEPTRE: Analysis of single-cell perturbation screens via conditional resampling.** A schematic and outline of the SCEPTRE methodology for one gene and one gRNA. SCEPTRE estimates the probability of gRNA detection in each cell based on its technical factors. It then builds a null distribution for the negative binomial z-value by independently resampling gRNA presence or absence for each cell according to these probabilities to form "negative control" datasets. A skew-t distribution is fit to the resulting histogram to obtain precise p-values based on a limited number of resamples, against which the original NB z-value is compared. The dashed line shows the standard normal distribution, against which the NB z-value typically would be compared.

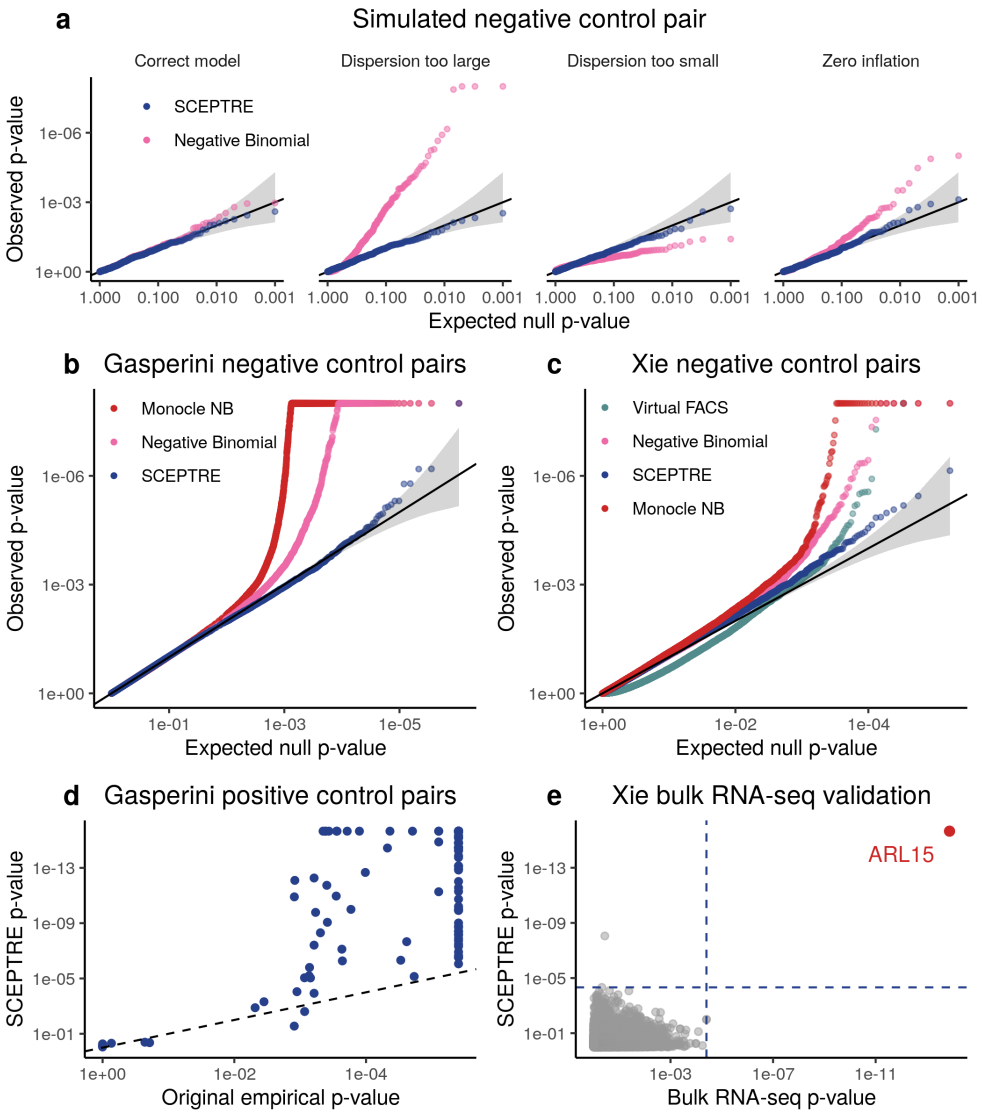
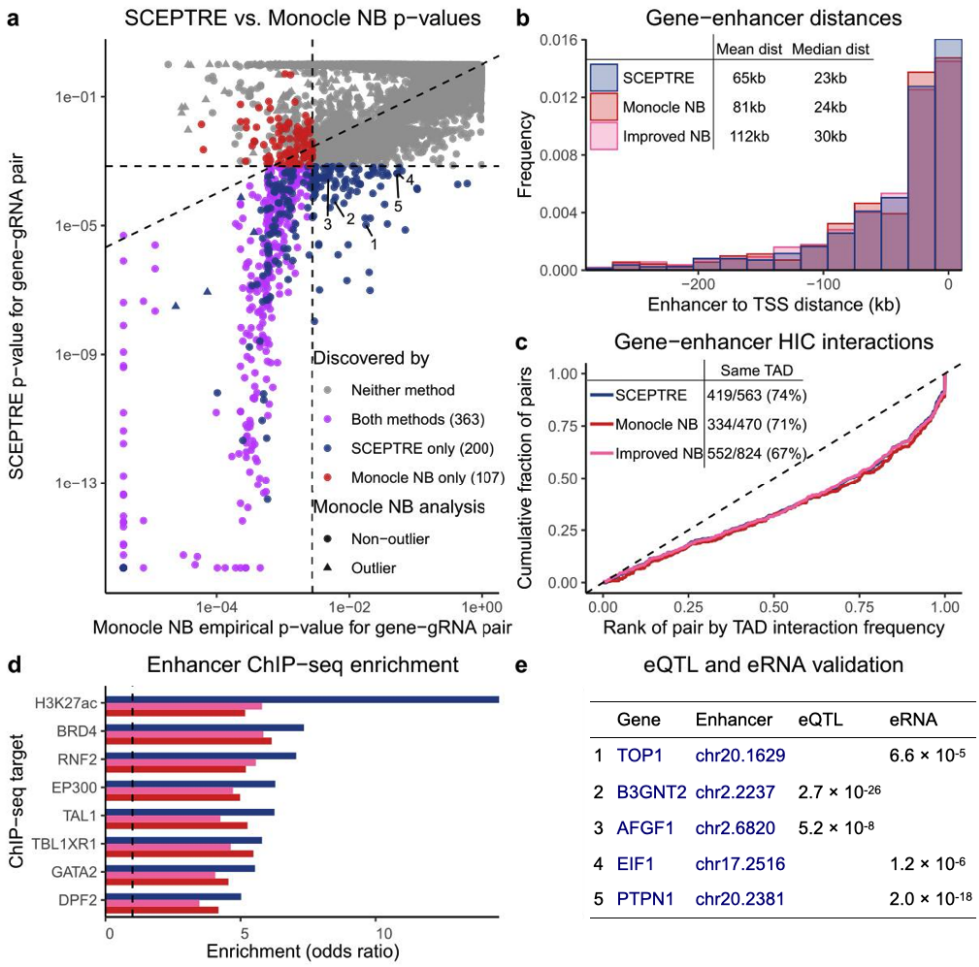


Figure 2.3: (Caption on next page.)

Figure 2.3: **SCEPTRE demonstrates good calibration and sensitivity under known ground truth. a**, Numerical simulation comparing SCEPTRE and improved negative binomial regression on four simulated datasets. The negative binomial model was incorrectly specified on three of the four datasets. SCEPTRE maintained good calibration across all four datasets, despite model misspecification and confounder presence. **b-c**, Application of SCEPTRE, improved negative binomial regression, Monocle regression, and Virtual FACS to pairs of negative control gRNAs and genes in (b) the Gasperini et al. data, and (c) the Xie et al. data. Compared to the other methods, SCEPTRE showed good calibration. **d**, SCEPTRE p-values for Gasperini et al. TSS-targeting controls were highly significant, and in general, more significant than the original empirical p-values. **e**, Comparison of p-values produced by SCEPTRE for ARL15-enh to p-values produced by an arrayed, bulk RNA-seq CRISPR screen of ARL15-enh. The results of the two analyses coincided almost exactly, with both analyses rejecting gene *ARL15* with high confidence after a BH correction. Dotted blue lines, rejection thresholds.



	Gene	Enhancer	eQTL	eRNA
1	TOP1	chr20.1629		6.6×10^{-5}
2	B3GNT2	chr2.2237	2.7×10^{-26}	
3	AFGF1	chr2.6820	5.2×10^{-8}	
4	EIF1	chr17.2516		1.2×10^{-6}
5	PTPN1	chr20.2381		2.0×10^{-18}

Figure 2.4: (Caption on next page.)

Figure 2.4: **Application of SCEPTRE to Gasperini et al. data yields biologically plausible gene-enhancer links. a,** Comparison of the original empirical p-values to those obtained from SCEPTRE. The two analysis methods differed substantially, with 200 gene-enhancer links discovered only by SCEPTRE and 107 discovered only by the original analysis. Annotations correspond to pairs in panel (e). **b,** Distribution of distances from TSS to upstream paired enhancers. Compared to Monocle NB (original) and improved NB analyses, SCEPTRE paired genes with nearer enhancers on average. **c,** For those gene-enhancer pairs falling in the same TAD, the cumulative distribution of the fractional rank of the HI-C interaction frequency compared to other distance-matched loci pairs within the same TAD. SCEPTRE showed similar enrichment despite finding 25% more within-TAD pairs compared to the original analysis. Inset table shows gene-enhancer pairs falling in the same TAD. SCEPTRE found 93 more total pairs compared to the original analysis, and a higher percentage of pairs fell within the same TAD. **d,** Enrichment of ChIP-seq signal from seven cell-type relevant transcription factors and one histone mark (H3K27ac) among paired enhancers. SCEPTRE showed stronger enrichment across all ChIP-seq targets. **e**, Five gene-enhancer pairs discovered by SCEPTRE but not the original analysis, each supported by a whole blood GTEx eQTL or FANTOM enhancer RNA correlation p-value.

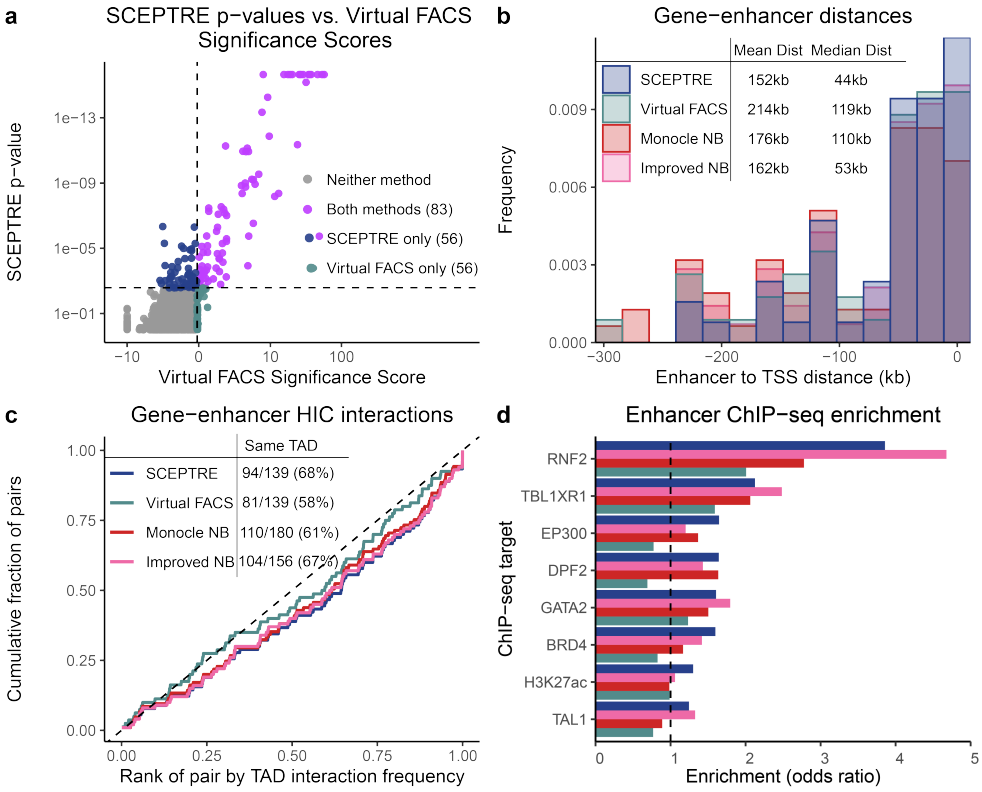


Figure 2.5: **SCEPTRE discovers biologically plausible gene-enhancer links on Xie et al. data. a,** Comparison of SCEPTRE p-values to Virtual FACS significance scores. Significant SCEPTRE p-values ($n = 139$) are colored in blue and purple, and the top 139 Virtual FACS pairs, as ranked by significance score, are colored in gray-green and purple. The two sets diverged substantially, with only 43% of pairs in shared across sets. **b-d,** These panels are similar to the corresponding panels in Figure 2.4. SCEPTRE pairs showed strong enrichment for biological signals associated with enhancer activity on (b) physical distance, (c) HIC interaction, and (d) ChIP-seq metrics relative to other methods.

2.5 Methods

Gasperini et al. and Xie et al. data

Gasperini et al. used CROP-seq (Datlinger et al., 2017; Hill et al., 2018) to transduce a library of CRISPR guide RNAs (gRNAs) into a population of 207,324 K562 cells expressing the Cas9-KRAB repressive complex at a high multiplicity of infection. Each cell received an average of 28 perturbations. The gRNA library targeted 5,779 candidate enhancers, 50 negative controls, and 381 TSS-targeting positive controls. Xie et al. used Mosaic-seq (Xie et al., 2019a, 2017) to perturb at a high multiplicity of infection 518 putative enhancers in a population of 106,670 Cas9-KRAB-expressing K562 cells. Each putative enhancer was perturbed in an average of 1,276 cells.

Cis and *in silico* negative control pairs for Xie et al. data

We generated the set of candidate *cis* gene-enhancer relationships on the Xie et al. data by pairing each protein-coding gene with each gRNA targeting a site within 1 Mb of the TSS of the gene. This procedure resulted in 3,553 candidate *cis* gene-enhancer links that we tested using SCEPTRE and Virtual FACS.

To generate the set of *in silico* negative control pairs for calibration assessment, we (i) identified gRNAs that targeted sites far (> 1 Mb) from the TSSs of known transcription factor genes and (ii) paired these gRNAs with genes located on other chromosomes. We excluded all pairs containing genes known to be transcription factors, and we downsampled the pairs so that each gRNA was matched to 500 genes. The final *in silico* negative control set consisted of 84,500 pairs, the elements of which were not expected to exhibit a regulatory relationship.

Conditional randomization test

Consider a given gene-gRNA pair. For each cell $i = 1, \ldots, n$, let $X_i \in \{0, 1\}$ indicate whether the gRNA was present in the cell, let $Y_i \in \{0, 1, 2, \ldots\}$ be the gene expression in the cell, defined as the number of unique molecular identifiers (UMIs) from this gene, and let $Z_i \in \mathbb{R}^d$ be a list of cell-level technical factors. Letting $(X, Y, Z) = \{(X_i, Y_i, Z_i)\}_{i=1}^n$, consider any test statistic $T(X, Y, Z)$ measuring the effect of the gRNA on the expression of the gene. The conditional randomization test (Candès et al., 2018b) is based on resampling the gRNA indicators independently for each cell. Letting $\pi_i = \mathbb{P}[X_i = 1 | Z_i]$, define random variables

$$\widetilde{X}_i \overset{\text{ind}}{\sim} \text{Ber}(\pi_i). \tag{2.1}$$

Then, the CRT p-value is given by

$$p_{\text{CRT}} = \mathbb{P}[T(\widetilde{X}, Y, Z) \geq T(X, Y, Z) \mid X, Y, Z]. \tag{2.2}$$

This translates to repeatedly sampling $\widetilde{X}$ from the distribution (2.1), recomputing the test statistic with X replaced by $\widetilde{X}$, and defining the p-value as the probability the resampled test statistic exceeds the original. Under the null hypothesis that the gRNA perturbation does not impact the cell (adjusting for technical factors), i.e. $Y \perp\!\!\!\perp X \mid Z$, we obtain a valid p-value (2.2), *regardless of the expression distribution* $Y|X, Z$ *and regardless of the test statistic* T. We choose as a test statistic T the z-score of X_i obtained from a negative binomial regression of Y_i on X_i and Z_i:

$$Y_i \overset{\text{ind}}{\sim} \text{NegBin}(\mu_i, \alpha); \quad \log(\mu_i) = \beta_0 + X_i\beta + Z_i^T\gamma, \tag{2.3}$$

where α is the dispersion. Following Hafemeister and Satjia (Hafemeister and Satija, 2019), we estimate α by pooling dispersion information across genes, and we include sequencing depth as an entry in the vector of technical factors Z_i (see section *Improvements to the negative binomial approach*).

Accelerations to the conditional randomization test

We implemented computational accelerations to the conditional randomization test. First, we employed the recently proposed (Liu et al., 2020) *distillation* technique to accelerate the recomputation of the negative binomial regression for each resample. The idea is to use a slightly modified test statistic, consisting of two steps:

1. Fit $(\widehat{\beta}_0, \widehat{\gamma})$ from the negative binomial regression (2.3) except without the gRNA term:

$$Y_i \overset{\text{ind}}{\sim} \text{NegBin}(\mu_i, \alpha); \quad \log(\mu_i) = \beta_0 + Z_i^T\gamma. \tag{2.4}$$

2. Fit $\widehat{\beta}$ from a negative binomial regression with the estimated contributions of Z_i from step 1 as offsets:

$$Y_i \overset{\text{ind}}{\sim} \text{NegBin}(\mu_i, \alpha); \quad \log(\mu_i) = X_i\beta + \widehat{\beta}_0 + Z_i^T\widehat{\gamma}. \tag{2.5}$$

Conditional randomization testing with this two step test statistic, which is nearly identical to the full negative binomial regression (2.3), is much faster. Indeed, since the first step is not a function of X_i, it remains the same for each resampled triple $(\widetilde{X}, Y, Z)$. Therefore, only the second step must be recomputed

with each resample, and this step is faster because it involves only a univariate regression.

Next, we accelerated the second step above using the sparsity of the binary vector $(X_1, \ldots, X_n)$ (or a resample of it). To do so, we wrote the log-likelihood of the reduced negative binomial regression (2.5) as follows, denoting by $\ell(Y_i, \log(\mu_i))$ the negative binomial log-likelihood:

$$\begin{aligned}\sum_{i=1}^{n} \ell(Y_i, X_i\beta + \widehat{\beta}_0 + Z_i^T\widehat{\gamma}) &= \sum_{i:X_i=0} \ell(Y_i, \widehat{\beta}_0 + Z_i^T\widehat{\gamma}) + \sum_{i:X_i=1} \ell(Y_i, \beta + \widehat{\beta}_0 + Z_i^T\widehat{\gamma}) \\ &= C + \sum_{i:X_i=1} \ell(Y_i, \beta + \widehat{\beta}_0 + Z_i^T\widehat{\gamma}).\end{aligned}$$

This simple calculation shows that, up to a constant that does not depend on β, the negative binomial log-likelihood corresponding to the model (2.5) is the same as that corresponding to the model with only intercept and offset term for those cells with a gRNA:

$$Y_i \overset{\text{ind}}{\sim} \text{NegBin}(\mu_i, \alpha); \quad \log(\mu_i) = \beta + \widehat{\beta}_0 + Z_i^T\widehat{\gamma}, \quad \text{for } i \text{ such that } X_i = 1. \tag{2.6}$$

The above negative binomial regression is therefore equivalent to equation (2.5), but much faster to compute, because it involves much fewer cells. For example, in the Gasperini et al. data, each gRNA is observed in only about 1000 of the 200,000 total cells.

SCEPTRE methodology

In practice, we must estimate the gRNA probabilities π_i as well as the p-value p_{CRT}. This is because usually we do not know the distribution $X|Z$ and cannot compute the conditional probability in equation (2.2) exactly. We propose to estimate π_i via logistic regression of X on Z, and to estimate p_{CRT} by resampling $\widetilde{X}$ a large number of times and then fitting a skew-t distribution to the resampling null distribution $T(\widetilde{X}, Y, Z)|X, Y, Z$. We outline SCEPTRE below:

1. Fit technical factor effects $(\widehat{\beta}_0, \widehat{\gamma})$ on gene expression using the negative binomial regression (2.4).
2. Extract a z-score $z(X, Y, Z)$ from the reduced negative binomial regression (2.6).
3. Assume that

$$X_i \overset{\text{ind}}{\sim} \text{Ber}(\pi_i); \quad \log\left(\frac{\pi_i}{1-\pi_i}\right) = \tau_0 + Z_i^T\tau \tag{2.7}$$

for $\tau_0 \in \mathbb{R}$ and $\tau \in \mathbb{R}^d$, and fit $(\widehat{\tau}_0, \widehat{\tau})$ via logistic regression of X on Z. Then, extract the fitted probabilities $\widehat{\pi}_i = (1 + \exp(-(\widehat{\tau}_0 + Z_i^T \widehat{\tau})))^{-1}$.

4. For $b = 1, \ldots, B$,
   - Resample the gRNA assignments based on the probabilities $\widehat{\pi}_i$ to obtain $\widetilde{X}^b$ (2.1).
   - Extract a z-score $z(\widetilde{X}^b, Y, Z)$ from the reduced negative binomial regression (2.6).
5. Fit a skew-t distribution $\widehat{F}_{\text{null}}$ to the resampled z-scores $\{z(\widetilde{X}^b, Y, Z)\}_{b=1}^B$.
6. Return the p-value $\widehat{p}_{\text{SCEPTRE}} = \mathbb{P}[\widehat{F}_{\text{null}} \leq z(X, Y, Z)]$.

In our data analysis we used $B = 500$ resamples. Note that, in the causal inference literature, the distribution $X|Z$ is called the *propensity score* (Rosenbaum and Rubin, 1983).

Numerical simulation to assess calibration

We simulated one gene Y_i, one gRNA X_i, and two confounders Z_{i1}, Z_{i2} in $n = 1000$ cells. We generated the confounders Z_{i1} and Z_{i2} by sampling with replacement the batch IDs and log-transformed sequencing depths of the cells in the Gasperini dataset. The batch ID confounder Z_{i1} was a binary variable, as the Gasperni data included two batches. Next, we drew the gRNA indicators X_i i.i.d. from the logistic regression model (7), with $\tau_0 = -7, \tau_1 = -2$, and $\tau_2 = 0.5$. We selected these parameters to make the probability of gRNA occurrence about 0.04 across cells. Finally, we drew the gene expression Y_i from the following zero-inflated negative binomial model:

$$Y_i \sim \lambda \delta_0 + (1 - \lambda)\text{NegBin}(\mu_i, \alpha), \;\; \log(\mu_i) = \beta_0 + Z_i^T \beta.$$

Note that gRNA presence or absence does not impact gene expression in this model. We set $\beta_0 = -2.5, \beta_1 = -2, \beta_2 = 0.5$ to make the average gene expression about 4 across cells. We generated the four datasets shown in Figure 2.3a by setting the dispersion parameter α and the zero inflation rate parameter λ equal to the following values:

$$(\lambda_1, \alpha_1) = (0, 1); \;\; (\lambda_2, \alpha_2) = (0, 5); \;\; (\lambda_3, \alpha_3) = (0, 0.2); \;\; (\lambda_4, \alpha_4) = (0.25, 1).$$

For the first, the negative binomial model is correctly specified. For the second and third, the dispersion estimate of 1 is too small and too large, respectively. The last setting exhibits zero inflation. We applied SCEPTRE

and negative binomial regression to the four problem settings, each with $n_{sim} = 500$ repetitions. The negative binomial method, and in turn SCEPTRE, was based on the z statistic from the Hafemeister-inspired negative binomial model (2.3) with $\alpha = 1$. We used $B = 500$ resamples for SCEPTRE, the default choice.

scMAGeCK

scMAGeCK-LR (Yang et al., 2020) is a method for high MOI single-cell CRISPR screen analysis. (A complimentary method, scMAGeCK-RRA, is designed for the low-MOI setting) scMAGeCK-LR (henceforth scMAGeCK) uses a permutation test with ridge regression test statistic to compute p-values for pairs of genes and gRNAs. Unfortunately, we were unable to apply scMAGeCK to the real data. First, we were unable to understand the documentation of the scMAGeCK software well enough to confidently apply the method. Second, scMAGeCK is prohibitively expensive to apply at-scale. The authors of the original scMAGeCK study applied their method only to a small subset of pairs in the Gasperini et al. data. We could not meaningfully compare scMAGeCK to SCEPTRE on calibration and sensitivity metrics without applying scMAGeCK to the full set of gRNA-gene pairs, which, to our knowledge, never has been done (and likely is infeasible).

To enable a simple comparison to scMAGeCK on the simulated data, we implemented a custom, in-house version of scMAGeCK based on a careful examination of the scMAGeCK codebase and a close reading of the original paper. We view this custom implementation as a faithful interpretation of the method in the specialized one-gene to one-NTC setting. We applied our implementation of scMAGeCK to the simulated data, using $B = 1,000$ permutations, the default option. To reduce confusion, we reported the results of the scMAGeCK simulation study in the supplementary materials (Figure A.6) rather than the Results section. We could not apply our custom implementation of scMAGeCK to the real data, because the real data are significantly more complex than the simulated data. For example, the real data consist of many genes and gRNAs, and the gRNAs are differently typed (e.g., negative control, positive control, enhancer-targeting, etc.), complicating the analysis considerably.

Definition of Gasperini et al. discovery set

Gasperini et al. reported a total of 664 gene-enhancer pairs, identifying 470 of these as “high-confidence." We chose to use the latter set, rather than the former, for all our comparisons. Gasperini et al. carried out their ChIP-seq and HI-C enrichment analyses only on the high-confidence discoveries, so for those

comparisons we did the same. Furthermore, the 664 total gene-enhancer pairs reported in the original analysis were the result of a BH FDR correction that included not only the candidate enhancers but also hundreds of positive controls. While Bonferroni corrections can only become more conservative when including more hypotheses, BH corrections are known to become anticonservative when extra positive controls are included (Finner and Roters, 2001). To avoid this extra risk of false positives, we chose to use the "high-confidence" set throughout.

Xie et al. significance scores and discovery set

Xie et al. reported a local (or *cis*) discovery set, which consisted of gene-gRNA pairs with a significance score of greater than zero (see original manuscript for definition of "significance score;" cutoff of zero arbitrary). This discovery set was not directly comparable to the SCEPTRE discovery set. First, the candidate set of *cis* gene-gRNA pairs tested by Xie et al. consisted of gRNAs within *two* Mb of a protein-coding gene *or* long-noncoding RNA. Our candidate *cis* set, by contrast, consisted of gRNAs within *one* Mb of a protein-coding gene. We defined our candidate *cis* set differently than Xie et al. to mantain consistency with Gasperini et al. Second, Xie et al. appear to have used a significantly more conservative threshold than Gasperini et al. in defining their discovery set, but this was challenging to ascertain given the impossibility of FDR correction on the significance scores. To enable a meaningful comparison between Virtual FACS and SCEPTRE, we therefore ranked the Virtual FACS pairs by their significance score and selected the top n pairs, where n was the size of the SCEPTRE discovery set at FDR 0.1.

ChIP-seq, HI-C enrichment analyses

ChIP-seq and HI-C enrichment analyses analyses on the Gasperini et al. data (see Figures 2.4e-f and A.5a) were carried out almost exactly following Gasperini et al. The only change we made is in our quantification of the ChIP-seq enrichment (Figure 2.4f). We used the odds ratio of a candidate enhancer being paired to a gene, comparing the top and bottom ChIP-seq quintiles. On the Xie et al. data, we binned the candidate enhancers into two (rather than five) quantiles due to the fewer number of candidate *cis* pairs. We computed odds ratios by comparing enhancers in the upper quantile to those that did not intersect a ChIP-seq peak at all (Figure A.5b).

Chapter acknowledgements

We thank Molly Gasperini, Jacob Tome, and Andrew Hill for clarifying several aspects of their data analysis (Gasperini et al., 2019) and to the Shendure lab for providing extensive feedback on an earlier draft of this paper. We

thank Shiqi Xie for providing guidance on using the Xie et al. (2019a) data and Wei Li for a helpful discussion on scMAGeCK. Finally, we thank Tom Norman, Atray Dixit, and Wesley Tansey for useful discussions on single-cell CRISPR screens. This work was supported, in part, by National Institute of Mental Health (NIMH) grant R01MH123184 as well as SFARI Grant 575547. Part of the data analysis used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns et al., 2014), which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges system (Nystrom et al., 2015), which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

Three

Effect size estimation at high multiplicity-of-infection

3.1 Introduction

In Chapter 2 we studied the problem of association testing for high multiplicity-of-infection (MOI) single-cell CRISPR screen data. We proposed SCEPTRE, a well-calibrated and powerful test of association based on the conditional randomization test. In this Chapter we shift our focus to estimation. We seek to estimate (with confidence) the effect size of a given CRISPR perturbation on the expression of a given gene in the high-MOI setting. We propose an estimation method GLM-EIV ("GLM-based errors in variables") for this purpose.

An analysis challenge in high-MOI single-cell CRISPR screens — one that we ignored in Chapter 2 for simplicity — is that the CRISPR perturbation is assigned randomly to cells and is not directly observable. As a consequence, one cannot know with certainty which cells were perturbed. Instead, one must leverage an indirect, quantitative proxy of perturbation presence or absence to "guess" which cells received a perturbation. This indirect proxy takes the form of a so-called guide RNA count, with higher counts indicating that a cell is more likely to have been perturbed. The standard approach to single-cell CRISPR screen analysis is to impute perturbation assignments onto the cells by simply thresholding the guide RNA counts; using these imputations, one can attempt to estimate the effect of the perturbation on gene expression. We call this standard approach "thresholded regression" or the "thresholding method."

We study the estimation problem in high-MOI from a statistical perspective, formulating the data generating mechanism using a new class of measurement error models. We assume that the response variable y is a GLM of an underlying

This Chapter is based on Barry et al. (2022), which is joint work with Kathryn Roeder and Eugene Katsevich.

predictor variable x^* and vector of confounders z. We do not observe x^* directly; rather, we observe a noisy version x of x^* that itself is a GLM of x^* and the same set of confounders z. The goal of the analysis is to estimate the effect of x^* on y using the observed data (x, y, z) only. In the context of the biological application, x^*, x, y, and z are CRISPR perturbations, guide RNA counts, gene expressions, and technical confounders, respectively.

This Chapter makes two main contributions. First, we conduct a detailed study of the thresholding method. Notably, we demonstrate on real data that the thresholding method exhibits attenuation bias and a bias-variance tradeoff as a function of the selected threshold, and we recover these phenomena in precise mathematical terms in a simplified Gaussian setting. Second, we introduce a new method, GLM-EIV ("GLM-based errors-in-variables"), for single-cell CRISPR screen analysis. GLM-EIV extends the classical errors-in-variables model (Carroll et al., 2006) to responses and noisy predictors that are exponential family-distributed and potentially impacted by the same set of confounding variables. GLM-EIV thereby implicitly estimates the probability that each cell was perturbed, obviating the need to explicitly impute perturbation assignments via thresholding. We implement several statistical accelerations (possibly of independent interest) to bring the cost of GLM-EIV down to within about an order of magnitude of the thresholding method.

Finally, we develop a Docker-containerized application to deploy GLM-EIV at-scale across tens or hundreds of processors on clouds (e.g., Microsoft Azure) and high-performance clusters. Leveraging this application, we apply GLM-EIV to analyze two recent, large-scale, single-cell CRISPR screen datasets. We find that in some settings, GLM-EIV outperforms thresholded regression by a considerable margin; in other settings the two methods work best in conjunction, with GLM-EIV providing a statistically principled and empirically effective procedure for selecting the threshold.

3.2 Assay background

The human genome consists of genes, enhancers (segments of DNA that regulate the expression of one or more genes), and other genomic elements (that are not of relevance to the current Chapter). GWAS have revealed that the majority ($> 90\%$) of variants associated with diseases lie outside genes and inside enhancers (Gallagher and Chen-Plotkin, 2018). These noncoding variants are thought to contribute to disease by modulating the expression of one or more disease-relevant genes. Scientists do not know the gene (or genes) through which most noncoding variants exert their effect, limiting the interpretability of GWAS results. A central open challenge in genetics, therefore, is to link enhancers

that harbor GWAS variants to the genes that they target at genome-wide scale (Morris et al., 2023).

High MOI single-cell CRISPR screens are the most promising biotechnology for solving this challenge. High MOI single-cell CRISPR screens combine CRISPR interference (CRISPRi) — a version of CRISPR that represses a targeted region of the genome — with single-cell sequencing. The experimental protocol is as follows. First, the scientist develops a library of several hundred to several thousand CRISPRi perturbations, each designed to target a candidate enhancer for repression. The scientist then cultures tens or hundreds of thousands of cells and delivers the CRISPRi perturbations to these cells. The perturbations assort into the cells randomly, with each cell receiving on average 10-40 distinct perturbations. Conversely, a given perturbation enters about 0.1-2% of cells (this Chapter).

After waiting several days for CRISPRi to take effect, the scientist profiles each cell's transcriptome (i.e., its gene expressions) and the set of perturbations that it received. Finally, the scientist conducts perturbation-to-gene association analyses. Figure 3.1a depicts this process schematically, with colored bars (blue, red, and purple) representing distinct perturbations. For a given perturbation (e.g., the perturbation represented in blue), the scientist partitions the cells into two groups: those that received the perturbation (top) and those that did not (bottom). Next, for a given gene, the scientist runs a differential expression analysis across the two groups of cells, producing an estimate for the magnitude of the gene expression change in response to the perturbation. If the estimated change in expression is large, the scientist can conclude that the enhancer *targeted* by the perturbation exerts a strong regulatory effect on the gene. This procedure is repeated for a large set of preselected perturbation-gene pairs. The enhancer-by-enhancer approach is valid because the perturbations assort into cells approximately independently of one another.

The genomics literature has produced a few applied methods for single-cell CRISPR screen analysis (Gasperini et al., 2019; Xie et al., 2019b; Barry et al., 2021a). Gasperini et al. applied negative binomial GLMs (as implemented in the Monocle software; Trapnell et al. 2014) to carry out the differential expression analysis described above. Xie et al., by contrast, applied chi-squared-like tests of independence for this purpose. Both of these approaches have limitations: the former is not robust to misspecification of the gene expression model, and the latter is unable to correct for the presence of technical confounders. In Chapter 2 we introduced SCEPTRE, a custom implementation of the conditional randomization test (Candès et al., 2018a; Liu et al., 2021) tailored to single-cell CRISPR screen data. SCEPTRE simultaneously adjusts for confounder presence and ensures robustness to expression model misspecification,

overcoming limitations of the prior methods and demonstrating state-of-the-art sensitivity and specificity on single-cell CRISPR screen data. In this Chapter we tackle a set of analysis challenges complimentary to those addressed by SCEPTRE. Most importantly, we seek to account for the fact that the perturbation is measured with noise, an issue that all available methods (including SCEPTRE) assume away via thresholding. Additionally, we seek to *estimate* (with confidence) the effect size of a perturbation on gene expression change, an objective that is challenging to attain within the nonparametric hypothesis testing framework of SCEPTRE.

3.3 Analysis challenges and proposed statistical model

High MOI single-cell CRISPR screens present several statistical challenges, four of which we highlight here. Throughout, we consider a single perturbation-gene pair. First, the "treatment" variable — i.e., the presence or absence of a perturbation — cannot be directly observed. Instead, perturbed cells transcribe molecules called *guide RNAs* (or *gRNAs*) that serve as indirect proxies of perturbation presence. We must leverage these gRNAs to impute (explicitly or implicitly) perturbation assignments onto the cells (Figure 3.1b). Second, "technical factors" — sources of variation that are experimental rather than biological in origin — impact the measurement of both gene and gRNA expressions and therefore act as confounders (Figure 3.1b). Third, the gene and gRNA data are sparse, discrete counts. Consequently, classical statistical approaches that assume Gaussianity or homoscedasticity are inapplicable. Finally, sequenced gRNAs sometimes map to cells that have not received a perturbation. This phenomenon, which we call "background contamination," results from errors in the sequencing and alignment processes. The marginal distribution of the gRNA counts is best conceptualized as a mixture model (Figure 3.1c; Gaussian distributions used for illustration purposes only). Unperturbed and perturbed cells both exhibit nonzero gRNA count distributions, but this distribution is shifted upward for perturbed cells. Figure 3.1d shows example data on four (of possibly tens or hundreds of thousands of) cells. The analysis objective is to leverage the gene expressions and gRNA counts to estimate the effect of the (latent) perturbation on gene expression, accounting for the technical factors.

We propose to model the single-cell CRISPR screen data-generating process using a pair of GLMs. Let $n \in \mathbb{N}$ be the number of cells assayed in the experiment. Consider a single perturbation and a single gene. For cell $i \in \{1, \ldots, n\}$, let $p_i \in \{0, 1\}$ indicate perturbation presence or absence; let $m_i \in \mathbb{N}$ be the number of gene transcripts sequenced; let $g_i \in \mathbb{N}$ be the number of gRNA transcripts sequenced; let $d_i^m \in \mathbb{N}$ be the number of gene transcripts

sequenced across *all* genes (i.e., the library size or sequencing depth); let d_i^g be the gRNA library size; and finally, let $z_i \in \mathbb{R}^{d-1}$ be the cell-specific technical factors (e.g., sequencing batch, percent mitochondrial reads, etc.) The letters "m," "g", and "d" stand for "mRNA," "gRNA," and "depth," respectively.

Building on the work of several previous authors (Townes et al., 2019; Hafemeister and Satija, 2019), Sarkar and Stephens (2021) proposed a simple strategy for modeling single-cell gene expression data, which, in the framework of negative binomial GLMs, is equivalent to using the log-transformed library size as an offset term. Sarkar and Stephens' framework enjoys strong theoretical and empirical support; therefore, we generalize their approach to model *both* gene and gRNA modalities in single-cell CRISPR screen experiments. To this end, we assume that the gene expression counts are given by

$$m_i|(p_i, z_i, d_i^m) \sim \mathrm{NB}_{s^m}(\mu_i^m); \quad \log(\mu_i^m) = \beta_0^m + \beta_1^m p_i + \gamma_m^T z_i + \log(d_i^m), \tag{3.1}$$

where (i) $\mathrm{NB}_{s^m}(\mu_i^m)$ is a negative binomial distribution with mean μ_i^m and known size parameter s^m; (ii) $\beta_0^m \in \mathbb{R}, \beta_1^m \in \mathbb{R}$, and $\gamma_m \in \mathbb{R}^{d-1}$ are unknown parameters; and (iii) $\log(d_i^m)$ is an offset term. Similarly, we model the gRNA counts by

$$g_i|(p_i, z_i, d_i^g) \sim \mathrm{NB}_{s^g}(\mu_i^g); \quad \log(\mu_i^g) = \beta_0^g + \beta_1^g p_i + \gamma_g^T z_i + \log(d_i^g), \tag{3.2}$$

where μ_i^g, s^g, β_0^g, β_1^g, γ_g, and d_i^g are analogous. We use a negative binomial GLM to model the gRNA counts as well as the gene expressions because the gRNA transcripts are generated via the same biological mechanism as the gene transcripts (Datlinger et al., 2017; Hill et al., 2018). Finally, we model the marginal perturbation probability as

$$p_i \sim \mathrm{Bern}(\pi),$$

where $\pi \in (0, 1/2]$, and p_i is unobserved. (We restrict π to the interval $(0, 1/2]$ so that the model is identifiable and thus estimable. This restriction is reasonable biologically as well as statistically, as a each perturbation infects a small fraction of cells.) Together, (3.1), (3.2), and the marginal distribution of p_i define the negative binomial GLM-EIV model.

The log-transformed sequencing depth $\log(d_i^m)$ is included as an offset term in (3.1) so that $\beta_0^m + \beta_1^m p_i + \gamma_m^T z_i$ can be interpreted as a relative expression. Exponentiating both sides of (3.1) reveals that the mean gene expression μ_i^m of the ith cell is $\exp\left(\beta_0^m + \beta_1^m p_i + \gamma_m^T z_i\right) d_i^m$. Because d_i^m is the sequencing depth, $\exp\left(\beta_0^m + \beta_1^m p_i + \gamma_m^T z_i\right)$ is the *fraction* of all transcripts sequenced in the cell produced by the gene under consideration. The target of inference β_1^m

is the log fold change in expression in response to the perturbation, controlling for the technical factors. Fold change in this context is the ratio of the mean gene expression in perturbed cells to the mean gene expression in unperturbed cells. Hence, $\exp(\beta_1^m) = 1$ (i.e., $\beta_1^m = 0$) indicates no change in expression, whereas $\exp(\beta_1^m) > 1$ (i.e., $\beta_1^m > 0$) and $\exp(\beta_1^m) < 1$ (i.e., $\beta_1^m < 0$) indicate an increase and decrease in expression, respectively.

In this Chapter we analyze two large-scale, high MOI, single-cell CRISPR screen datasets published by Gasperini et al. (2019) and Xie et al. (2019b). Gasperini (resp., Xie) targeted approximately 6,000 (resp., 500) candidate enhancers in a population of approximately 200,000 (resp., 100,000) cells. Gasperini additionally designed several hundred positive control, gene-targeting perturbations and 50 non-targeting, negative control perturbations to assess method sensitivity and specificity.

3.4 Analysis of the thresholding method

We study thresholding from empirical and theoretical perspectives, highlighting several limitations of the approach. In the context of the negative bionomial GLM-EIV model introduced above (3.1-3.2), the thresholding method leverages the gRNA counts (3.2) to impute the latent perturbation indicator (3.2), thereby reducing the full data generating process to a single, gene expression model (3.1). We study Gasperini et al.'s variant of the thresholding method (i.e., thresholded negative binomial regression), as this version of the thresholding method relates most closely to GLM-EIV. The method is defined as follows:

1. For a given threshold $c \in \mathbb{N}$, let the imputed perturbation assignment $\hat{p}_i \in \{0, 1\}$ be given by $\hat{p}_i = 0$ if $g_i < c$ and $\hat{p}_i = 1$ otherwise.

2. Assume that m_i is related to $\hat{p}_i, d_i^m$, and z_i through the following GLM:

$$\begin{cases} m_i | (\hat{p}_i, z_i, d_i^m) \sim \mathrm{NB}_{s^m}(\mu_i^m) \\ \log(\mu_i^m) = \beta_0^m + \beta_1^m \hat{p}_i + \gamma_m^T z_i + \log\left(d_i^m\right). \end{cases} \tag{3.3}$$

   The model (3.3) is equivalent to the model (3.2), but the latent perturbation indicator p_i has been replaced by the imputed perturbation indicator $\hat{p}_i$.

3. Fit a GLM to (3.3) to obtain an estimate and CI for the target of inference β_1^m.

To shed light on empirical challenges of the thresholding method, we applied thresholded negative binomial regression to analyze the set of positive control perturbation-gene pairs in the Gasperini dataset. The positive control pairs consisted of perturbations that targeted gene transcription start sites (TSSs) for inhibition. Repressing the TSS of a given gene decreases its expression; therefore, the positive control pairs *a priori* are expected to exhibit a strong decrease in expression.

To investigate the sensitivity of the thresholding method to threshold choice, we deployed the method using three different choices for the threshold: 1, 5, and 20. We found that the chosen threshold substantially impacted the results (Figure 3.2a-b): estimates for fold change produced by threshold = 1 were smaller in magnitude (i.e., closer to the baseline of 1) than those produced by threshold = 5. (Figure 3.2a.) On the other hand, estimates produced by threshold = 5 and threshold = 20 were more concordant (Figure 3.2b).

We reasoned that thresholded regression systematically underestimated true effect sizes on the positive control pairs, especially for threshold = 1. For a given perturbation, the majority ($> 98\%$) of cells are unperturbed. This imbalance leads to an asymmetry: misclassifying *unperturbed* cells as *perturbed* is intuitively "worse" than misclassifying *perturbed* cells as *unperturbed*. Misclassified unperturbed cells contaminate the set of truly perturbed cells, leading to attenuation bias; by contrast, misclassified perturbed cells are swamped in number and "neutralized" by the truly unperturbed cells. Setting the threshold to a large number reduces the unperturbed-to-perturbed misclassification rate, decreasing bias.

We hypothesized, however, that the reduction in bias obtained by selecting a large threshold causes the variance of the estimator to increase. To investigate, we compared p-values and confidence intervals produced by threshold = 5 and threshold = 20 for the target of inference β_1^m. We found that threshold = 5 yielded smaller (i.e., more significant) p-values and narrower confidence intervals than did threshold = 20 (Figures 3.2c-d). We concluded that the threshold controls a bias-variance tradeoff: as the threshold increases, the bias of the estimator decreases and the variance increases.

Finally, to determine whether there is an "obvious" location at which to draw the threshold, we examined the empirical gRNA count distributions and checked for bimodality. Figures 3.2e and 3.2f display the empirical distribution of a randomly-selected gRNA from the Gasperini and Xie datasets, respectively (counts of 0 omitted). The distributions peak at 1 and then taper off gradually; there does not exist a sharp boundary that cleanly separates the perturbed from the unperturbed cells. Overall, we concluded that the thresholding method faces several challenges: (i) the threshold is a tuning parameter that

significantly impacts the results; (ii) the threshold mediates an intrinsic bias-variance tradeoff; and (iii) the gRNA count distributions do not imply a clear threshold selection strategy.

Next, we studied the thresholding method from a theoretical perspective, recovering in a simplified Gaussian setting phenomena revealed in the empirical analysis. Due to space constraints we relegate this analysis to Appendix **??**, but we briefly summarize the main results here. First, we derived an exact expression for the asymptotic relative bias of the thresholding estimator $\hat{\beta}_1^m$. Leveraging this exact expression, we showed that (i) the thresholding estimator strictly underestimates (in absolute value) the true value of β_1^m over all choices of the threshold and over all values of the regression coefficients (an example of *attenuation bias*; Stefanski 2000); and (ii) the magnitude of the bias decreases monotonically in β_1^g, comporting with the intuition that the problem becomes easier as the gRNA mixture distribution becomes increasingly well-separated. Second, we derived an asymptotically exact bias-variance decomposition for $\hat{\beta}_m$, demonstrating that as the threshold tends to infinity, the bias decreases and the variance increases.

3.5 GLM-based errors-in-variables (GLM-EIV)

We introduce the general GLM-EIV model, which generalizes the negative binomial GLM-EIV model (3.1-3.2) to arbitrary exponential family response distributions and link functions, thereby providing much greater modeling flexibility. We derive efficient methods for estimation and inference in this model and develop a pipeline to deploy the model at-scale. Appendix B.3 develops a parallel methodology in which the gRNA counts are modeled as zero-inflated.

Model and model properties

General model. This section is more technical than the previous ones; a first-time reader can skip to Section 3.5 without loss of information relevant to the high-level narrative. Let $\tilde{x}_i = [1, p_i, z_i]^T \in \mathbb{R}^d$ be the vector of covariates (including an intercept term) for the ith cell. (We use the tilde as a reminder that the vector is partially unobserved.) Let $\beta_m = [\beta_0^m, \beta_1^m, \gamma_m]^T \in \mathbb{R}^d$ and $\beta_g = [\beta_0^g, \beta_1^g, \gamma_g]^T \in \mathbb{R}^d$ be the unknown coefficient vectors corresponding to the gene and gRNA expression models, respectively. Finally, let o_i^m and o_i^g be the (possibly zero) offset terms for the gene and gRNA models; in practice, we typically set o_i^m and o_i^g to the log-transformed library sizes (i.e., $\log(d_i^m)$ and $\log(d_i^g)$, respectively).

We use a pair of GLMs to model the gene and gRNA expressions. Considering first the gene expression model, let the ith linear component l_i^m of the model be

$$l_i^m \equiv \langle \tilde{x}_i, \beta_m \rangle + o_i^m.$$

Next, let the mean μ_i^m of the ith observation be

$$r_m(\mu_i^m) \equiv l_i^m,$$

where $r_m : \mathbb{R} \to \mathbb{R}$ is a strictly increasing, differentiable link function. Let $\psi_m : \mathbb{R} \to \mathbb{R}$ be the differentiable, cumulant-generating function of the selected exponential family distribution. We can express the canonical parameter η_i^m in terms of ψ_m and r_m by

$$\eta_i^m = \left([\psi_m']^{-1} \circ r_m^{-1} \right)(l_i^m) \equiv h_m(l_i^m).$$

Finally, let $c_m : \mathbb{R} \to \mathbb{R}$ be the carrying density of the selected exponential family distribution. The density f_m of m_i conditional on the the canonical parameter η_i is

$$f_m(m_i; \eta_i^m) = \exp\left\{ m_i \eta_i^m - \psi_m(\eta_i^m) + c_m(m_i) \right\}.$$

We implicitly set the "scale" term (i.e., the $a(\phi)$ term in McCullagh and Nelder (1990), Eqn. 2.4, p. 28) to unity; this slightly simplified model encompasses the most important distributions for our purposes, including the Poisson, negative binomial, and Gaussian (with unit variance) distributions.

Let the terms $l_i^g, o_i^g, \mu_i^g, \eta_i^g, \psi_g, r_g, h_g$ and c_g be defined in an analogous way for the gRNA model, i.e. $l_i^g \equiv \langle \tilde{x}_i, \beta_g \rangle + o_i^g$, $r_g(\mu_i^g) \equiv l_i^g$, and $\eta_i^g = \left([\psi_g']^{-1} \circ r_g^{-1} \right)(l_i^g) \equiv h_g(l_i^g)$. The density f_g of g_i given the canonical parameter is

$$f_g(m_i; \eta_i^g) = \exp\left\{ g_i \eta_i^g - \psi_g(\eta_i^g) + c_g(g_i) \right\}.$$

Finally, the unobserved variable p_i is assumed to follow a Bernoulli distribution with mean $\pi \in (0, 1/2]$. Its marginal density f_p is given by

$$f_p(p_i) = \pi^{p_i}(1 - \pi)^{1 - p_i}.$$

The unknown parameters in the model are $\theta = [\beta_m, \beta_g, \pi]^T \in \mathbb{R}^{2d+1}$.

Notation. We briefly introduce notation that we will use throughout. For $j \in \{0, 1\}$, let $\tilde{x}_i(j) \equiv [1, j, z_i]^T$ denote the value of $\tilde{x}_i$ that results from setting

p_i to j. Next, let $l_i^m(j)$, $\eta_i^m(j)$, and $\mu_i^m(j)$ be the values of l_i^m, η_i^m, and μ_i^m, respectively, that result from setting p_i to j, i.e.,

$$l_i^m(j) \equiv \langle \tilde{x}_i(j), \beta_m \rangle + o_i^m; \quad \eta_i^m(j) \equiv h_m(l_i^m(j)); \quad \mu_i^m(j) \equiv r_m^{-1}(l_i^m(j)).$$

Let the corresponding gRNA quantities $l_i^g(j)$, $\eta_i^g(j)$, and $\mu_i^g(j)$ be defined analogously. Next, let $X \in \mathbb{R}^{n \times (d-1)}$ be the observed design matrix, and let $\tilde{X} \in \mathbb{R}^{n \times d}$ be the augmented design matrix that results from concatenating the column of (unobserved) p_is to X, i.e.

$$X \equiv \begin{bmatrix} 1 & z_1 \\ \vdots & \vdots \\ 1 & z_n \end{bmatrix}; \quad \tilde{X} \equiv \begin{bmatrix} 1 & p_1 & z_1 \\ \vdots & \vdots & \vdots \\ 1 & p_n & z_n \end{bmatrix} = \begin{bmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_n^T \end{bmatrix}.$$

Furthermore, for $j \in \{0, 1\}$, let $\tilde{X}(j) \in \mathbb{R}^{n \times d}$ be the matrix that results from setting p_i to j for all $i \in \{1, \ldots, n\}$ in $\tilde{X}$, and let $[\tilde{X}(0)^T, \tilde{X}(1)^T]^T$ denote the $\mathbb{R}^{2n \times d}$ matrix that results from vertically concatenating $\tilde{X}(0)$ and $\tilde{X}(1)$. Furthermore, define $m := [m_1, \ldots, m_n]$, and let g, p, o^m, and o^g be defined analogously. Finally, let $[m, m]^T \in \mathbb{R}^{2n}$ be the vector that results from concatenating m to itself, i.e. $[m, m]^T \equiv [m_1, \ldots, m_n, m_1, \ldots, m_n]$, and let $[g, g]^T$, $[o^g, o^g]^T$, and $[o^m, o^m]^T$ be defined similarly.

Log likelihood and model properties. We state the log-likelihood and several properties of the GLM-EIV model. We conduct estimation and inference conditional on the library sizes and technical factors l_i^m, l_i^g, and z_i; therefore, we treat these quantities as fixed constants. We assume that the gene expression m_i and gRNA expression g_i are conditionally independent given the perturbation p_i. The model log-likelihood is

$$\mathcal{L}(\theta; m, g) = \sum_{i=1}^{n} \log \big[(1 - \pi) f_m(m_i; \eta_i^m(0)) f_g(g_i; \eta_i^g(0)) + \pi f_m(m_i; \eta_i^m(1)) f_g(g_i; \eta_i^g(1)) \big]. \quad (3.4)$$

We see from (3.4) that the GLM-EIV model is equivalent to a two-component mixture of *products* of GLM densities. Additionally, the GLM-EIV model is a generalization of the simple errors-in-variables model (when the predictor is binary); the latter is defined as follows:

$$y_i = \beta_0 + \beta_1 x_i^* + \epsilon_i; \quad x_i = x_i^* + \tau_i, \quad (3.5)$$

where, $x_i^* \sim \text{Bern}(\pi), \epsilon_i, \tau_i \sim N(0, 1)$, and ϵ_i,τ_i, and x_i^* are independent. GLM-EIV extends (3.5) in at least three directions: first, GLM-EIV allows y_i and

x_i to follow exponential family (i.e, not just Gaussian) distributions; second, GLM-EIV allows y_i and x_i to be related to x_i^* through arbitrary (i.e., not just linear) link functions; and finally, GLM-EIV allows confounders z_i to impact both x_i and y_i. Therefore, x_i and y_i can be conditionally *dependent* given x_i^*, enabling GLM-EIV to capture more complex dependence relationships between x_i and y_i than is possible in (3.5) or other standard measurement error models.

Estimation and inference, and computational infrastructure

$$T_i(1) = \mathbb{P}(p_i = 1 | M_i = m_i, G_i = g_i, \beta_m^{(\mathrm{t})}, \beta_g^{(\mathrm{t})}, \pi^{(\mathrm{t})}).$$

We can calculate this quantity by applying (i) Bayes rule, (ii) the conditional independence property of M_i and G_i, (iii) the density of M_i and G_i, and (iv) a log-sum-exp-type trick to ensure numerical stability. Next, we produce updated estimates $\pi^{(\mathrm{t}+1)}$, $\beta_g^{(\mathrm{t}+1)}$, and $\beta_m^{(\mathrm{t}+1)}$ of the parameters by maximizing the M step objective function. It turns out that maximizing this objective function is equivalent to setting $\pi^{(\mathrm{t}+1)}$ to the mean of the current membership probabilities and setting $\beta_g^{(\mathrm{t}+1)}$ and $\beta_m^{(\mathrm{t}+1)}$ to the fitted coefficients of a GLM weighted by the current membership probabilities (Algorithm 1). We iterate through the E and M steps until the log likelihood (3.4) converges (Appendix B.2). Our EM algorithm is reminiscent of (but distinct from) that of Ibrahim (1990), who also applied weighted GLM solvers to carry out an M step of an EM algorithm.

After fitting the model, we perform inference on the estimated parameters. The easiest approach, given the complexity of the log likelihood, would be to run a bootstrap. This strategy, however, is prohibitively slow, as the data are large and the EM algorithm is iterative. Therefore, we derive an analytic formula for the asymptotic observed information matrix using Louis's Theorem (Louis (1982); Appendix B.2). Leveraging this analytic formula, we can calculate standard errors quickly, enabling us to perform inference in practice on real, large-scale data.

A downside of the the EM algorithm (Algorithm 1) is that it requires fitting many GLMs. Assuming that we run the algorithm 15 times using randomly-generated pilot estimates (to improve chances of convergence to the global maximum), and assuming that the algorithm iterates through E and M steps about 10 times per run, we must fit approximately 300 GLMs. (These numbers are based on exploratory applications of the method to real and simulated data.) We instead devised a strategy to produce a highly accurate pilot estimate of the true parameters, enabling us to run the algorithm once and converge upon the MLE within a few iterations. The strategy involves layering several statistical "tricks" on top of one another; details are deferred to Appendix B.4. Overall,

Algorithm 1: EM algorithm for GLM-EIV model.

Input Pilot estimates $\beta_m^{\text{curr}}, \beta_g^{\text{curr}}$, and π^{curr}; data m, g, o^m, o^g, and X; gene expression distribution f_m and link function r_m; gRNA expression distribution f_g and link function r_g.

while *Not converged* **do**

  for $i \in \{1, \ldots, n\}$ **do**

    // E step

    $T_i(1) \leftarrow \mathbb{P}\left(p_i = 1 | M_i = m_i, G_i = g_i, \beta_m^{\text{curr}}, \beta_g^{\text{curr}}, \pi^{\text{curr}}\right)$

    $T_i(0) \leftarrow 1 - T_i(1)$

  end

  $\pi^{\text{curr}} \leftarrow (1/n) \sum_{i=1}^{n} T_i(1)$ // M step

  $w \leftarrow [T_1(0), T_2(0), \ldots, T_n(0), T_1(1), T_2(1), \ldots, T_n(1)]^T$

  for $k \in \{g, m\}$ **do**

    Fit a GLM GLM_k with responses $[k, k]^T$, offsets $[o^k, o^k]^T$, weights w, design matrix $[\tilde{X}(0)^T, \tilde{X}(1)^T]^T$, distribution f_k, and link function r_k.

    Set β_k^{curr} to the estimated coefficients of GLM_k.

  end

  Compute log likelihood using β_m^{curr}, β_g^{curr}, and π^{curr}.

end

$\hat{\beta}_m \leftarrow \beta_m^{\text{curr}}$; $\hat{\beta}_g \leftarrow \beta_g^{\text{curr}}$; $\hat{\pi} \leftarrow \pi^{\text{curr}}$.

return $(\hat{\beta}_m, \hat{\beta}_g, \hat{\pi})$

the statistical accelerations reduce the number of GLMs that we must fit to < 10 in most cases.

Next, we developed a computational infrastructure to apply GLM-EIV to large-scale, single-cell CRISPR screen data. The infrastructure leverages `Nextflow`, a programming language that facilitates building data-intensive pipelines, and `ondisc`, an `R`/`C++` package that we developed (in a separate project) to facilitate large-scale computing on single-cell data. `Nextflow` and `ondisc` together enable the construction of highly portable single-cell pipelines: one can analyze data *out-of-memory* on a laptop or in a *distributed* fashion across hundreds of processors on a cloud (e.g., Microsoft Azure, Google Cloud) or high-performance cluster. Leveraging these technologies, we built a Docker-containerized pipeline for deploying GLM-EIV at-scale. The pipeline aggressively recycles computation when possible, saving a considerable amount of compute; see Appendix B.4.2 for details. Overall, the statistical accelerations and computational infrastructure make the deployment of GLM-EIV to

large-scale single-cell CRISPR screen feasible.

3.6 Simulation study

We conducted a simulation study to compare the empirical performance of GLM-EIV to that of the thresholding method. We generated data on $n = 150,000$ cells from the GLM-EIV model using realistic parameter values, setting the the target of inference β_1^m to $\log(0.25)$ and the probability of perturbation π to 0.02. $\beta_1^m = \log(0.25)$ represents a decrease in gene expression by a factor of 4, which is a fairly large effect size on the order of what we might observe for a positive control pair. We included "sequencing batch" (modeled as a Bernoulli-distributed variable) as a covariate and sequencing depth (modeled as a Poisson-distributed variable) as an offset. We varied the log-fold change in gRNA expression, β_1^g, over a grid on the interval $[\log(1), \log(4)]$; β_1^g controls problem difficulty, with higher values corresponding to easier problem settings. Finally, we generated the gene expression and gRNA count data from two response distributions: Poisson and negative binomial (size parameter fixed at $s = 20$ for the latter). For each parameter setting (defined by a β_1^g-distribution pair), we synthesized $n_{\text{sim}} = 500$ i.i.d. datasets. Section B.5 presents additional simulation results on Gaussian response distributions.

We applied three methods to the simulated data: "vanilla" GLM-EIV, accelerated GLM-EIV, and thresholded regression. We used the Bayes-optimal decision boundary for classification as the threshold for the thresholding method. We ran all methods on the negative binomial data twice: once treating the size parameter s as a known constant and once treating s as unknown. In the latter case we used the `glm.nb` function from the `MASS` package to estimate s before applying the methods (Ripley et al., 2013). We note that neither the thresholding method nor GLM-EIV account for the error in estimating s when computing coefficient standard errors. We display the results of the simulation study in Figure 3.3. Columns correspond to distributions (i.e., Poisson, NB with known s, and NB with unknown s), and rows correspond to performance metrics (i.e., bias, mean squared error, CI coverage rate (nominal rate 95%), CI width, and method execution time). The problem difficulty parameter β_1^g is plotted on the horizontal axis, and the methods are depicted in different colors (GLM-EIV masked by accelerated GLM-EIV in several panels).

First, we observed that GLM-EIV dominated thresholded regression on all statistical metrics: GLM-EIV exhibited lower bias (row 1) and mean squared error (row 2) than thresholded regression; additionally, GLM-EIV had superior confidence interval coverage (row 3) despite having produced generally narrower confidence intervals (row 4). Intuitively, GLM-EIV outperformed the threshold-

ing method because (i) GLM-EIV leveraged information from *both* modalities (rather than the gRNA modality alone) to assign perturbation identities to cells, and (ii) GLM-EIV produced soft rather than hard assignments, capturing the inherent uncertainty in whether a perturbation occurred. We additionally found that accelerated GLM-EIV performed as well as vanilla GLM-EIV on all statistical metrics (rows 1-4) despite having substantially lower computational cost (bottom row). In fact, the execution time of accelerated GLM-EIV was almost within an order of magnitude of that of the thresholding method (bottom row).

Interestingly, thresholded regression exhibited better confidence interval coverage under estimated s than under known s (row 3). Estimating s leads to slight inflation bias (i.e., overestimating the true effect size), whereas, as we showed previously, thresholding leads to attenuation bias (i.e., underestimating the true effect size). These phenomena partially canceled, yielding less biased estimates. GLM-EIV exhibited worse performance under unknown s than known s, likely due to poor s estimation. We note that GLM-EIV and the thresholding method in principle are compatible with *any* estimation procedure, including those based on more sophisticated techniques, such as regularization (Hafemeister and Satija, 2019). We defer rigorous investigation of the impact of different s estimation strategies on these methods to future work.

3.7 Data analysis

Leveraging our computational infrastructure, we applied GLM-EIV and the thresholding method to analyze the entire Gasperini and Xie datasets. We report only the most important aspects of the analysis and results in the main text; full details are available in Appendix B.6. We set the threshold in the thresholding method to the approximate Bayes-optimal decision boundary, as our theoretical analyses and simulation studies indicated that the Bayes-optimal decision boundary is a good choice for the threshold when the gRNA count distribution is well-separated. Operating under the assumption that the effect of the perturbation on gRNA expression is similar across pairs, we leveraged the fitted GLM-EIV models to approximate the Bayes boundary in the following way: we (i) sampled several hundred gene-perturbation pairs, (ii) extracted the fitted values $\hat{\beta}_g$ and $\hat{\pi}$ from the GLM-EIV models fitted to these pairs, (iii) computed the median $\overline{\hat{\beta}_g}$ and $\overline{\hat{\pi}}$ across the $\hat{\beta}_g$s and $\hat{\pi}$s, and (iv) used $\overline{\hat{\beta}_g}$ and $\overline{\hat{\pi}}$ to estimate a dataset-wide Bayes-optimal decision boundary. We repeated this procedure on both datasets, yielding a threshold of 3 for Gasperini and 7 for Xie.

We compared GLM-EIV to thresholded regression on the real data, focusing specifically on the negative control pairs (i.e., gene-perturbation pairs for which the ground truth fold change is known to be 1; Appendix B.6). We found that GLM-EIV and the thresholding method produced similar results (Figure 3.4a-b): estimates, CI coverage rates, and CI widths were concordant. CI coverage rates, which ranged from 87.7%-91.2%, were slightly below the nominal rate of 95%, likely due to model misspecification. The estimated effect of the perturbation on gene expression $\exp(\hat{\beta}_1^g)$ was unexpectedly large: the 95% CI for this parameter was $[4306, 5186]$ and $[300, 316]$ on the Gasperini and Xie data, respectively. We reasoned that the datasets lay in an "easy" region of the parameter space, making thresholding a tenable strategy (provided the threshold is selected well). However, this was not obvious *a priori* and may not be the case for other datasets. We note that GLM-EIV produced outlier estimates (defined as estimated fold change < 0.75 or > 1.25) on a small ($< 2.5\%$ on Gasperini, $< 0.05\%$ on Xie) number of pairs consisting of a handful of genes, likely due to non-global EM convergence. These outliers are not plotted in Figures 3.4a-b but were used to compute the CI coverage reported in the inset tables.

To evaluate performance of GLM-EIV versus thresholding in more challenging settings, we increased the difficulty of the perturbation assignment problem by generating partially-synthetic datasets. First, for a given pair, we sampled gRNA counts directly from the fitted GLM-EIV model. Next, to simulate elevated background contamination, we sampled gRNA counts from a slightly modified version of the fitted model in which we increased the mean gRNA expression of *unperturbed* cells while holding constant the mean gRNA expression of *perturbed* cells. We defined a parameter called "excess background contamination" (normed to take values in $[0, 1]$) to quantify the relative distance between the unperturbed and perturbed gRNA count distributions. We held fixed the real-data gene expressions, library sizes, covariates, and fitted perturbation probabilities in all settings.

We generated partially-synthetic data in the above manner for each of the 322 positive control pairs in the Gasperini dataset, varying excess background contamination over the interval $[0, 0.4]$. We then applied GLM-EIV and the thresholding method to analyze the data. We present results on two example pairs (the pair containing gene *LRIF1* and the pair containing gene *NDUFA2*) in Figures 3.4c-d. We observed that the estimate produced by the methods on the raw data (depicted as a horizontal black line) coincided almost exactly with the estimate produced by the methods on the partially-synthetic data generated by setting excess background contamination to zero (This result replicated across nearly all pairs; average relative difference 0.003.) We additionally observed that

as excess background contamination increased, the performance of thresholded regression degraded considerably while that of GLM-EIV remained stable.

We generalized the above analysis to the entire set of positive control pairs. First, for each pair we computed the "relative estimate change" (REC) as a function of excess background contamination, defined as the relative difference between the estimate at a given level of excess contamination and zero excess contamination (Figure 3.4d). Next, we computed the median REC across all positive control pairs (Figure 3.4e; upper and lower bands indicate the pointwise interquartile range of the REC). As excess background contamination increased, thresholded regression exhibited severe attenuation bias (as reflected by large median REC values); GLM-EIV, by contrast, remained mostly stable. Finally, letting $\hat{\beta}_1^m$ denote the estimate obtained on the raw data, we computed the CI coverage of $\hat{\beta}_1^m$ as a function of excess contamination. Under the assumption that $\hat{\beta}_1^m$ is close to the true parameter β_1^m, the CI coverage of the former is similar to that of the latter. We computed the CI coverage of $\hat{\beta}_1^m$ by calculating each individual pair's coverage of $\hat{\beta}_1^m$ (across the Monte Carlo replicates) and then averaging this quantity across all pairs. GLM-EIV exhibited significantly higher CI coverage than thresholded regression as the data became increasingly contaminated (Figure 3.4f; bands indicate 95% pointwise CIs). Coverage rates were slightly above the nominal level of 95% in some settings because we covered an *estimate* of β_1^m rather than β_1^m itself, leading to mild "overfitting." Nonetheless, this experiment was meaningful to assess the stability of both methods to elevated background contamination.

3.8 Discussion

In this Chapter we introduced GLM-EIV ("GLM-based errors in variables"), a new model and associated method for single-cell CRISPR screen analysis. GLM-EIV extends the classical errors-in-variables model to responses and noisy predictors that are exponential family-distributed and potentially impacted by the same set of confounding variables. These extensions enable GLM-EIV to resolve novel analysis challenges posed by single-cell CRISPR screens. We demonstrated through simulation studies, real data analyses, and theory that GLM-EIV outperforms thresholded regression by a considerable margin in high background contamination settings. GLM-EIV intuitively achieves this performance gain by leveraging information from *both* modalities (rather than the gRNA modality alone) to assign perturbation identities to cells. On the other hand, in low background contamination settings, GLM-EIV and thresholded regression work best in conjunction, with GLM-EIV providing a statistically principled and empirically effective procedure for selecting the

threshold. GLM-EIV thereby neutralizes a tuning parameter that, until this point, has been selected using heuristic procedures, with little confidence that the choice is near optimal. Figure 3.5 summarizes how we anticipate GLM-EIV being used in practice.

To our knowledge this is the first single-cell CRISPR screen work oriented toward a statistical audience. We hope that this Chapter helps to introduce the broader statistics community to an emerging class of functional genomics assays that likely will exert a major impact on biological research in the coming years (Przybyla and Gilbert, 2021). Additionally, this is the first work to leverage the `ondisc-Nextflow`-HPC/cloud technology stack, a tightly-integrated, user-friendly, and powerful set of tools for large-scale single-cell analysis. We expect this technology stack to be of interest to other single-cell researchers.

We anticipate that GLM-EIV could be applied to other types of single-cell CRISPR screen and multimodal single-cell data. For example, GLM-EIV might be extended (with some effort) to "low-multiplicity of infection" screens (Schraivogel et al., 2020) in which each cell receives exactly one perturbation rather than dozens (as is the case in "high multiplicity screens," studied in this Chapter). We also could apply GLM-EIV to analyze multimodal single-cell chromatin accessibility assays. A question of interest in such experiments is whether chromatin state (i.e., closed or open) is associated with the expression of a gene or abundance of a protein (Mimitou et al., 2021). We do not directly observe the chromatin state of a cell; instead, we observe tagged DNA fragments that serve as count-based proxies for whether a given region of chromatin is open or closed. GLM-EIV might be applied in such experiments to aid in the selection of thresholds or to analyze whole datasets.

The closest parallels to GLM-EIV in the statistical methodology literature are Grün and Leisch (2008) and Ibrahim (1990). Grün and Leisch derived a method for estimation and inference in a k-component mixture of GLMs. While we prefer to view GLM-EIV as a generalized errors-in-variables method, the GLM-EIV model is equivalent to a two-component mixture of products of GLM densities. Ibrahim proposed a procedure for fitting GLMs in the presence of missing-at-random covariates. Our method, by contrast, involves fitting two conditionally independent GLMs in the presence of a totally latent covariate. Thus, while Ibrahim and Grün & Leisch are helpful references, our estimation and inference tasks are more complex than theirs. Next, Aigner (1973) and Savoca (2000) proposed measurement error models that consist of unobserved *binary* rather than *continuous* predictors; the latter are more commonly used in measurement error models. GLM-EIV likewise consists of a latent binary predictor, but unlike Aigner and Savoca, GLM-EIV handles a much broader class of exponential family-generated data. Finally, GLM-EIV accounts for

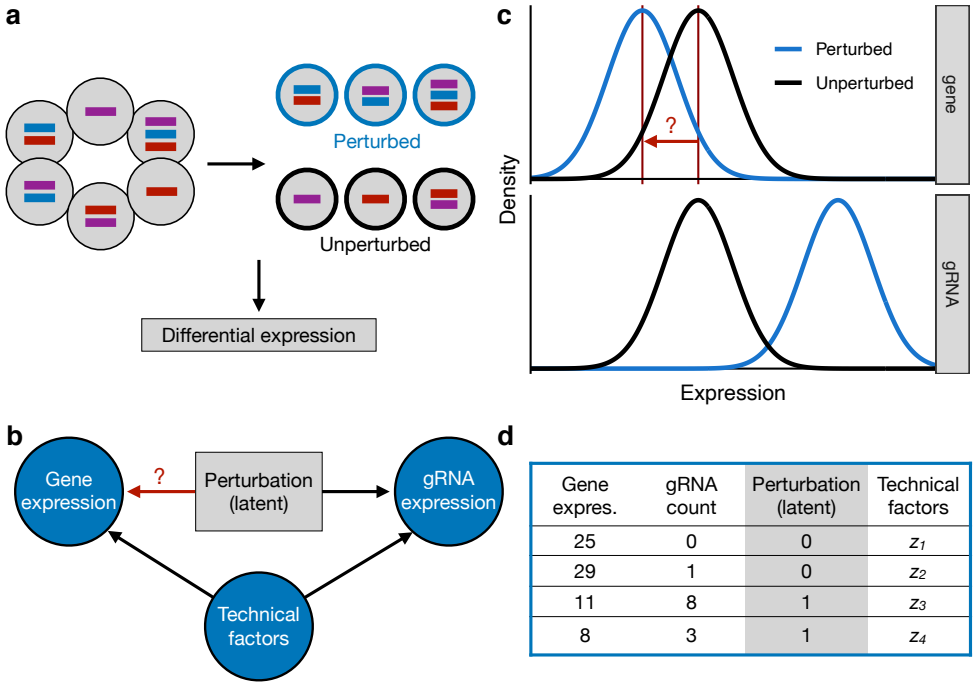
a common source of measurement error between the predictor and response, a property not shared by classical measurement error models (Carroll et al., 2006). Additional related work is relayed in Appendix B.7.

GLM-EIV might be applied to areas beyond genomics, such as psychology. Many psychological constructs (e.g., presence or absence of a social media addiction) are latent and can be assessed only through an imperfect proxy (e.g., the number of times one has checked social media). Researchers might use GLM-EIV to regress an outcome variable (e.g., self-reported well-being) onto the latent construct via the imperfect proxy, potentially resolving challenges related to attenuation bias and threshold selection. Applications to psychology and other areas are a topic of future investigation.

Chapter acknowledgements

We thank Eric Tchetgen Tchetgen for helpful conversations, Xuran Wang for helping to process the Xie dataset, and Songcheng Dai for helping to deploy the GLM-EIV pipeline on Azure. This Chapter used the Extreme Science and Engineering Discovery Environment (XSEDE; NSF grant ACI-1548562) and the Bridges-2 system (NSF grant ACI-1928147) at the Pittsburgh Supercomputing Center. This Chapter is funded by National Institute of Mental Health (NIMH) grant R01MH123184 and NSF grant DMS-2113072.

3.9 Figures



Gene expres.	gRNA count	Perturbation (latent)	Technical factors
25	0	0	z_1
29	1	0	z_2
11	8	1	z_3
8	3	1	z_4

Figure 3.1: **Experimental design and analysis challenges**: **a,** Experimental design. For a given perturbation (e.g., the perturbation indicated in blue), we partition the cells into two groups: perturbed and unperturbed. Next, for a given gene, we conduct a differential expression analysis across the two groups, yielding an estimate of the impact of the given perturbation on the given gene. **b,** DAG representing all variables in the system. The perturbation (latent) impacts both gene expression and gRNA expression; technical factors act as confounders, also impacting gene and gRNA expression. The target of estimation is the effect of the perturbation on gene expression. **c,** Schematic illustrating the "background read" phenomenon. Due to errors in the sequencing and alignment processes, unperturbed cells exhibit a nonzero gRNA count distribution (bottom). The target of estimation is the change in mean gene expression in response to the perturbation (top). **d**, Example data on four cells for a given perturbation-gene pair. Note that (i) the perturbation is unobserved, and (ii) the gene and gRNA data are discrete counts.

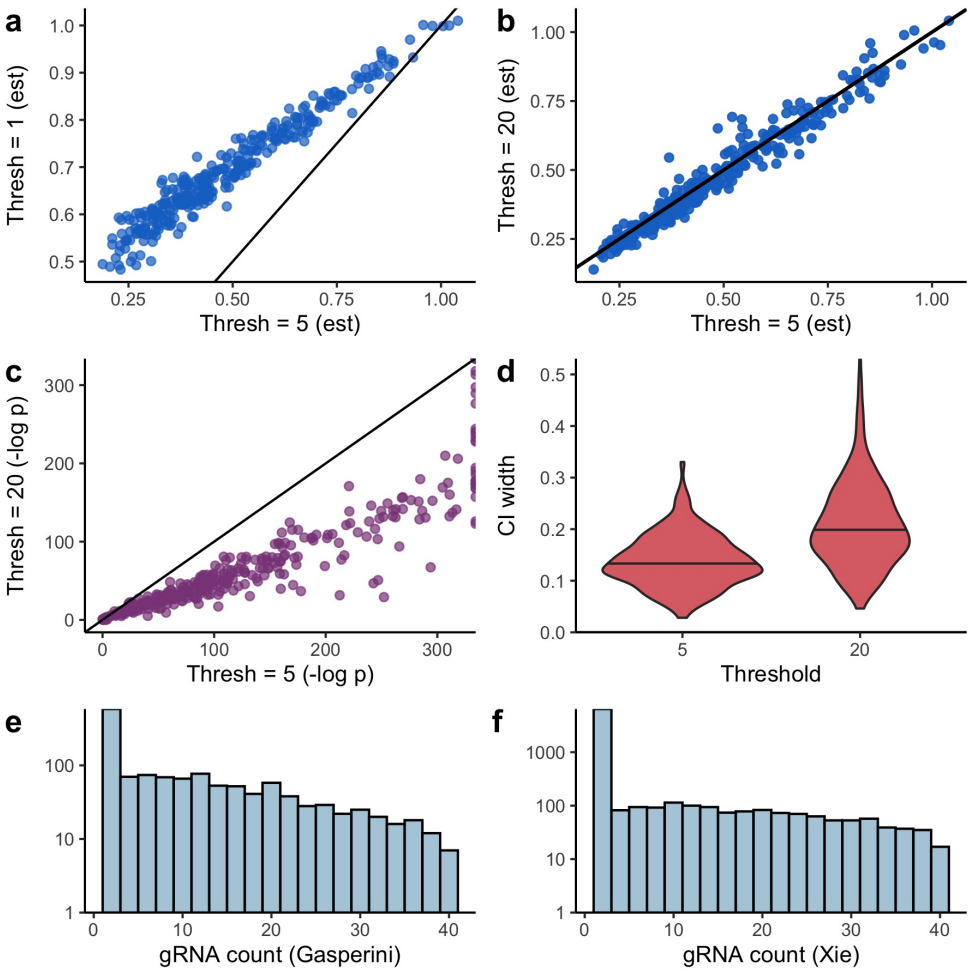


Figure 3.2: **Empirical challenges of thresholded regression. a-b,** Fold change estimates produced by threshold = 1 versus threshold = 5 (a) and threshold = 20 versus threshold = 5 (b). The selected threshold substantially impacts the results. **c-d,** p-values (c) and CI widths (d) produced by threshold = 20 versus threshold = 5. The latter threshold yields more confident estimates. **e-f**, Empirical distribution of randomly-selected gRNA from Gasperini (e) and Xie (f) data (0 counts not shown). The gRNA data do not appear to imply an obvious threshold selection strategy.

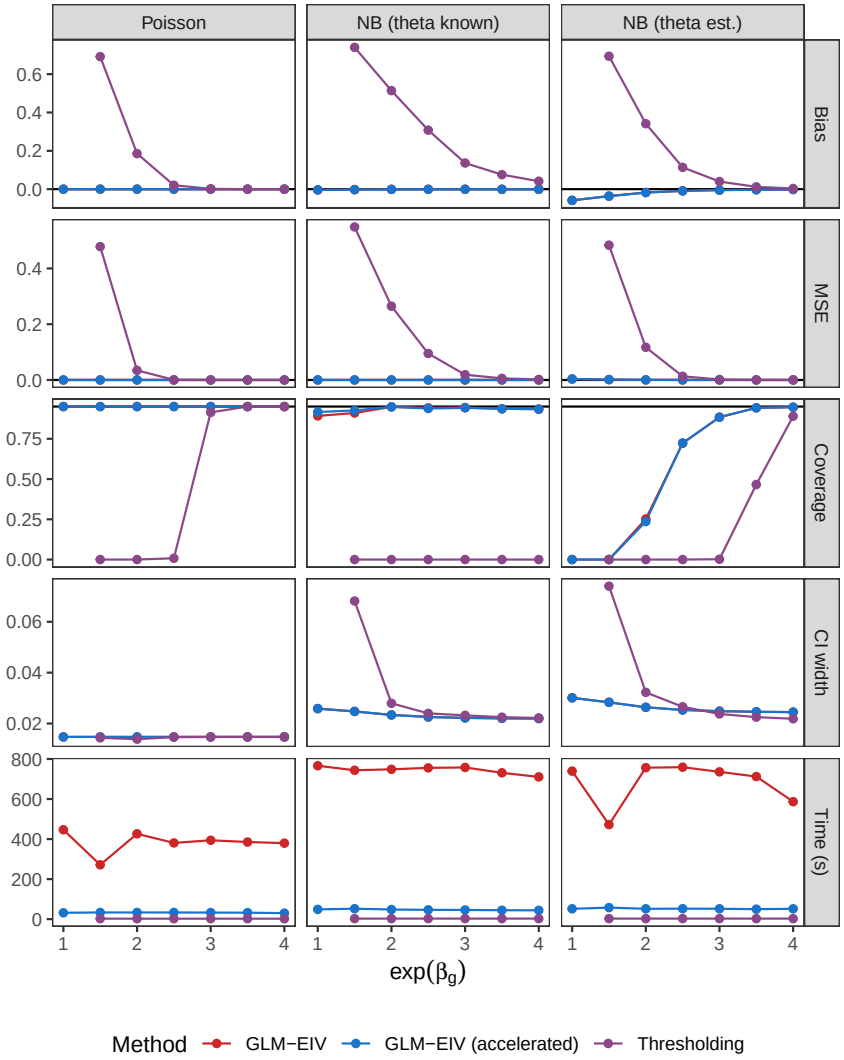


Figure 3.3: **Simulation study**. Columns correspond to distributions (Poisson, NB with known s, NB with estimated s), and rows correspond to metrics (bias, MSE, coverage, CI width, and time). Methods are shown in different colors; GLM-EIV (red) is masked by accelerated GLM-EIV (blue) in several panels. GLM-EIV demonstrated superior statistical performance to the thresholding method on all metrics (rows 1-4). Accelerated GLM-EIV had substantially lower computational cost than "vanilla" GLM-EIV (bottom row), despite demonstrating identical statistical performance (rows 1-4).

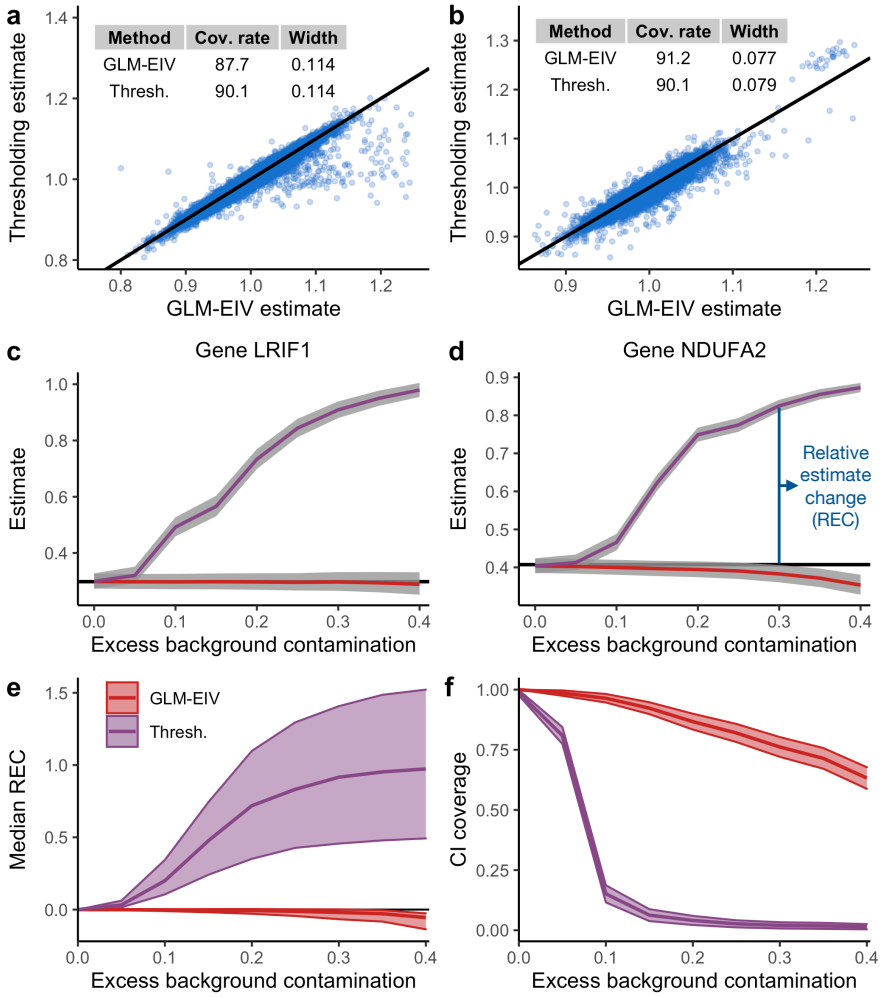


Figure 3.4: **Data analysis**. **a-b**, Estimates for fold change produced by GLM-EIV and thresholded regression on Gasperini (**a**) and Xie (**b**) negative control pairs. **c-d**, Estimates produced by GLM-EIV and thresholded regression on two positive control pairs – *LRIF1* (**a**) and *NDUFA2* (**b**) – plotted as a function of excess background contamination. Grey bands, 95% CIs for the target of inference outputted by the methods. **e-f**, Median relative estimate change (REC; **e**) and confidence interval coverage rate (**f**) across *all* 322 positive control pairs, plotted as a function of excess background contamination. Panels (**c-f**) together illustrate that GLM-EIV demonstrated greater stability than thresholded regression as background contamination increased.

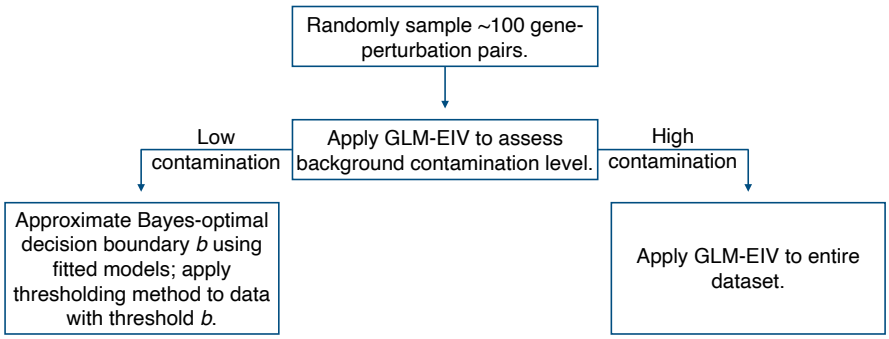


Figure 3.5: **Use of GLM-EIV in practice**. The decision tree above illustrates how we anticipate GLM-EIV could be used in practice. First, apply GLM-EIV to a set of randomly-sampled gene-perturbation pairs to assess background contamination level (positive control pairs work best for this purpose). If GLM-EIV indicates that background contamination is high (e.g., $\exp(\beta_1^g) \lessapprox 10$), apply GLM-EIV to analyze the entire dataset; otherwise, approximate the Bayes-optimal decision boundary using the fitted GLM-EIV models. Next, apply a thresholding method (e.g., SCEPTRE or thresholded negative binomial regression) to analyze the data, setting the threshold to the estimated Bayes-optimal decision boundary.

Four

Robust association testing at low multiplicity-of-infection

4.1 Introduction

Chapters 2 and 3 studied association testing and estimation, respectively, for high multiplicity-of-infection (MOI) single-cell CRISPR screens. In this Chapter we shift our focus to low-MOI single-cell CRIPSR screens. The most fundamental statistical task involved in the analysis of low-MOI single-cell CRISPR screen data is to test for association between a perturbation and a univariate, count-based molecular phenotype, like the expression of a gene or protein. In our previous work on high-MOI single-cell CRISPR screen analysis in Chapter 2, we discovered that existing methods for association testing are prone to an excess of false positive hits. In that Chapter we proposed SCEPTRE, a well-calibrated method for association testing on high-MOI data.

Since low-MOI screens currently outnumber high-MOI screens, the low-MOI association testing problem is even more pressing. A variety of methods has been deployed for association testing in low-MOI (Dixit et al., 2016; Yang et al., 2020; Schraivogel et al., 2020; Replogle et al., 2020; Papalexi et al., 2021; Frangieh et al., 2021; Wang, 2021; Liscovitch-Brauer et al., 2021). However, there is no consensus as to which of these methods represents the "state of the art;" these methods have not undergone rigorous statistical validation and comparison; and in fact there is no commonly accepted framework for quantifying the statistical validity of single-cell CRISPR screen association testing methods. Resolving these fundamental issues is essential to ensuring the reliability of biological conclusions made on the basis of single-cell CRISPR screen experiments.

This Chapter is based on Barry et al. (2023), which is joint work with Kaishu Mason, Kathyrn Roeder, and Eugene Katsevich. Timothy Barry and Eugene Katsevich both contributed to the text and figures.

We aimed to address the aforementioned challenges by making three contributions. First, we developed a simple framework for evaluating the calibration of association testing methods for single-cell CRISPR screens. We then leveraged this framework to conduct the first-ever comprehensive benchmarking study of association methods on low-MOI data, applying six leading methods to analyze six diverse datasets. We found that all existing methods exhibit varying degrees of miscalibration, indicating that results obtained using these methods may be contaminated by excess false positive discoveries. Second, to shed light on why existing methods might demonstrate miscalibration, we conducted an in-depth empirical investigation of the data, uncovering three core analysis challenges: confounding, model misspecification, and data sparsity. No existing method addresses all of these analysis challenges, explaining their collective lack of calibration. Finally, we developed SCEPTRE (low-MOI), a substantial extension of the original SCEPTRE (Chapter 2) tailored to the analysis of low-MOI single-cell CRISPR screens. SCEPTRE (low-MOI) is based on the novel and statistically principled technique of permuting negative binomial score statistics. (We often will refer to the low-MOI version of SCEPTRE simply as "SCEPTRE" in this Chapter for the sake of brevity). SCEPTRE addresses all three core analysis challenges both in theory and in practice, demonstrating markedly improved calibration and power relative to existing methods across datasets.

4.2 Results

A survey of leading analysis methods

Association testing on low-MOI single-cell CRISPR screen data is a variation on the classical single-cell differential expression testing problem (Figure 4.1a). To test for association between a given targeting CRISPR perturbation and gene, one first divides the cells into two groups: those that received the targeting perturbation, and those that received a non-targeting (NT) perturbation. (All other cells typically are ignored.) One then tests for differential expression of the given gene across these two groups of cells, yielding a fold change estimate and p-value. One repeats this procedure for a (typically) large, preselected set of perturbation-gene pairs. Finally, one computes the discovery set by subjecting the tested pairs to a multiplicity correction procedure (e.g., Benjamini-Hochberg).

We use the term "control group" to refer to the cells against which the cells that received the targeting perturbation are compared. As indicated above, the control group typically is the set of cells that received an NT perturbation (i.e., the "NT cells"). Certain single-cell CRISPR screen methods, however,

take as their control group the set of cells that did *not* receive the targeting perturbation (i.e., the "complement set"). In low-MOI screens the NT cells generally constitute a more natural control group than the complement set, as we seek to compare the effect of the targeting perturbation to that of a "null" perturbation rather than to the average of the effects of all other perturbations introduced in the pooled screen. In high-MOI screens, however, the complement set is the only choice because few (if any) cells receive only NT perturbations.

We surveyed recent analyses of single-cell CRISPR screen data and identified five methods commonly in use: the default Seurat `FindMarkers()` function based on the Wilcoxon test (Seurat-Wilcox; Butler et al. 2018), MIMOSCA (Dixit et al., 2016), a t-test on the library-size-normalized expressions (Liscovitch-Brauer et al., 2021), MAST (Finak et al., 2015), and a Kolmogorov-Smirnov (KS) test on the library-size-normalized expressions (Adamson et al., 2016). We also considered applying `FindMarkers()` with negative binomial (NB) regression rather than the Wilcoxon test (Seurat-NB). These methods vary along several dimensions (Table 4.1; Existing methods details), including their testing paradigm (two-sample test versus regression-based test), how they normalize the data, whether they make parametric assumptions, and whether they use the NT cells or the complement set as their control group. Most of these methods are popular single-cell differential expression procedures that have been adapted to single-cell CRISPR screen data.

Comprehensive benchmarking study of leading analysis methods

We sought to assess whether these methods are correctly calibrated (i.e., whether they yield uniformly distributed p-values under the null hypothesis of no association between the perturbation and gene). Methods that are not correctly calibrated can produce discovery sets that are contaminated by excess false positives or false negatives. Unfortunately, there does not exist a standard protocol for assessing the calibration of single-cell CRISPR screen association methods. The closest existing analysis (Schraivogel et al., 2020) proceeds by applying methods to analyze gene-perturbations pairs for perturbations with known targets. Any pair where the gene is not the known target of the perturbation is considered null. As acknowledged by the original authors, this approach underestimates precision because downstream effects of perturbations are not taken into account.

To help fill this methodological gap, we designed a simple procedure to check the calibration of a single-cell CRISPR screen association method (Figure 4.1b). We constructed a set of "null" or "negative control" perturbation-gene pairs by pairing each NT gRNA to each gene. We then deployed a given method to analyze these null pairs. (For methods that use the NT cells as their

Method	Paradigm	Parametric assumption	Null distribution	Normalization/ Adjustments	Control group
Seurat-Wilcox Mimitou et al. (2019); Papalexi et al. (2021); Wessels et al. (2022)	Two-sample test	No	Asymptotic	Library size	NT cells
MIMOSCA Dixit et al. (2016); Genga et al. (2019); Jin et al. (2020); Lalli et al. (2020); Frangieh et al. (2021); Ursu et al. (2022)	Regression-based	No	Permutation	Library size, other covariates	Complement set
t-test Liscovitch-Brauer et al. (2021)	Two-sample test	Yes	Asymptotic	Library size	NT cells
MAST Schraivogel et al. (2020)	Regression-based	Yes	Asymptotic	Library size, expressed genes	NT cells
KS test Adamson et al. (2016); Replogle et al. (2020, 2022)	Two-sample test	No	Asymptotic	Library size, batch	NT cells
Seurat-NB (single-cell DE)	Two-sample test	Yes	Asymptotic	Library size	NT cells

Table 4.1: **A summary of low-MOI single-cell CRISPR screen DE methods currently in use**. The applications of each method to single-cell CRISPR screens are cited below the method name. The methods vary along several key axes, including the use (or lack thereof) of parametric assumptions, the construction of the null distribution, the variables adjusted for, and the control group. NT, non-targeting.

control group — the majority of methods — this check consists of comparing cells containing a given NT gRNA to cells containing *any other* NT gRNA.) The output of this check is a set of $N_{\text{gene}} \cdot N_{\text{NT}}$ null p-values, where N_{gene} is the number of genes and N_{NT} is the number of NT gRNAs. Since the

null perturbation-gene pairs are devoid of signal, a well-calibrated association method should output uniformly distributed p-values on these pairs. Deviations from uniformity — and thus miscalibration of the method — can be detected by inspecting a QQ plot of the p-values. Quantitatively, the number of null pairs passing a Bonferroni correction measures the extent of the miscalibration; well-calibrated methods should have roughly zero such pairs.

We note that there are two uses of the proposed calibration check procedure. The first use is for the goal of benchmarking existing analysis methods to identify which, if any, are suitable for broad application (the primary goal of this section). The second use is for the goal of testing whether a *given* method is well-calibrated on a *given* dataset. These two goals are distinct; a method may not be broadly well-calibrated but may perform adequately on a given dataset. In the context of the second goal, we recommend applying a modified calibration check where the set of negative control perturbation-gene pairs is matched to the set of pairs under consideration based on several criteria (described later).

We employed the above framework to systematically benchmark the performance of the existing methods, implementing each as faithfully as possible in a publicly available R package `lowmoi` (`github.com/katsevich-lab/lowmoi`). We applied the calibration check procedure using six single-cell CRISPR screen datasets, five real and one simulated (Tables C.1-C.2). The five real datasets came from three recent papers: Frangieh et al. (2021) (three datasets), Papalexi et al. (2021) (one dataset), and Schraivogel et al. (2020) (one dataset). The data were diverse, varying along the axes of CRISPR modality (CRISPRko or CRISPRi), technology platform (perturb-CITE seq, ECCITE-seq, or targeted perturb-seq), cell type (TIL, K562, or THP1), and genomic element targeted (enhancers or gene TSS). Notably, the Papalexi data were multimodal, containing both gene and protein expression measurements. For simplicity we analyzed the gene and protein modalities separately throughout.

Surprisingly, the results of our analyses (Figures 4.1c-f, C.1, C.2, C.3) revealed substantial miscalibration for many dataset-method pairs. On the Papalexi data, for example, the KS test produced inflated p-values, yielding over 9,000 false Bonferroni discoveries. MAST was similarly inflated on the Frangieh IFN-γ data, falsely rejecting nearly 2,000 null perturbation-gene pairs. MIMOSCA, meanwhile, exhibited noticeably non-uniform behavior on both datasets, outputting p-values strictly less than 0.26 across all pairs. Overall, the two best methods appeared to be Seurat-Wilcox and Seurat-NB, although these two methods still demonstrated clear signs of miscalibration. We observed that the calibration quality of a given method could vary significantly across datasets; this is explained by the fact that different datasets pose different challenges.

Nevertheless, we concluded that none of the methods was adequately calibrated across all datasets tested, suggesting that existing methods may not be suitable for broad application.

Systematic identification of core analysis challenges

We conducted an extensive empirical investigation of the data to search for possible sources of miscalibration, uncovering three core analysis challenges: sparsity, confounding, and model misspecification. No method that we examined addressed more than one of these analysis challenges (Table C.3), explaining their collective lack of calibration.

Single-cell CRISPR screen data typically are sparse, both in terms of gene expression and perturbation presence. Many genes have nonzero expression in only a small fraction of cells. On the other hand, due to the pooling of a large number of perturbations in a single experiment, the perturbation presence data are also sparse: most perturbations are present in only a small fraction of cells. The latter sparsity distinguishes single-cell CRISPR screens from other single-cell applications and is particularly pronounced in low-MOI. To summarize both sources of sparsity in a single number, we defined the "effective sample size" for a given perturbation-gene pair as the number of cells containing both the perturbation and nonzero gene expression.

We found that effective sample size had a substantial effect on the calibration of many methods under consideration (Figure C.4), especially those based on asymptotic approximations, such as Seurat-Wilcox. Asymptotic approximations tend to break down when the effective sample size is too low. For example, we compared the exact null distribution of the Wilcoxon test statistic (obtained via permutations) to the asymptotic Gaussian distribution used by Seurat-Wilcox; the latter is a computationally tractable approximation to the former in large samples. The Gaussian distribution provided a reasonable approximation to the exact null distribution for some pairs (Figure 4.2a, left) but not others (4.2a, right). Furthermore, as the effective sample size decreased and the Gaussian approximation degraded in accuracy, the p-value obtained via the Gaussian approximation likewise degraded in accuracy (Figure 4.2b). Finally, stratifying the Seurat-Wilcox null p-values by effective sample size on the Frangieh IFN-γ data revealed that pairs with small effective sample sizes yielded more inflated p-values than pairs with large effective sample sizes (Figure 4.2c).

Second, technical factors, such as biological replicate, batch, and library size, impact not only a cell's expression level, but also its probability of receiving a perturbation, thereby creating a confounding effect that, if not accounted for, can lead to spurious associations (Chapter 2; Figure 4.2d, Figure C.5). All existing methods adjust for library size, but few adjust for other technical

factors (Table 4.1). To assess the utility of adjusting for technical factors beyond library size, we applied negative binomial (NB) regression — both with and without biological replicate included as a covariate — to the Papalexi negative control data (Figure 4.2e). The variant of NB regression with biological replicate, though not perfectly calibrated, outperformed its counterpart without biological replicate. Methods not adjusting for biological replicate on the Papalexi data (such as Seurat-Wilcox) exhibited *worse* calibration for large effective sample sizes (Figure C.4), where there is more power to detect the spurious confounding-driven associations.

Third, methods that rely upon parametric models for the gene expression distribution, such as NB regression and MAST, can yield miscalibrated p-values when those models are misspecified (Li et al., 2022). To assess this effect, we monitored p-value calibration of the NB regression method on the Frangieh IFN-γ data while gradually increasing the effective sample size (Figure 4.2f). We found that the calibration quality improved until a point before plateauing; even for large effective sample sizes, noticeable miscalibration remained. (The non-parametric Seurat-Wilcox method, by contrast, attained good calibration for large effective sample sizes on this dataset.) This pattern was consistent with poor fit of the NB regression model, potentially due to inadequate estimation of the NB size parameter.

SCEPTRE (low-MOI) addresses the analysis challenges

We next developed SCEPTRE (low-MOI), a method for robust single-cell CRISPR screen association testing on low-MOI data (Figure 4.3a). For a given targeting perturbation-gene pair, SCEPTRE first regresses the vector of gene expressions onto the vector of perturbation indicators and matrix of technical factors via an NB GLM. (A given entry of the perturbation indicator vector is set to "1" if the corresponding cell contains a targeting perturbation and "0" if it contains a non-targeting perturbation.) SCEPTRE then computes the z-score z_{obs} for a test of the null hypothesis that the the coefficient corresponding to the perturbation indicator in the fitted GLM is zero. Next, SCEPTRE permutes the perturbation indicator vector B times (while holding fixed the gene expression vector and technical factor matrix) and recomputes a z-score for each of the permuted indicator vectors, yielding B "null" z-scores. Finally, SCEPTRE fits a smooth (skew-normal) density to the histogram of null z-scores and computes a p-value by evaluating the tail probability of the fitted density based on the original test statistic z_{obs}.

SCEPTRE possesses several appealing theoretical and computational properties. Theoretically, SCEPTRE is robust to the calibration threats of sparsity, confounding, and model misspecification. A key observation is that the techni-

cal factors (e.g., biological replicate) may or may not exert a confounding effect on the perturbation indicator and gene expression (Figure 4.3b). If confounding is absent for a given perturbation-gene pair, then SCEPTRE is valid regardless of misspecification of the NB model or the level of sparsity. On the other hand, if confounding is present, then SCEPTRE retains validity if the NB model is correctly specified and the problem is not too sparse (Figure 4.3c). (In fact, in the latter case, the NB model need only be specified correctly up to the dispersion parameter, sidestepping the difficult problem of NB dispersion parameter estimation; Love et al. (2014); Lause et al. (2021)) In this sense SCEPTRE is the only method that addresses all three core analysis challenges (Table C.3). We explored the above key robustness property of SCEPTRE in a brief simulation experiment (Figure C.6).

SCEPTRE also is performant, capable of analyzing hundreds of perturbation-gene pairs per second. We attained this efficiency by implementing several computational accelerations. First, we elected to use a *score* test (as opposed to a more standard *Wald* or *likelihood ratio* test) to compute the NB z-scores; the score test enabled us to fit a single NB GLM per perturbation-gene pair and share this fitted GLM across all permuted perturbation indicator vectors. Second, we derived a new algorithm for computing GLM score tests; this new algorithm is hundreds of times faster than the classical algorithm when the perturbation indicator vector is sparse, as is the case in single-cell CRISPR screen analysis. Finally, we developed several strategies for recycling computation across distinct perturbation-gene pairs. We note that SCEPTRE (low-MOI) is inspired by, but distinct in several ways from SCEPTRE (high-MOI; Chapter 2). We clarify similarities and differences between these two methods in Comparison of SCEPTRE (low-MOI) and SCEPTRE (high-MOI).

Application of SCEPTRE to negative and positive control data

We added SCEPTRE to the calibration benchmarking analysis presented before. An inspection of the QQ plots revealed that SCEPTRE markedly improved on the calibration of the two best existing methods, namely Seurat-Wilcox and Seurat-NB (Figure 4.4a-b). For example, on the Frangieh IFN-γ data, SCEPTRE made one Bonferroni rejection and yielded p-values that lay mostly within the gray 95% confidence band. The Seurat methods, by contrast, made fifteen false rejections each and produced p-values that fell considerably outside the confidence band. Next, we tabulated the number of Bonferroni-significant false positives for each dataset-method pair (Figure 4.4c; smaller values are better). SCEPTRE generally made the fewest number of false discoveries among all methods. On average over datasets, SCEPTRE made only 0.7 false discoveries, a roughly tenfold improvement over the Seurat methods.

Next, we assessed the power of the methods by applying them to positive control data. We constructed positive control pairs for each dataset by coupling perturbations targeting TSSs or known enhancers to the genes (or proteins) regulated by these elements. We examined the number of "highly significant" discoveries — operationally defined as rejections made at level $\alpha = 10^{-5}$ — made by each method on each dataset (Figure 4.4c; larger values are better). Methods that exhibited extreme miscalibration on a given dataset (defined as > 50 Bonferroni rejections on the negative control pairs of that dataset) were excluded from the positive control analysis, as assessing the power of such methods is challenging. We found that SCEPTRE matched or outperformed the other methods with respect to power on nearly every dataset (while at the same time achieving better calibration on negative control data).

Pairwise quality control

Quality control (QC) — the removal of low-quality genes and cells — is a key step in the analysis of single-cell data. In the context of single-cell CRISPR screens, it is useful not only only to remove low quality genes, perturbations, and cells but also low-quality perturbation-gene pairs. We term this latter type of QC "pairwise QC." As discussed previously, effective sample size — the number of cells containing both the perturbation and nonzero gene expression — affects the calibration of several methods considered. It also affects power, as small effective sample sizes yield low power and therefore needlessly increase the multiplicity burden. We found that SCEPTRE rarely rejected positive control pairs with an effective sample size below seven (Figure C.7); moreover, SCEPTRE maintained calibration for negative control pairs with an effective sample size of seven and above. For this reason our pairwise QC strategy consisted of filtering for pairs with an effective sample size of seven or greater. We applied this pairwise QC throughout.

Application of SCEPTRE for discovery analyses

The standard workflow involved in applying SCEPTRE to analyze a new single-cell CRISPR screen dataset consists of three main steps. First, the user prepares the data to pass to SCEPTRE and defines the "discovery set," which is the set of perturbation-gene pairs that the user seeks to test for association. (A reasonable default choice is the set of all possible pairs.) Second, the user runs the "calibration check" to verify that SCEPTRE is adequately calibrated on the dataset under analysis. The calibration check involves applying SCEPTRE to analyze a set of automatically-constructed negative control pairs. These negative control pairs are "matched" to the discovery pairs in several respects. For example, the negative control pairs and discovery pairs are subjected to

the exact same pairwise QC, and the number of negative control pairs is set equal to the number of discovery pairs. If the calibration check fails, the user can take steps to improve calibration, such as adding covariates or increasing the pairwise QC threshold. After verifying adequate calibration, the user runs the "discovery analysis," which entails applying SCEPTRE to analyze the pairs contained in the discovery set (Figure 4.5a).

To illustrate the above workflow, we applied SCEPTRE to carry out a complete *trans* analysis of the Papalexi (gene expression) and Frangieh (control) datasets. Many of the genes targeted for knockout in these datasets were transcription factors (TFs); thus, our main biological objective was to map the TFs to their target genes. We carried out a calibration check and discovery analysis on both datasets (Figure 4.5b). These fairly large analyses completed within a matter of hours on a single laptop processor and required a few gigabytes of memory. We used publicly-available ChIP-seq data to validate the SCEPTRE-discovered targets of IRF1 and STAT1 on the Papalexi data. (IRF1 and STAT1 were the only TFs for which cell-type-relevant ChIP-seq data were available.) We found significant enrichment for both TFs (IRF1: odds ratio = 3.94, $p = 8 \times 10^{-76}$; STAT1: odds ratio = 1.37, $p = 5 \times 10^{-16}$), increasing our confidence in the discovery results.

4.3 Discussion

Single-cell CRISPR screens have emerged as a powerful method for linking genetic perturbations to rich phenotypic profiles in individual cells. Although poised to impact a variety of research areas, single-cell CRISPR screens will play an especially important role in dissecting the regulatory logic of the noncoding genome. The bulk of genetic risk for diseases lies in noncoding regions, implicating dysregulation of gene expression (Ward and Kellis, 2012; Maurano et al., 2012; Finucane et al., 2015). A major challenge in genetics, therefore, is to map noncoding disease variants to the genes that they target, target genes to the molecular programs that they regulate, and — ultimately — molecular programs to disease (Schnitzler et al., 2022). Single-cell screens have enabled breakthrough progress on these tasks. For example, two recent studies leveraged high-MOI single-cell screens to perturb blood disease (Morris et al., 2023) and cancer (Tuano et al., 2023) GWAS variants (in some cases at single nucleotide resolution) and link these variants to target genes in disease-relevant cell types. (Both studies used SCEPTRE (high-MOI) to analyze their data.) Another recent study leveraged low-MOI single-cell screens to knock down genes regulated by heart disease GWAS variants and map these genes to downstream molecular programs (Schnitzler et al., 2022). Given the promise that single-cell

screens have demonstrated in understanding noncoding variation, a wave of single-cell screens aiming to link noncoding variants to genes and genes to molecular programs likely will emerge over the coming decade.

It is therefore crucial that reliable methods for single-cell CRISPR screen data analysis be made available. The broad objective of this Chapter was to put single-cell CRISPR screen analysis onto a solid statistical foundation. To this end we devised a simple framework for assessing the calibration and power of competing methods; applied this framework to conduct the first-ever comprehensive benchmarking study of existing methods; identified core statistical challenges that the data pose; and developed a method, SCEPTRE, that combines careful modeling with a resampling procedure to produce a well-calibrated, powerful, fast, and memory-efficient test of association. Taken together, these contributions help bring statistical clarity and rigor to the practice of single-cell CRISPR screen data analysis. Furthermore, given the appealing theoretical properties and empirical performance of the proposed method, we anticipate that the method could be extended (with appropriate modifications) to applications beyond single-cell CRISPR screens, such as single-cell eQTL analysis and single-cell case-control differential expression analysis.

We identified sparsity, confounding, and model misspecification as key challenges in single-cell CRISPR screen analysis. However, the data pose additional challenges that SCEPTRE does not currently address. First, some NT gRNAs may have off-targeting effects. In such cases testing for association by comparing cells that contain a targeting perturbation to those that contain an NT perturbation can result in a loss of error control. At least one prior work has attempted to address this problem (Replogle et al., 2022). Second, some targeting gRNAs are ineffective, i.e., they fail to perturb their target. Including such defective gRNAs in the analysis can result in a loss of power. Several methods, including MIMOSCA (Dixit et al., 2016), MUSIC (Duan et al., 2019), and Mixscape (Papalexi et al., 2021), have attempted to resolve this issue. Third, it is challenging to distinguish between direct and indirect effects, in the sense that perturbations can be associated with their direct targets or with targets further downstream. Disentangling direct from indirect effects likely admits a statistical solution, but to our knowledge, this problem remains unaddressed. Finally, genes often are co-expressed in "gene modules." An exciting opportunity is to pool information across genes within the same module to increase the power of perturbation-to-gene association tests; the recent method GSFA attempts to do just this (Zhou et al., 2022).

In summary single-cell CRISPR screens, though promising, present a variety of statistical challenges, demanding the development of robust analytic methods.

The SCEPTRE toolkit, which now supports both low- and high-MOI CRISPR screens, provides practitioners with a unified solution for reliable and efficient single-cell CRISPR screen differential expression analysis.

4.4 Figures

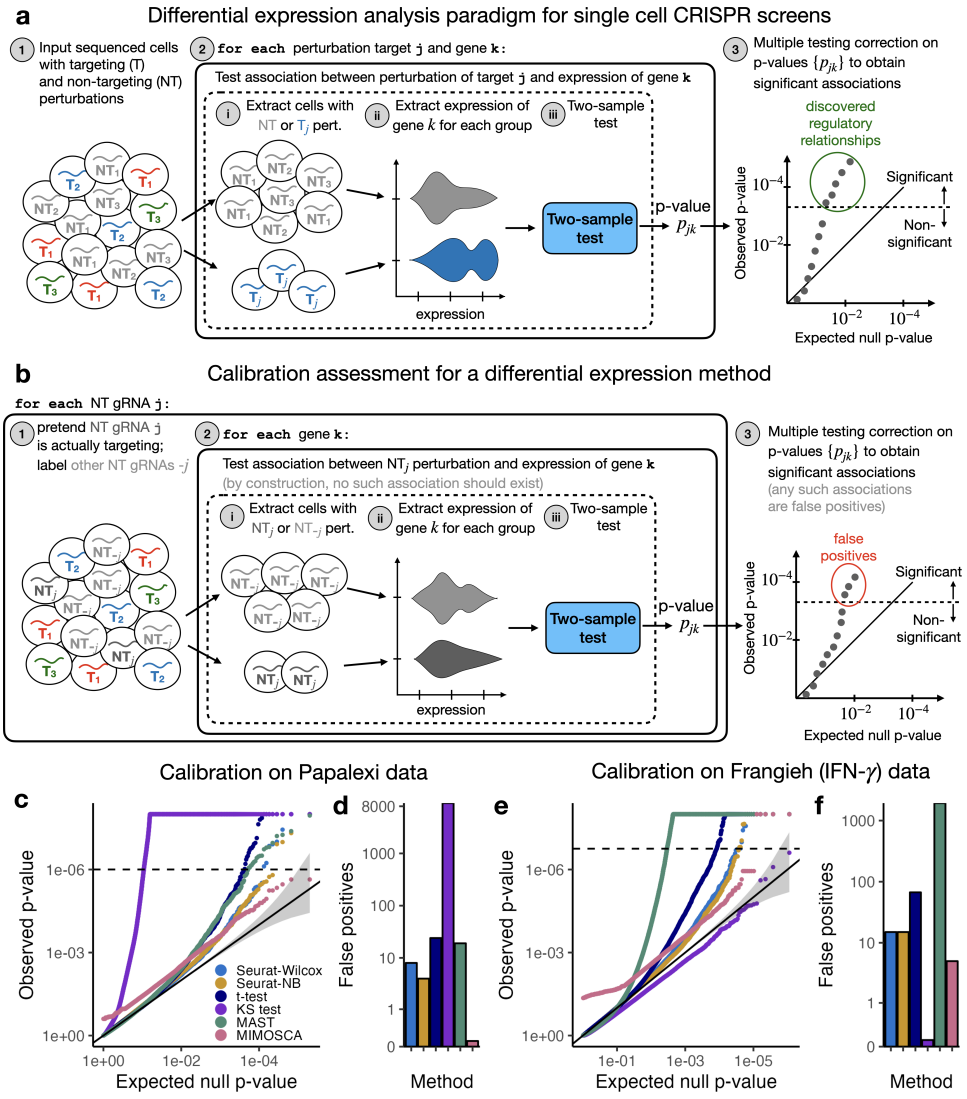


Figure 4.1: (Caption on next page.)

Figure 4.1: **Comprehensive benchmarking study of single-cell CRISPR screen association testing methods on low-MOI data**. **a**, The standard paradigm for association testing on low-MOI single-cell CRISPR screen data. To test for association between a given targeting perturbation and gene, one tests for differential expression of the gene across two groups of cells: those containing the given targeting perturbation, and those containing a non-targeting (NT) perturbation. One typically repeats this procedure for a large, preselected set of targeting-perturbation gene pairs, obtaining a discovery set by subjecting the resulting p-values to a multiple comparison correction procedure (e.g., Benjamini-Hochberg). **b**, The calibration check paradigm. One constructs "null" or "negative control" perturbation-gene pairs by coupling each individual NT gRNA to the entire set of genes. One then assesses the calibration of a method by deploying the method to analyze these null pairs. Any p-values that survive the multiple testing correction procedure correspond to false positive discoveries. **c-d**, Results of the calibration check benchmarking analysis on the Papalexi gene expression data. **c**, QQ plot of the null p-values (colored by method) plotted on a negative log transformed scale. Gray region, 95% confidence band. **d**, Number of false discoveries that each method makes on the null pairs after a Bonferroni correction at level 0.1. **e-f**, Similar to panels **c-d**, but for the Frangieh IFN-γ data.

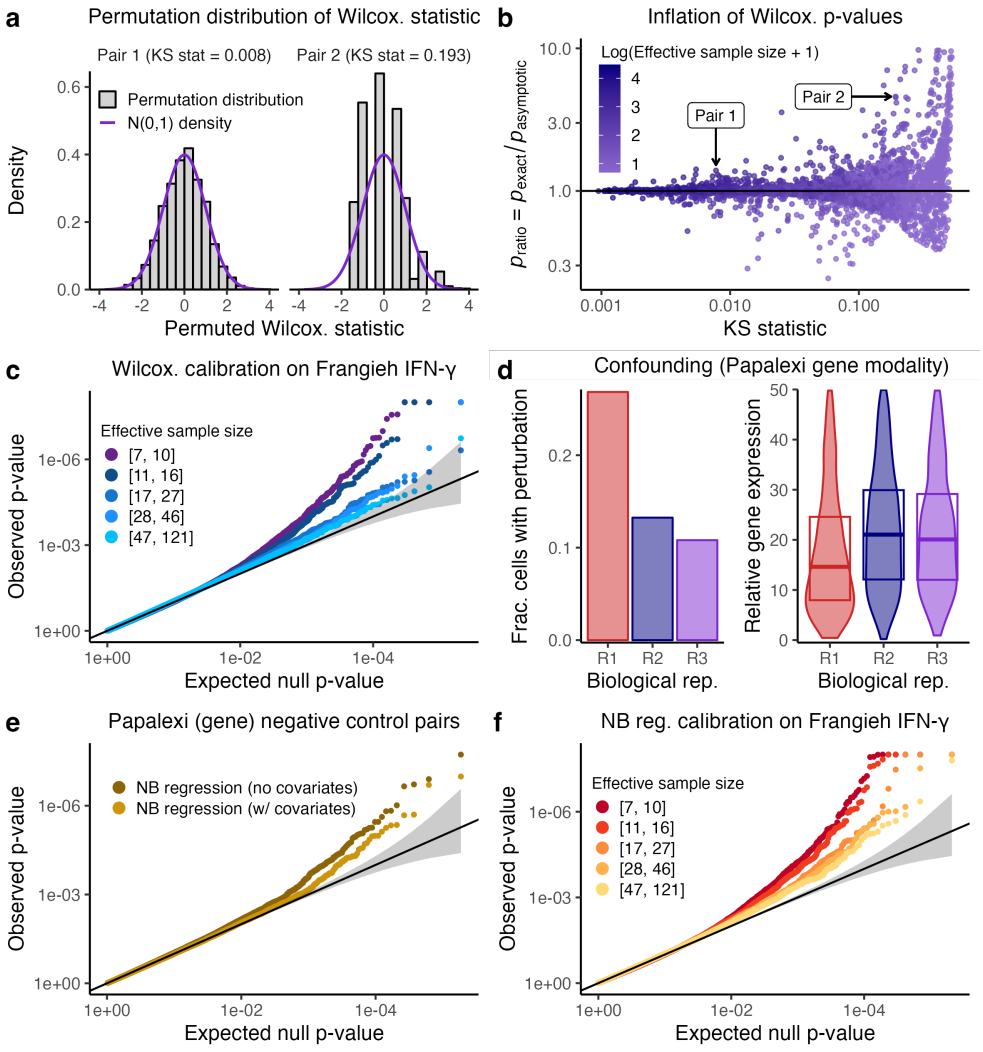


Figure 4.2: (Caption on next page.)

Figure 4.2: **Sparsity, confounding, and model misspecification are core analysis challenges in single-cell CRISPR screen analysis**. **a**, The exact null distribution of the Wilcoxon test statistic (obtained via permutations; gray) on two pairs from the Frangieh IFN-γ data. The Wilcoxon test (and thus Seurat-Wilcox) approximates the exact null distribution using a standard Gaussian density (purple). For pair 1 (left), the Gaussian approximation to the exact null distribution is good (KS statistic = 0.008), while for pair 2 (right) the approximation is inadequate (KS statistic = 0.193). **b**, A plot of p_{ratio} (defined as the ratio of the exact Wilcoxon p-value, p_{exact}, to the asymptotic Wilcoxon p-value, $p_{\text{asymptotic}}$) vs. goodness of fit of the Gaussian distribution to the exact null distribution (as quantified by the KS statistic). Each point represents a gene-gRNA pair; pairs 1 and 2 (from panel **a**) are annotated. As the KS statistic increases (indicating worse fit of the Gaussian distribution to the exact Wilcoxon null distribution), p_{ratio} deviates more from one, indicating miscalibration. Points are colored according to the effective sample size of the corresponding pair. **c**, Stratification of the Seurat-Wilcox p-values on the Frangieh IFN-γ negative control data by effective sample size. **d**, An example of confounding on the Papalexi data. Left (resp. right), the fraction of cells that received a given NT gRNA (resp., the relative expression of a given gene) across biological replicates "R1," "R2," and "R3." **e**, Application of NB regression with and without covariates to the Papalexi data. **f**, Stratification of the NB regression p-values on the Papalexi (gene expression) negative control data by effective sample size.

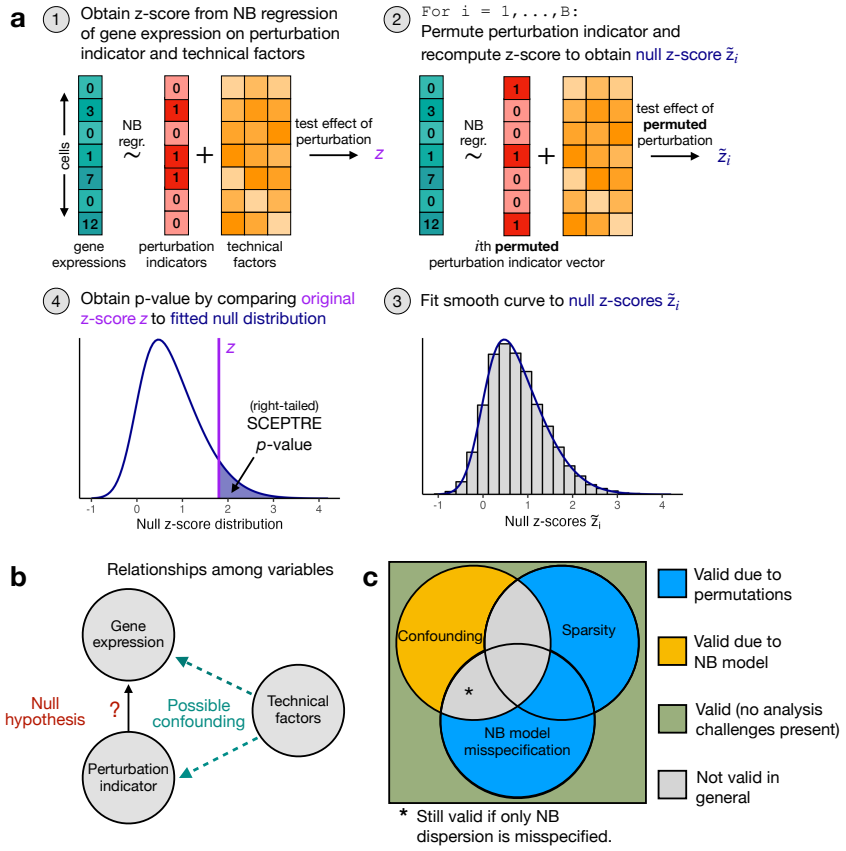
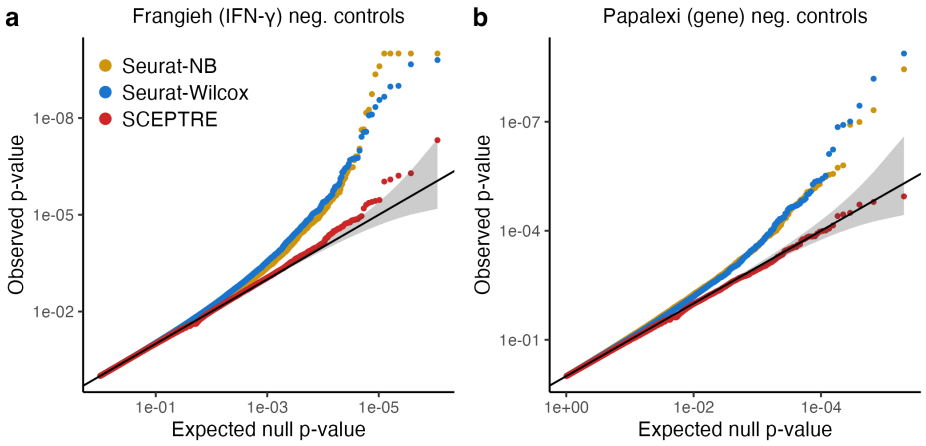


Figure 4.3: (Caption on next page.)

Figure 4.3: **SCEPTRE addresses the core analysis challenges of sparsity, confounding, and model misspecification in theory**. **a**, The SCEPTRE algorithm. First, the gene expressions are regressed onto the perturbation indicators and technical factors, and the z-score z_{obs} corresponding to the perturbation indicator is computed. Second, the perturbation indicators are permuted (while the gene expressions and technical factors are held fixed) and the z-score is recomputed, yielding B "null" z-values. Third, a smooth density is fit to the histogram of the null z-values. Fourth, a p-value is computed by evaluating the tail probability of the fitted density at z_{obs}. **b**, A diagram representing the relationship between the variables in the analysis. The technical factors often (but not always) exert a confounding effect on the perturbation indicator and gene expression. **c**, A diagram illustrating the robustness properties of SCEPTRE. The circles represent analysis challenges. A perturbation-gene pair can be affected any subset of the analysis challenges. The color in each region of the diagram indicates whether SCEPTRE is valid on pairs affected by that subset of analysis challenges (blue, yellow, or green = valid; gray = not valid in general). For regions in which SCEPTRE is valid, the color of the region indicates *why* SCEPTRE is valid (yellow = NB model, blue = permutations). The validity of SCEPTRE is overdetermined on pairs unaffected by *any* analysis challenge (green region) due to the combination of the NB model and permutations.



c Number of false positives

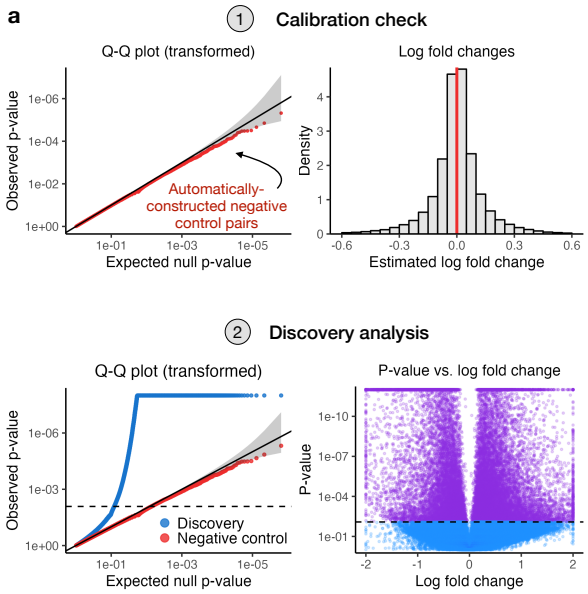
Dataset	SCEPTRE	Seurat-Wilcox	Seurat-NB	t-test	MAST	KS test	MIMOSCA	NT pairs
Frangieh (Co Culture)	1	13	10	89	2083	**0**	4	596344
Frangieh (Control)	**0**	7	16	69	1873	**0**	**0**	528239
Frangieh (IFN-γ)	1	15	15	67	1933	**0**	5	565502
Papalexi (Gene)	**0**	8	4	24	19	9191	**0**	100458
Papalexi (Protein)	**0**	2	**0**	1	2	**0**	**0**	36
Schraivogel	3	2	3	4	**1**	**1**	8	4357
Simulated	**0**	**0**	**0**	7	16	**0**	1	96944
Average	**0.7**	6.7	6.9	37.3	846.7	1313.1	2.6	

d Number of true positives

Dataset	SCEPTRE	Seurat-Wilcox	Seurat-NB	t-test	MAST	KS test	MIMOSCA	PC pairs
Frangieh (Co Culture)	**103**	98	94	-	-	90	5	181
Frangieh (Control)	**77**	74	72	-	-	70	4	170
Frangieh (IFN-γ)	**94**	89	81	-	-	81	8	181
Papalexi (Gene)	**13**	12	**13**	11	11	-	0	25
Papalexi (Protein)	**2**	**2**	**2**	**2**	**2**	**2**	**2**	2
Schraivogel	22	22	21	**23**	22	19	0	25

Figure 4.4: (Caption on next page.)

Figure 4.4: **SCEPTRE demonstrates improved calibration and power relative to existing methods across datasets**. **a** (resp. **b**), QQ plot of the p-values outputted by Seurat-NB, Seurat-Wilcox, and SCEPTRE on the Frangieh IFN-γ (resp., Papalexi gene expression) negative control data. Gray band, 95% confidence region. **c**, Number of false discoveries (at Bonferroni correction level 0.1) on the negative control data for each method-dataset pair. **d**, Number of true discoveries (significant at level $\alpha = 10^{-5}$) on the positive control data for each each method-dataset pair.



b Computational performance metrics* of SCEPTRE

Dataset	Frangieh (Control)	Papalexi (gene)
Number of pairs	2.2 million	323,906
Number of cells	30,486	20,729
Calibration check run time	45 min (794 pair/s)	32 min (167 pair/s)
Calibration check peak memory	3.4 GB	2.4 GB
Discovery analysis run time	8.6 h (70 pair/s)	1.8 h (51 pair/s)
Discovery analysis peak memory	4.0 GB	2.8 GB

*Benchmarking conducted on standard laptop using one processor

Figure 4.5: **Applying SCEPTRE to make biological discoveries**. **a**, The standard workflow involved in applying SCEPTRE to a new dataset, using the Papalexi gene expression data as a running example. First, SCEPTRE is applied to analyze a set of automatically-constructed negative control pairs (the "calibration check"). The resulting negative control p-values are plotted on a QQ plot to ensure uniformity (upper left), and the negative control log-fold changes are plotted on a histogram to ensure symmetry about zero (upper right). Second, SCEPTRE is applied to analyze the discovery pairs (the "discovery analysis"). The discovery p-values are superimposed over the negative control p-values to ensure that signal is present in the discovery set (lower left), and a volcano plot is created (lower right). **b**, Computational performance metrics of SCEPTRE on the Frangieh (control) and Papalexi (gene expression) data. A complete *trans* analysis was conducted on both datasets. Several metrics are reported, including calibration check run time, calibration check peak memory usage, discovery analysis run time, and discovery analysis peak memory usage.

4.5 Methods

Dataset details

We downloaded, processed, and harmonized five single-cell CRISPR screen datasets (Table C.1), inheriting several data-related analysis decisions made by the original authors. First, we used the gRNA-to-cell assignments that the original authors used, thereby circumventing the need to assign gRNAs to cells using gRNA UMI and/or read count matrices. Papalexi and Schraivogel employed a simple strategy for this purpose: Papalexi identified the gRNA with the greatest UMI count in a given cell and assigned that gRNA to the cell, while Schraivogel assigned gRNAs by thresholding gRNA UMI counts. Frangieh, meanwhile, assigned gRNAs to cells via a more complex approach involving a separate dial-out PCR procedure. We found the gRNA-to-cell assignments adequate and thus used them without modification. Next, we inherited the cell-wise QC that the original authors implemented. For example, Papalexi removed likely duplets (as determined by the `Seurat` function `MULTIseqDemux`; McGinnis et al. 2019; Hao et al. 2021) as well as cells with excessive mitochondrial content and low gene expression.

We generated a synthetic single-cell CRISPR screen dataset to use for benchmarking purposes. The synthetic dataset contained 5,000 genes, 25 gRNAs, and 10,000 cells. We generated the matrix of gene expressions by sampling counts from a negative binomial distribution, allowing each gene to have its own mean and size parameter. (We drew gene-wise means and sizes i.i.d. from a Gamma$(0.5, 2)$ distribution and a Unif$(1, 25)$ distribution, respectively.) We randomly inserted gRNAs into cells such that the expected number of cells per gRNA was equal across gRNAs. The dataset was entirely devoid of signal and confounding: no gRNA affected the expression of any gene, and no technical factors impacted the gRNA assignments or gene expressions.

We applied our own minimal gene-wise, gRNA-wise, and cell-wise QC uniformly to the datasets. We filtered for genes expressed in at least 0.005 of cells, gRNAs expressed in at least 10 cells, and cells with exactly one gRNA, respectively. Table C.2 summarizes the statistical attributes (e.g., number of genes, number of cells, etc.) of each dataset. Finally, we obtained the set of cell-specific covariates (or technical factors) for each dataset, which we list below. Frangieh co-culture, control, and IFN-γ datasets: number of gene UMIs, number of genes expressed; Papalexi (gene modality): number of gene UMIs, number of genes expressed, biological replicate, and percent of gene transcripts that mapped to mitochondrial genes; Papalexi (protein modality): number of protein UMIs, biological replicate, and percent of gene transcripts that mapped to mitochondrial genes; Schraivogel: number of gene UMIs, number of genes

expressed, sequencing lane.

Existing methods details

We benchmarked the performance of six methods: Seurat-Wilcox, Seurat-NB, t-test, MAST, KS test, and MIMOSCA. The first five of these methods are generic single-cell differential expression methods that have been adapted to single-cell CRISPR screens (either by us or other single-cell researchers), while MIMOSCA is specific to single-cell screens. To facilitate benchmarking of the methods, we implemented all in an R package `lowmoi` (`github.com/Katsevich-Lab/lowmoi`). We implemented Seurat-Wilcox and Seurat-NB via a call to the Seurat `FindMarkers()` function. In the case of Seurat-Wilcox, we called `NormalizeData()` before `FindMarkers()` to normalize the gene expressions by dividing the gene expressions by library size. Next, we implemented the t-test via a call to `t.test()` in R. Following Liscovitch-Brauer et al. (2021), we normalized the gene expression vector for a given gene-perturbation pair by dividing by the library size, subtracting the mean, and dividing by the standard deviation. We used the implementation of MAST that Schraivogel et al. (2020) used to analyze their single-cell screen data. To this end we copied and pasted relevant portions of the Schraivogel et al. Github codebase (`github.com/argschwind/TAPseq_manuscript`) into `lowmoi`. Similarly, we used the implementation of the KS test that Replogle et al. (2020) used to analyze their single-cell screen data, again copying and pasting relevant portions of the corresponding codebase into `lowmoi` (`github.com/thomasmaxwellnorman/Perturbseq_GI`). Finally, we implemented MIMOSCA by copying and pasting relevant sections of the MIMOSCA package (`github.com/klarman-cell-observatory/Perturb-CITE-seq`) into `lowmoi`. Replogle et al.'s implementation of the KS test and MIMOSCA both were written in Python. Thus, we used the `reticulate` package to access these methods from within R. To ensure consistency of the API across methods, we implemented the methods in such a way that each took the same inputs and returned the same output. Finally, to ensure correctness, we tested for agreement between the output of our implementations and those of the original methods (when possible).

We did not apply the methods directly "out of the box" and instead adjusted them in two small ways to enable their more meaningful application to the data. First, some methods have an internal QC step in which gene-perturbation pairs that are unpromising or low-quality (as determined by the method itself) are removed. For example, Seurat DE by default filters out gene-perturbation pairs for which the log-fold change of the expression of the gene (across the treatment and control cells) falls below a certain threshold. We disabled such method-

specific pairwise QC, allowing us to apply competing methods to the exact same set of gene-perturbation pairs on each dataset, facilitating head-to-head comparisons across methods. Second, we deployed SCEPTRE and MIMOSCA such that they adjusted for the cell-specific technical factors (beyond library size) of each dataset (as listed in the Section Dataset details).

We applied several variants of NB regression to the data. First, as described above, we applied Seurat-NB the negative control and positive control pairs of all datasets. Furthermore, as part of our investigation into the analysis challenges, we applied NB regression as implemented by the `MASS` package (Ripley et al., 2013) to the Papalexi (gene expression) and Frangieh IFN-γ negative control data. (These results are depicted in Figure 4.2e-f). We used the `MASS` implementation of NB regression in exploring the analysis challenges, as `MASS` enables the inclusion of covariates. Within the context of `MASS` NB regression, we tested for association between a perturbation and the expression of a gene via a GLM score test, as implemented by the `statmod` package (Dunn and Smyth, 2018). We elected to use a score test (as opposed to a more standard Wald or likelihood ratio test) test to make our implementation of NB regression more comparable to SCEPTRE, as SCEPTRE uses a permutation test built upon an NB regression score test statistic.

Details of the calibration check procedure

We describe the calibration check procedure in greater detail. Suppose there are d distinct NT gRNAs; index these gRNAs from 1 to d. Let $\mathcal{C}_1$ denote the set of cells containing NT gRNA 1, $\mathcal{C}_2$ the set of cells containing NT gRNA 2, etc. Let $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_d$ denote the set of cells containing *any* NT gRNA (i.e., the "NT cells"). Next, let $\mathcal{C} \setminus \mathcal{C}_i$ denote the set of cells containing *any* NT gRNA *excluding* NT gRNA i. Additionally, let $\mathcal{T}$ denote the set of cells containing *any targeting* gRNA. (Observe that $\mathcal{T} \cup \mathcal{C}$ is the set of all cells.) Finally, let $\mathcal{T} \cup \mathcal{C} \setminus \mathcal{C}_i$ denote the set of all cells excluding the cells that contain NT gRNA i. Let there be p distinct genes.

Suppose we seek to check the calibration of a given method. The way in which we deploy the method to analyze a given negative control pair depends on whether the method uses the NT cells or the complement set as its control group (Table 4.1). Consider the negative control pair formed by coupling NT gRNA i to gene j. If the method uses the NT cells as its control group (e.g., Seurat-Wilcox, Seurat-NB, SCEPTRE, etc.), then we apply the method to test for differential expression of gene j across the groups of cells $\mathcal{C}_i$ and $\mathcal{C} \setminus \mathcal{C}_i$. By contrast, if the method uses the complement set as its control group (e.g., MIMOSCA), then we apply the method to test for differential expression of gene j across the groups of cells $\mathcal{T} \cup \mathcal{C} \setminus \mathcal{C}_i$ and $\mathcal{C}_i$. The "effective sample size"

of the given negative control pair is the number of cells in the set $\mathcal{C}_i$ for which the expression of gene j is nonzero. In carrying out our benchmarking analysis (Figure 4.1c-f, Figure 4.4), we restricted our attention to the subset of the $d \cdot p$ possible negative control pairs whose effective sample size was greater than or equal to seven.

For testing calibration on a given input dataset, the SCEPTRE software automatically constructs a set of negative control pairs that is matched to the pairs in the "discovery set" — i.e., the set of targeting perturbation-gene pairs that the user seeks to test for association — in several respects. First, the negative control pairs and discovery pairs are subjected to the same pairwise QC. Second, the number of negative control pairs is set equal to the number of positive control pairs (assuming the number of possible negative control pairs matches or exceeds the number of discovery pairs). Third, if the user elects to "group" together gRNAs that target the same site (as opposed to running an analysis in which singleton gRNAs are tested for significance), then the negative control pairs likewise are constructed by "grouping" together individual NT gRNAs. Overall, the negative control pairs are designed to mirror the discovery pairs, the difference being that the negative control pairs are devoid of biological signal. More information about the construction of the negative control pairs is available on the SCEPTRE website: `katsevich-lab.github.io/sceptre/`.

Details of the investigation into the core analysis challenges

We describe in greater detail our empirical investigations into the core analysis challenges of sparsity, confounding, and model misspecification (as described in Section Systematic identification of core analysis challenges).

Sparsity. To explore the impact of sparsity on calibration, we deployed the two-sample Wilcoxon test to a randomly-selected subset of 5,400 negative control gene-gRNA pairs from the Frangieh IFN-γ data. (The pairs were selected such that each had an effective sample size of one or greater.) Following Seurat-Wilcox, we deployed the Wilcoxon test as follows: first, we normalized the gene expressions by dividing the raw counts by the cell-specific library sizes; then, we applied the Wilcoxon test (as implemented by the `wilcox.test` function from the `stats` package in R) to the normalized data, comparing the treatment cells to the control cells. Finally, we computed the Wilcoxon p-value in two ways. First, we calculated the asymptotic p-value $p_{\text{asymptotic}}$ by comparing the Wilcoxon test statistic to the standard Gaussian distribution. This approach implicitly assumes that the number of cells with nonzero expression (across both groups) is large enough for the null distribution of the Wilcoxon test statistic to be approximately Gaussian. Next, we calculated the exact p-value p_{exact} by (i) computing the Wilcoxon statistic on the original data; (ii) permuting the

gRNA indicator vector $B = 200,000$ times (while holding fixed the vector of normalized gene expressions), resulting in B permuted datasets; (iii) computing the Wilcoxon test statistic on each of these B permuted datasets, yielding a permutation (or "null") distribution of Wilcoxon statistics; and then (iv) calculating the p-value p_{exact} by comparing the original Wilcoxon statistic to the null Wilcoxon statistics (Chung and Romano, 2016). The latter approach, though computationally expensive (due to the slowness of computing the Wilcoxon statistic), yields a much more accurate p-value than the asymptotic approach for lowly expressed genes. Seurat-Wilcox returns the asymptotic p-value $p_{\text{asymptotic}}$ instead of the exact p-value p_{exact} in virtually all cases.[1]

To study the impact of making the above approximation, we plotted the asymptotic null distribution of the Wilcoxon statistic (i.e., the standard Gaussian distribution) superimposed on top of the exact null distribution of the Wilcoxon statistic (i.e., the permutation distribution) for two pairs from the Frangieh IFN-γ negative control data (Figure 4.2a). The asymptotic and exact distributions must be highly similar for the asymptotic p-value $p_{\text{asymptotic}}$ to be accurate. We measured goodness of fit of the Gaussian distribution to the exact null distribution by calculating the Kolmogorov–Smirnov (KS) statistic; this statistic ranges from zero to one, with smaller values indicating better fit of the Gaussian distribution to the exact null distribution. We reported the KS statistic for both example pairs in the panels of the plot.

Next, we calculated p_{ratio}, defined as the ratio of the exact p-value p_{exact} to the asymptotic p-value $p_{\text{asymptotic}}$, for each of the the 5,400 negative control pairs sampled from the Frangieh IFN-γ data. A p_{ratio} value of one indicates that the asymptotic and exact p-values coincide; a p_{ratio} value of greater than one (resp., less than one), on the other hand, indicates inflation (resp., deflation) of the asymptotic p-value relative to the exact p-value. We sought to explore visually how a small effective sample sizes lead to degradation of the Gaussian approximation, thereby resulting in p-value miscalibration (as reflected by p_{ratio} values that deviate from one). To this end, we plotted p_{ratio} versus goodness of fit of the the Gaussian distribution to the exact null distribution (as quantified by the KS statistic) for each pair (Figure 4.2b). We colored the points according to their effective sample size. Pairs 1 and 2 from Figure 4.2a were annotated in Figure 4.2b.

Finally, to directly assess the impact of sparsity on calibration, we applied Seurat-Wilcox to the IFN-γ negative control data, binning the pairs into five

[1]The `wilcox.test` function on which Seurat-Wilcox relies returns p_{exact} only if (i) there are fewer than 50 cells across both treatment and control groups and (ii) no two cells (in either the treatment or the control group) have the same normalized expression level. This condition is expected to hold rarely, if ever.

categories based on their effective sample size. The bins were defined by effective sample sizes in the ranges [7,10], [11,16], [17,27], [28,46], and [47,121]. The bins were constructed such that an approximately equal number of pairs would fall into each bin. We observed that as the effective sample size increased, the Seurat-Wilcox p-values converged to uniformity, illustrating that sparsity is a cause of the miscalibration of Seurat-Wilcox.

Confounding. We explored how the variable of biological replicate confounded the Papalexi (gene modality) data. The Papalexi data were generated and sequenced across three independent experimental replicates (which we label "R1," "R2," and "R3")[2]. We visually examined the relationship between biological replicate and a given NT gRNA ("NTg4") and a given gene (*FTH1*). We plotted the fraction of cells in each biological replicate that received the NT gRNA (Figure 4.2d, left); additionally, we created a violin plot of the relative expression of the gene across biological replicate. (The relative expression r_i of the gene in cell i was defined as $r_i = 1000 \cdot \log\left(u_i/l_i + 1\right)$, where u_i was the UMI count of the gene in cell i, and l_i was the library size of cell i. The violin plots were truncated at a relative expression level of 50). We superimposed boxplots indicating the 25th, 50th, and 75th percentiles of the empirical relative expression distributions on top of the violin plots (Figure 4.2d, right). We observed clear visual evidence that biological replicate impacted both NTg4 and *FTH1*, creating a confounding effect.

Next, we extended the above analysis to investigate the entire set of NT gRNAs and genes. First, we tested for association between each NT gRNA and biological replicate. To this end, we constructed a contingency table of gRNA presences and absences across biological replicate, testing for significance of the contingency table using a using a Fisher exact test (as implemented in the R function `fisher.test`). Next, we tested for association between the relative expression of each gene and biological replicate. To do so, we fit two NB regression models to each gene; the first contained only library size as a covariate, while the second contained both library size *and* biological replicate as covariates. We compared these two models via a likelihood ratio test, yielding a p-value for the test of association between relative gene expression and biological replicate. Finally, we created QQ plots of the resulting p-values (Figure C.5; gRNA p-values, left; gene p-values, right). An inflation of the p-values across modalities suggested that the bulk of gene-NT gRNA pairs was confounded by biological replicate.

Finally, we directly assessed the impact of adjusting for biological replicate

[2]The original data contained a fourth biological replicate as well, but this replicate was removed by the original authors, as it was deemed to be low quality.

(alongside other potential confounders) by applying two variants of NB regression to the Papalexi (gene modality) negative control data: (i) NB regression with library size (only) included as a covariate, and (ii) NB regression with library size as well as all potential confounders (including biological replicate) included as covariates. We plotted the negative control p-values on a QQ plot (Figure 4.2e). The variant of NB regression with confounders included as covariates exhibited superior calibration, demonstrating that confounding is an analysis challenge. To reduce the effect of sparsity (i.e., the first analysis challenge), we restricted our attention in this plot to gene-gRNA pairs with an effective sample size greater than 10.

Model misspecification. To explore the analysis challenge of model misspecification, we applied NB regression to the Frangieh IFN-γ negative control data. As in Figure 4.2c (in which we applied Seurat-Wilcox to the Frangieh IFN-γ negative control data), we partitioned the pairs into five categories based on the effective sample size of each pair. As the number of nonzero treatment cells increased, the NB regression p-values failed to converge to uniformity (in contrast to the Seurat-Wilcox p-values). The key difference between Seurat-Wilcox and NB regression is that the former is a nonparametric method while the latter is parametric method. Thus, we reasoned that miscalibration of the NB regression p-values likely was due to misspecification of the NB regression model. (We note that miscalibration of the NB regression p-values likely was not due to confounding, as Seurat-Wilcox, which does not adjust for confounding, was well-calibrated for pairs with high expression levels.)

SCEPTRE (low-MOI) overview

Consider a given gene and perturbation. We call the cells that contain the targeting perturbation the "treatment cells" and those that contain an NT perturbation the "control cells." Suppose there are n cells across treatment and control groups. Let $Y = [Y_1, \ldots, Y_n]^T$ be the vector of raw gene (or protein) expressions, and let $X = [X_1, \ldots, X_n]^T$ be the vector of perturbation indicators, where an entry of one (resp., zero) indicates presence of the targeting (resp. NT) perturbation. Finally, for cell $i \in \{1, \ldots, n\}$, let Z_i be the p-dimensional vector of technical factors for cell i (containing library size, batch, etc.). We include an entry of one in each Z_i to serve as an intercept term. Let Z be the $n \times p$ matrix formed by concatenating the Z_is, and let $[X, Z]$ be the $n \times (p+1)$ matrix formed by concatenating X and Z.

We model Y_i as a function of X_i and Z_i via an NB generalized linear model (GLM):

$$Y_i \sim \text{NB}_\theta(\mu_i); \quad \log(\mu_i) = \gamma X_i + \beta^T Z_i, \tag{4.1}$$

where $\text{NB}_\theta(\mu_i)$ denotes a negative binomial distribution with mean μ_i and size parameter θ, and $\gamma \in \mathbb{R}$ and $\beta \in \mathbb{R}^p$ are unknown constants. (In fact, SCEPTRE is compatible with arbitrary GLMs, including Poisson GLMs, which may be more appropriate for highly sparse data.) SCEPTRE is a permutation test that uses as its test statistic the z-score that results from testing the hypothesis $\gamma = 0$ in the model (4.1). We present the basic SCEPTRE algorithm in Algorithm 2. Five key accelerations speed Algorithm 2 by several orders of magnitude.

Algorithm 2: Basic SCEPTRE algorithm.

1. Regress Y onto the matrix $[X, Z]$ by fitting the GLM (4.1). Compute a z-score z_{orig} for a test of the null hypothesis $H_0 : \gamma = 0$.
2. Permute the X vector B (e.g., $B = 5,000$) times, resulting in permuted vectors $\tilde{X}_1, \ldots, \tilde{X}_B$.
3. For each $i \in \{1, \ldots, B\}$, regress Y onto the matrix $[\tilde{X}_i, Z]$. Test the null hypothesis $H_0 : \gamma = 0$, and label the resulting z-score z_i.
4. Compute a left-tailed (p_{left}), right-tailed (p_{right}), or two-tailed (p_{both}) p-value using the standard permutation test p-value formula:

$$\begin{cases} p_{\text{right}} = \frac{1}{B+1}\left(1 + \sum_{i=1}^{B} \mathbb{I}(z_{\text{orig}} \geq z_i)\right) \\ p_{\text{left}} = \frac{1}{B+1}\left(1 + \sum_{i=1}^{B} \mathbb{I}(z_{\text{orig}} \leq z_i)\right) \\ p_{\text{both}} = 2 \cdot \min\left\{p_{\text{right}}, p_{\text{left}}\right\}. \end{cases}$$

Acceleration 1: Score test. First, we use a GLM score test to compute the test statistics $z_{\text{orig}}, z_1, \ldots, z_B$. Consider the following simplified NB GLM in which the gene expression Y_i is modelled as a function of the technical factor vector Z_i only:

$$Y_i \sim NB_\theta(\mu_i); \quad \log(\mu_i) = \beta^T Z_i. \tag{4.2}$$

Regressing Y onto Z by fitting the GLM (4.2) produces estimates $\hat{\beta}$ and $\hat{\theta}$ of the coefficient vector β and the size parameter θ, respectively, under the null hypothesis of no relationship between the gRNA indicator and the gene expression. Denote the ith fitted mean of the model by $\hat{\mu}_i = \exp(\hat{\beta}^T Z_i)$, and let $\hat{\mu} = [\hat{\mu}_1, \ldots, \hat{\mu}_n]^T$ be the vector of fitted means. We can test the gRNA indicator vector X for inclusion in the fitted model by computing a score statistic z_{score}, as follows:

$$z_{\text{score}} = \frac{X^T W M (Y - \hat{\mu})}{\sqrt{X^T W X - X^T W Z (Z^T W Z)^{-1} Z^T W X}}. \tag{4.3}$$

See Additional mathematical details of SCEPTRE (low MOI) for a derivation of this formula. Here, W and $M(Y - \hat{\mu})$ are a matrix and vector, respectively, that depend on the fitted means $\hat{\mu}$, gene expressions Y, and estimated size $\hat{\theta}$:

$$W = \text{diag}\left\{\frac{\hat{\mu}_1}{1 + \hat{\mu}_1/\hat{\theta}}, \ldots, \frac{\hat{\mu}_n}{1 + \hat{\mu}_n/\hat{\theta}}\right\}; \quad M(Y - \hat{\mu}) = \left[\frac{Y_1}{\hat{\mu}_1} - 1, \ldots, \frac{Y_n}{\hat{\mu}_n} - 1\right]^T.$$

The score statistic (4.3) is asymptotically equivalent to the Wald or likelihood ratio statistic that one obtains by testing $H_0 : \gamma = 0$ in the full model (4.1). However, unlike the Wald statistic, the score statistic only depends on a fit of the model under the null hypothesis. SCEPTRE (with score statistic; Algorithm 3) exploits this useful property of the score statistic to accelerate the basic SCEPTRE algorithm.

Algorithm 3: SCEPTRE (with score statistic) algorithm.

1. Regress Y onto the matrix Z by fitting the GLM (4.2).
2. Compute the score statistic for X using the formula (4.3), yielding z_{orig}.
3. Permute the X vector B times, generating $\tilde{X}_1, \ldots, \tilde{X}_B$.
4. For each $i \in \{1, \ldots, B\}$, repeat step 2, substituting the vector $\tilde{X}_i$ for X. Label the resulting z-scores $z_1, \ldots, z_B$.
5. Compute a p-value using the standard permutation test p-value formula from step 4 of Algorithm 2.

Acceleration 2: A fast score test for binary treatments. Calculating the score statistic (4.3) is not trivial. The quadratic form

$$X^T W Z (Z^T W Z)^{-1} Z^T W X$$

in the denominator of (4.3) is hard to compute, as the matrix $WZ(Z^TWZ)^{-1}Z^TW$ is a large, dense matrix. The classical solution is to algebraically manipulate the score statistic so that it can be evaluated via a QR decomposition. However, the QR decomposition approach fails to leverage the structure in X when X contains many zeros (as is the case in single-cell CRISPR screen analysis). We therefore devised an alternate strategy for computing the score statistic that instead is based on a spectral decomposition; the proposed strategy is hundreds of times faster than the QR decomposition approach in the single-cell CRISPR screen setting.

First, observe that Z^TWZ is a symmetric matrix. Thus, Z^TWZ can be spectrally decomposed as $Z^TWZ = U^T \Lambda U$, where U is an orthonormal matrix

and Λ is a diagonal matrix of eigenvalues. Exploiting this decomposition, we can express the quadratic form in the denominator of (4.3) as follows:

$$X^T W Z (Z^T W Z)^{-1} Z^T W X = X^T W Z U \Lambda^{-1/2} \Lambda^{-1/2} U^T Z^T W X = L^T L = ||L||^2,$$

where $L = \Lambda^{-1/2} U^T Z^T W X$ is a p-dimensional vector. Evaluating the above expression reduces to computing the vector L and then summing over the squared entries of L, which is fast. This insight motivates Algorithm 4, which computes the score statistics for $X, \tilde{X}_1, \ldots, \tilde{X}_B$ via a spectral decomposition.[3] The inner product and matrix-vector multiplication operations of step 3 are extremely fast because X_{curr} is sparse. Furthermore, we program step 3 in C++ (via Rcpp; Eddelbuettel and François 2011) for maximum speed.

Algorithm 4: Computing the GLM score statistics for $X, \tilde{X}_1, \ldots, \tilde{X}_B$ via spectral decomposition. Below, w is the n-dimensional vector constructed from the diagonal entries of W.

1. Spectrally decompose the matrix $Z^T W Z$, yielding diagonal matrix of eignvalues Λ and an orthonormal matrix U.
2. Compute the matrix $B = \Lambda^{-1/2} U^T Z^T W$ and the vector $a = WM(Y - \hat{\mu})$.

for $X_{\text{curr}} \in \{X, \tilde{X}_1, \ldots, \tilde{X}_B\}$ **do**

3. Compute
$$\begin{cases} \texttt{top} = a^T X_{\text{curr}} \\ \texttt{bottom\_right} = B X_{\text{curr}} \\ \texttt{bottom\_left} = w^T X_{\text{curr}}. \end{cases}$$
4. Compute $z = \texttt{top} / (\sqrt{\texttt{bottom\_left} - ||\texttt{bottom\_right}||^2})$

end

Acceleration 3: Adaptive permutation testing. Computing a large number of permutation resamples for a gene-gRNA pair that yields an unpromising p-value after only a few thousand resamples is wasteful. To reduce this inefficiency, we implement a two-step adaptive permutation testing scheme. First, we compute the p-value of a given gene-gRNA pair out to a small number (e.g., $B_1 = 500$) of resamples. If this initial p-value is unpromising (i.e., if it

[3] A Cholesky decomposition of $Z^T W Z$ could be used in place of the spectral decomposition, but the spectral decomposition is slightly more general, as it applies to matrices with eigenvalues equal to zero, which can occur (for example) when columns of Z are highly correlated.

exceeds some pre-selected threshold of p_{thresh}, where $p_{\text{thresh}} \approx 0.01$), then we return this p-value to the user. Otherwise, we draw a larger number ($B_2 = 5,000$) of fresh resamples and compute the p-value using this second set of resamples. As most pairs are expected to be null (and thus yield unpromising p-values), this procedure eliminates most of the compute associated with carrying out the permutation tests.

Acceleration 4: Skew-normal fit. The null distribution of the test statistics $z_1, \ldots, z_B$ converges to a standard Gaussian distribution as the number of cells increases. Thus, to compute a precise p-value using a small number of permutations, we fit a skew-normal distribution to the set of null statistics. We then compute a p-value by evaluating the tail probability of the fitted skew-normal distribution at the observed test statistic z_{obs}. If the skew-normal fit to the null statistics is poor (an event that happens rarely), we instead return the standard permutation test p-value. We fit the skew-normal distribution via a method of moments estimator and evaluate the skew-normal tail probability via the C++ Boost library. All operations involving the skew-normal distribution are fast.

Acceleration 5: sharing compute across permutation tests. When carrying out a permutation test to test for association between a gene expression vector $Y = [Y_1, \ldots, Y_n]^T$ and a perturbation indicator vector $X = [X_1, \ldots, X_n]^T$, one must randomly permute the perturbation indicator vector B times, where B is some large number (e.g., $B \approx 5,000$). Unfortunately, randomly permuting the perturbation indicator vector B times is slow; this cost becomes prohibitive when testing many perturbation-gene pairs. We therefore derived a novel strategy for "sharing" a set of B randomly permuted indicator vectors across all perturbation-gene pairs, even pairs for which the number of cells containing the targeting perturbation differs. This strategy — which we call "inductive without replacement" (IWOR) sampling — considerably reduces the cost associated with applying SCEPTRE to the data. (In fact, this method is generic, compatible with any permutation-based single-cell CRISPR screen association testing method.) IWOR sampling is decribed in Additional mathematical details of SCEPTRE (low MOI).

Statistical robustness property of SCEPTRE

SCEPTRE possesses a robustness property that we term "confounder adjustment via marginal permutations," or "CAMP." We state CAMP in a slightly more formal way here. For simplicity we consider the version of SCEPTRE that does *not* involve fitting a skew-normal distribution to the null test statistics and instead computes the standard permutation test p-value by directly comparing the observed test statistic to the null test statistics. If at least one of the

following conditions holds, then the left-, right-, and two-tailed SCEPTRE p-values are valid: (i) the perturbation is unconfounded (i.e., the vector of technical factors Z_i contains all possible confounders, and Z_i is independent of X_i); (ii) the NB GLM (4.1) is correctly specified up to the size parameter θ and the effective sample size is sufficiently large. CAMP imbues SCEPTRE with two considerable advantages relative a standard NB GLM. First, SCEPTRE always yields valid inference when the perturbation is unconfounded, even if the NB model is arbitrarily misspecificed or the effective sample size is small. Second, when confounding is non-negligible, SCEPTRE yields valid inference if the NB GLM is correctly specified up to the size parameter and the effective sample size is sufficiently large, sidestepping the difficult problem of NB size parameter estimation (Lause et al., 2021; Love et al., 2014). These two improvements enable SCEPTRE to address the core single-cell CRISPR screen analysis challenges of sparsity, confounding, and model misspecification in theory.

Simulation study details

We conducted a simulation study (Figure C.6) to demonstrate the existence and utility of the CAMP ("confounder adjustment via marginal permutations") phenomenon. We based the simulation study on a gene (namely, *CXCL10*) and perturbation (namely, "CUL3") from the Papalexi data. Following the notation introduced in Section SCEPTRE (low-MOI) overview, let $Y = [Y_1, \ldots, Y_n]^T$ denote the vector of gene expressions of *CXCL10* and $X = [X_1, \ldots, X_n]^T$ the vector of perturbation indicators of "CUL3." Next, let $Z_i \in \mathbb{R}^p$ denote the vector of technical factors of the ith cell (for $i \in \{1, \ldots, n\}$), and let Z denote the $n \times p$ matrix formed by assembling the Z_is into a matrix. We regressed Y onto Z by fitting the GLM (4.2), yielding estimates $\hat{\beta}$ for β and θ^* for θ under the null hypothesis of no association between the perturbation and gene. An examination of $\hat{\beta}$ revealed that the gene expressions Y were moderately associated with the technical factors Z. Letting $\hat{\mu}_i = \exp(\hat{\beta}^T Z_i)$ denote the fitted mean of cell i, we sampled B i.i.d. synthetic expressions $\tilde{Y}_i^1, \ldots, \tilde{Y}_i^B$ from an NB model with mean $\hat{\mu}_i$ and size parameter θ^*. We then constructed B synthetic gene expression vectors $\tilde{Y}^j = [\tilde{Y}_1^j, \ldots, \tilde{Y}_n^j]^T \in \mathbb{R}^n$ for $j \in \{1, \ldots, B\}$. Next, we generated a synthetic perturbation indicator vector $\tilde{X} \in \mathbb{R}^n$ such that $\tilde{X}$ was independent of Z. To this end, we marginally sampled synthetic perturbation indicators $\tilde{X}_1, \ldots, \tilde{X}_n$ i.i.d. from a Bernoulli model with mean $\hat{\pi}$, where $\hat{\pi} = (1/n) \sum_{i=1}^n X_i$ was the fraction of cells that received the targeting perturbation. (The observed perturbation indicator vector X was moderately associated with Z.)

We assessed three methods in the simulation study: NB regression, SCEP-

TRE, and the standard permutation test. We deployed NB regression and SCEPTRE in a slightly different way than usual: we set the NB size parameter θ upon which these methods rely to a fixed value. (Typically, NB regression and SCEPTRE estimate θ using the data.) This enabled us to assess the impact of misspecification of the size parameter on the calibration of NB regression and SCEPTRE. We set the test statistic of the standard permutation test to the sum of the gene expressions in the treatment cells. We then generated B confounded (resp., unconfounded) datasets by pairing the synthetic response vectors $\tilde{Y}_1, \ldots, \tilde{Y}_B$ to the design matrix $[X, Z]$ (resp., $[\tilde{X}, Z]$). We applied the methods to the datasets twice: once setting the SCEPTRE/NB regression size parameter to the correct value of θ^*, and once setting this parameter to the incorrect value of $5 \cdot \theta^*$. We displayed the results produced by the methods in each of the four settings (i.e., confounded versus unconfounded, correct versus incorrect specification of the size parameter; Figure C.6) on a QQ plot. We sought to show that SCEPTRE maintains calibration in all settings, while the standard permutation test and NB regression break down under confounding and incorrect specification of the size parameter, respectively.

Positive control analysis

We grouped together gRNAs that targeted the same genomic location, referring to these grouped gRNAs as "gRNA groups" (Gasperini et al., 2019). We constructed positive control pairs by coupling a given gRNA group to the gene or protein that the gRNA group targeted. We developed a Nextflow pipeline to apply all methods to analyze the positive control pairs of all datasets.

ChIP-seq enrichment analysis

We obtained ChIP-seq data for CD14+ monocyte cultures with MCSF (10ng/ml) and stimulated with IFN-gamma (100U/ml) for 24 hours (Qiao et al., 2013). We filtered peaks by calling the top 25% by enrichment score. We defined a gene as being targeted by a TF if a peak fell within 5kb upstream or downstream of the TSS. To determine if SCEPTRE's discoveries were consistent with the ChIP-seq data, we computed odds ratios and their corresponding p-values via a Fisher exact test on the contingency table consisting of the downstream genes SCEPTRE found to be associated with the perturbed gene and genes whose promoter regions overlapped with a ChIP-seq peak. We carried out this analysis for the STAT1 and IRF1 genes.

Comparison of SCEPTRE (low-MOI) and SCEPTRE (high-MOI)

We compare and contrast SCEPTRE (low-MOI; as proposed in this Chapter) and SCEPTRE (high-MOI; as proposed in Chapter 2 on high-MOI analysis).

First, SCEPTRE (high-MOI) uses the complement set as its control group, as few (if any) cells contain only NT perturbations in the high-MOI setting. SCEPTRE (low-MOI), by contrast, uses the NT cells as its control group. Second, the high-MOI problem suffers stronger confounding by sequencing depth than the low-MOI problem, but the low-MOI problem suffers from stronger sparsity. Due to these differences, we reasoned that permutations rather than conditional resampling would give better calibration in the low-MOI setting. Third, SCEPTRE (low-MOI) uses a new and improved test statistic (i.e., the NB score statistic) compared the original SCEPTRE (high-MOI), which used a less powerful distilled NB Wald statistic. Finally, SCEPTRE (low-MOI) has built into it a number of novel computational accelerations not present in the original SCEPTRE.

Methods not included in the benchmarking analysis

Several methods that recently have been proposed for single-cell CRISPR screen analysis were not included in our benchmarking study. First, guided sparse factor analysis (GSFA; introduced by Zhou et al. 2022) couples factor analysis to differential expression analysis to infer the effects of perturbations on gene modules and individual genes. GSFA is a Bayesian method, returning a posterior inclusion probability instead of a p-value for each test of association. Given that the methods that we studied in this work were frequentist (and thus returned a p-value), we deprioritized GSFA for benchmarking. Next, Normalisr (proposed by Wang 2021) is a method for single-cell differential expression, co-expression, and CRISPR screen analysis. Normalisr non-linearly transforms the gene expression counts to Gaussianity and then models the transformed counts via a linear model. We were unable to locate an example low-MOI single-cell CRISPR screen analysis in the Normalisr Github repository (although gene co-expression, case-control differential expression, and high-MOI CRISPR screen examples are available). Given this, and given the complexity of the Normalisr codebase, we deprioritized Normalisr for benchmarking.

Chapter acknowledgements

We thank Hugh MacMullan and Gavin Burris for extensive support in using the Wharton high performance computing cluster (HPCC). We thank Sophia Lu for preliminary work on benchmarking existing methodologies. We thank Ziang Niu for carrying out analyses related to modeling the SCEPTRE resampling distributions. We thank John Morris for help in designing the computational experiments and providing feedback on an early draft. We thank Stephanie Hicks, Kasper Hansen, and the Hicks and Hansen Labs at Johns Hopkins University for detailed feedback on the exposition and content of the paper. We

thank Chris Frangieh for providing instructions on downloading and processing the Frangieh data and using the MIMOSCA method. Finally, we thank Rahul Satija for clarifying several points about the Papalexi data.

4.6 Additional mathematical details of SCEPTRE (low MOI)

This section contains additional mathematical details for the proposed method.

Derivation of the expression for the GLM score test statistic

We derive the GLM score test statistic for adding a single variable to a fitted GLM model. Let $Y \in \mathbb{R}^n$ be the vector of responses. Let $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$ be the matrix of variables onto which we are regressing Y. Let $X \in \mathbb{R}^n$ be the vector we seek to test for inclusion in the fitted GLM. Let $\boldsymbol{D} = [X \quad \boldsymbol{Z}] \in \mathbb{R}^{n \times p}$ be the augmented design matrix. We model the mean μ_i of Y_i according to

$$g(\mu_i) = \beta_0 + \sum_{i=1}^{p} \beta_i Z_i + \gamma X_i,$$

where $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^T$ and γ are constants, and $g : \mathbb{R} \to \mathbb{R}$ is a link function. Let $\theta = [\beta, \gamma]^T$ denote the vector of parameters. Also, let $\eta_i = g^{-1}(\mu_i)$ denote the linear component of the model, and let $V(\mu_i)$ denote the variance of the response y_i given mean μ_i. Standard GLM results indicate that the score u and Fisher information I of the model evaluated at θ are

$$\begin{cases} u(\theta) = \nabla \mathcal{L}(\theta) = \boldsymbol{D}^T W M (Y - \mu) \\ I(\theta) = -\mathbb{E}(\nabla^2 \mathcal{L}(\theta)) = \boldsymbol{D}^T W \boldsymbol{D}, \end{cases}$$

where

$$W = \operatorname{diag} \left\{ \frac{1}{V(\mu_i)(d\eta_i / d\mu_i)^2} \right\}_{i=1}^{n}$$

is the matrix of "weights" and

$$M = \operatorname{diag} \left\{ \frac{d\eta_i}{d\mu_i} \right\}_{i=1}^{n}$$

is the matrix of derivatives of the linear component with respect to the mean. We can rewrite the score as

$$u(\theta) = \begin{bmatrix} \boldsymbol{Z}^T \\ X^T \end{bmatrix} W M (Y - \mu) = \begin{bmatrix} \boldsymbol{Z}^T W M (Y - \mu) \\ X^T W M (Y - \mu) \end{bmatrix}.$$

Next, we can rewrite the Fisher information as

$$I(\theta) = \begin{bmatrix} \boldsymbol{Z}^T \\ X^T \end{bmatrix} W [\boldsymbol{Z} \quad X] = \begin{bmatrix} \boldsymbol{Z}^T W \boldsymbol{Z} & \boldsymbol{Z}^T W X \\ X^T W \boldsymbol{Z} & X^T W X \end{bmatrix} = \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}.$$

Define I_X by

$$I_X = I_{22} - I_{21} I_{11}^{-1} I_{12}.$$

We have that

$$I_X = X^T W X - X^T W \boldsymbol{Z} (\boldsymbol{Z}^T W \boldsymbol{Z})^{-1} \boldsymbol{Z}^T W X.$$

Let $\hat{\beta}$ be the MLE for β. Next, let u_X denote the component of the score vector corresponding to γ, i.e.,

$$u_X(\theta) = X^T W M (Y - \mu).$$

The value of u_X evaluated at $(\hat{\beta}, 0)$ is

$$u_X(\hat{\beta}, 0) = X^T \hat{W} \hat{M} (Y - \hat{\mu}),$$

where $\hat{W}$, $\hat{M}$, and $\hat{\mu}$ denote W, M, and μ, respectively, evaluated at $(\hat{\beta}, 0)$. Similarly, the value of $I_X(\theta)$ evaluated at $(\hat{\beta}, 0)$ is

$$I(\hat{\beta}, 0) = X^T \hat{W} X - X^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X.$$

Standard asymptotic results indicate that

$$X^T \hat{W} \hat{M} (Y - \hat{\mu}) \xrightarrow{d} N(0, X^T \hat{W} X - X^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X).$$

The GLM score test z-score is therefore

$$z_{\text{score}} = \frac{X^T \hat{W} \hat{M} (Y - \hat{\mu})}{\sqrt{X^T \hat{W} X - X^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X}}. \tag{4.4}$$

The GLM score test statistic typically is expressed in a different way in books and papers (e.g., Dunn and Smyth 2018). Let E_2 denote the matrix of residuals after least squares regression of the columns of X onto $\boldsymbol{Z}$, i.e.,

$$E_2 = X - \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X.$$

Furthermore, define $\hat{r} = \hat{M}(y - \hat{\mu})$ as the "working residual" vector. (This vector can be obtained in R via `fit$residuals`, where `fit` is a fitted GLM.) The GLM score test statistic typically is expressed as

$$z_{\text{score}} = \frac{X^T \hat{W} \hat{r}}{\sqrt{E_2^T \hat{W} E_2}}. \tag{4.5}$$

The expressions (4.4) and (4.5) are identical:

$$\begin{aligned} X^T \hat{W} X - X^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X \\ = X^T \hat{W} X - X^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X \\ - X^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X \\ + X^T \boldsymbol{W} \boldsymbol{Z} (\boldsymbol{Z}^T W \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \hat{W} X \\ = (X - \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z} \hat{W} X)^T \hat{W} (X - \boldsymbol{Z} (\boldsymbol{Z}^T \hat{W} \boldsymbol{Z})^{-1} \boldsymbol{Z} \hat{W} X) \\ = E_2^T \hat{W} E_2. \end{aligned}$$

Expression (4.5) is evaluated via QR decomposition in practice. By contrast, we propose to evaluate (4.4) via spectral or Cholesky decomposition, thereby exploiting the sparsity of X.

Inductive without replacement sampling

Consider a gene expression vector $Y \in \mathbb{R}^n$, technical factor matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$, and binary vector of perturbation indicators $X \in \{0,1\}^n$. Applying a permutation test (e.g., SCEPTRE) requires permuting the X perturbation indicator vector B times. We can identify a given perturbation indicator vector with the set of entries within that vector that is nonzero. In other words, we can represent the vector X as the set $S = \{i : X[i] = 1\}$, where $X[i]$ denotes the value of the vector X at position i. Suppose that X has m nonzero entries. Randomly permuting X is equivalent to sampling m elements without replacement (WOR) from the set $\{1, \ldots, n\}$. Unfortunately, drawing B WOR samples for every perturbation-gene pair — which is required for carrying out the permutation test — is slow. We derive a strategy for sharing WOR samples across all gene-perturbation pairs, even those containing different numbers of cells.

A key observation is that the number of control cells is fixed across all hypotheses, while the number of treatment cells — i.e., the number of cells containing the targeting perturbation — varies from hypothesis to hypothesis (depending on the targeting perturbation). We exploit this structure of the problem as follows. Let N be the number of control cells; label the control cells by $c_1, c_2, \ldots, c_N$. Assume that there are p targeting perturbations. For

perturbation $i \in \{1, \ldots, p\}$, let k_i be the number of cells containing perturbation i. Finally, let $M = \max_{i \in \{1,\ldots,p\}} k_i$ be the number of cells containing the perturbation that infects the greatest number of cells. Label the treatment cells by $t_1, \ldots, t_M$. We construct a length-M random sequence $a_1, a_2, \ldots, a_M$ that satisfies the following three properties:

1. $a_i \in \{c_1, \ldots, c_N, t_1, \ldots, t_i\}$ for all $i \in \{1, \ldots, M\}$; that is, the ith element of the sequence is a control cell or one of the first i treatment cells.
2. $a_i \neq a_j$ for $i \neq j$. That is, the elements of the sequence are unique.

Let $A_i := \{a_1, \ldots, a_i\}$ denote the set containing the first i elements of the sequence. (Observe that there are $N + i$ elements that possibly could be in A_i: the N control cells and the first i treatment cells.) We additionally require the following property to hold:

3. $\mathbb{P}(c_1 \in A_i) = \cdots = \mathbb{P}(c_N \in A_i) = \mathbb{P}(t_1 \in A_i) = \cdots = \mathbb{P}(t_i \in A_i) = i/(i + N)$. That is, each of the elements that possibly could be in A_i has an equal chance of being in A_i.

The sequence $a_1, \ldots, a_M$ yields a sequence of increasing sets $A_1 \subset A_2 \subset \cdots \subset A_m$. The first set A_1, which contains a single element, contains the control cells and the first treatment cell with equal probability. The second set A_2, which contains two elements, contains the control cells and each of the first *two* treatment cells with equal probability, and so on. Thus, if a given gene-perturbation pair contains $k \leq M$ treatment cells, then A_k constitutes a valid WOR sample for that pair. Importantly, because the sequence of sets $A_1, \ldots, A_M$ is increasing, we can share the random sample $a_1, \ldots, a_M$ across all gene-perturbation pairs. We call this technique "inductive without replacement sampling" (or "IWOR sampling"). The SCEPTRE (low MOI) algorithm generates a set of B length-M IWOR samples and shares this set of samples across all pairs, thereby reducing the amount of compute associated with carrying out the permutation tests.

An abstract approach for constructing an IWOR sample

We describe an abstract approach for constructing an IWOR sample and prove its correctness. The procedure is inductive.

Step 1. Sample one element from the set $\{c_1, \ldots, c_N, t_1\}$, putting an equal mass of $1/(N+1)$ onto each of the elements. Call the sampled element a_1. Set $A_1 = \{a_1\}$.
Step i **for** $i \in \{2, \ldots, M\}$. Let $B_i := \{c_1, \ldots, c_N, t_1 \ldots, t_{i-1}\} \setminus A_{i-1}$ denote the set of "leftover" elements not sampled in steps $1, \ldots, i-1$. There are N elements in the set B_i. Draw an element at random from the set $B_i \bigcup \{t_i\}$, placing a mass of $\frac{i}{i+N}$ on t_i and a mass of $\left(1 - \frac{i}{i+N}\right)/N$ on each of the elements in B_i. Call the sampled element a_i. Set $A_i = A_{i-1} \cup \{a_i\}$.

The resulting sequence $a_1, \ldots, a_M$ satisfies the three properties listed above, which we now prove.

Proof: It is clear that $a_i \in \{c_1, \ldots, c_N, t_1, \ldots, t_i\}$ for all i and that $a_i \neq a_j$ for $i \neq j$. We focus on proving the third property, which we do inductively.

Base case: Let $i = 1$. Then

$$\mathbb{P}(c_1 \in A_1) = \ldots \mathbb{P}(c_N \in A_1) = \mathbb{P}(t_1 \in A_1) = 1/(1+N).$$

Inductive step: Suppose that

$$\mathbb{P}(c_1 \in A_i) = \cdots = \mathbb{P}(c_N \in A_i) = \mathbb{P}(t_1 \in A_i) = \cdots = \mathbb{P}(t_i \in A_i) = \frac{i}{N+i}.$$

To construct the set A_{i+1}, we sample t_{i+1} with probability $(i+1)/(N+i+1)$ and each element of B_i with probability

$$\frac{1}{N}\left(1 - \frac{i+1}{N+i+1}\right).$$

We call the sampled element a_{i+1}, and we set $A_{i+1} = A_i \cup \{a_{i+1}\}$. Our goal is to show that, for all $u \in \{c_1, \ldots, c_N, t_1, \ldots, t_{i+1}\}$,

$$\mathbb{P}(u \in A_{i+1}) = \frac{i+1}{N+i+1}. \tag{4.6}$$

(That is, for all elements that *could* be in A_{i+1}, each has an equal chance of *actually being* in A_{i+1}.) The equality (4.6) holds for $u = t_{i+1}$ by construction. Next, suppose that $u \in \{c_1, \ldots, c_N, t_1, \ldots, t_i\}$. We compute $\mathbb{P}(u \in A_{i+1})$ using the law of total probability:

$$\begin{aligned}\mathbb{P}(u \in A_{i+1}) = \mathbb{P}(u \in A_{i+1} | u \in A_i)\mathbb{P}(u \in A_i) \\ + \mathbb{P}(u \in A_{i+1} | u \notin A_i)\mathbb{P}(u \notin A_i).\end{aligned} \tag{4.7}$$

We consider first the term $\mathbb{P}(u \in A_{i+1}|u \in A_i)\mathbb{P}(u \in A_i)$. We have that $\mathbb{P}(u \in A_{i+1}|u \in A_i) = 1$, as membership of u in A_i implies membership of u in A_{i+1}. By the inductive hypothesis, $\mathbb{P}(u \in A_i) = i/(N+i)$, implying that

$$\mathbb{P}(u \in A_{i+1}|u \in A_i)\mathbb{P}(u \in A_i) = i/(N+i). \tag{4.8}$$

Next, we consider the term $\mathbb{P}(u \in A_{i+1}|u \notin A_i)\mathbb{P}(u \notin A_i)$. If $u \notin A_i$ (i.e., if u is a "leftover" element at step i), then $u \in B_i$, implying

$$\mathbb{P}(u \in A_{i+1}|u \notin A_i) = \frac{1}{N}\left(1 - \frac{i+1}{N+1+i}\right)$$

by construction. Furthermore, by the inductive hypothesis, $\mathbb{P}(i \notin A_i) = 1 - \mathbb{P}(u \in A_i) = 1 - i/(N+i)$. Thus,

$$\mathbb{P}(u \in A_{i+1}|u \notin A_i)\mathbb{P}(u \notin A_i) = \frac{1}{N}\left(1 - \frac{i+1}{N+1+i}\right)(1 - i/(N+i)). \tag{4.9}$$

Combining (4.7), (4.8), and (4.9),

$$\begin{aligned}
\mathbb{P}(u \in A_{i+1}) &= \frac{i}{N+i} + \frac{1}{N}\left(1 - \frac{i+1}{N+1+i}\right)\left(1 - \frac{i}{N+i}\right) \\
&= \frac{i}{N+i} + \frac{1}{N}\left(\frac{N+1+i}{N+1+i} - \frac{i+1}{N+1+i}\right)\left(\frac{N+i}{N+i} - \frac{i}{N+i}\right) \\
&= \frac{i}{N+i} + \frac{1}{N}\left(\frac{N}{N+1+i}\right)\left(\frac{N}{N+i}\right) = \frac{i}{N+i} + \frac{N}{(N+1+i)(N+i)} \\
&= \frac{i(N+1+i)+N}{(N+i)(N+1+i)} = \frac{iN+i+i^2+N}{(N+1)(N+1+i)} = \frac{i+1}{N+1+i}.
\end{aligned}$$

□

A concrete algorithm for IWOR sampling

We now derive a concrete algorithm for constructing an IWOR sample of size M on the set $\{c_1, \ldots, c_N, t_1, \ldots, t_M\}$. The algorithm requires only a high-quality uniform random number generator, which is readily available in most programming languages. First, we describe a simple algorithm for sampling from the discrete probability distribution with support $\{0, \ldots, N\}$ that places mass $i/(i+N)$ on N and mass $(1/N)[1 - i/(i+N)]$ on $\{0, 1, ..., N-1\}$. (We call this distribution the $\texttt{IWOR}(N, i)$ distribution.) The algorithm, given in Algorithm 5, takes $\mathcal{O}(1)$ time and space.

Next, we provide an algorithm for constructing an IWOR sample of length M on the set of control cells $\{c_1, \ldots, c_N\}$ and treatment cells $\{t_1, \ldots, t_M\}$.

Algorithm 5: Generating a sample from the $\texttt{IWOR}(N, i)$ distribution.

Input N and i
$u \sim U(0, 1)$
$p \leftarrow i/(N + i)$
if $u > 1 - p$ **then**
  $d \leftarrow N$
else
  $d \leftarrow \lfloor uN/(1 - p) \rfloor$ // floor operator
end
return d

Algorithm 6 is a concrete instantiation of the "abstract approach for constructing an IWOR sample," as described in the previous section. The vector r is a vector of length $N + 1$. At step i, the N cells that remain (i.e., those that have not yet been sampled among $\{c_1, ..., c_N, t_1, \ldots, t_{i-1}\}$) are stored in the leftmost N positions of r. The cell t_i is then inserted into the rightmost position of r. A cell is sampled from r via the $\texttt{IWOR}(N, i)$ distribution, which places more mass on the cell in the rightmost position (i.e., t_i). The sampled cell is moved into v. Then, the cell in the rightmost entry of r (i.e., t_i) is moved into the position vacated by the sampled cell. Finally, the cell t_{i+1} is moved into the rightmost position of r. We repeat this process until we have iterated through all of the treatment cells. This algorithm is fast and efficient, taking $\mathcal{O}(M)$ time and $\mathcal{O}(M + N)$ space.

Algorithm 6: Generating an IWOR sample of size M given control cells $c_1, \ldots, c_N$ and treatment cells $t_1, \ldots, t_M$.

Input Control cells $c_1, \ldots, c_N$ and treatment cells $t_1, \ldots, t_M$.
Initialize an empty vector v of size M.
Initialize the vector $r \leftarrow [c_1, c_2, \ldots, c_N, t_1]$.
for $i = 1 \ldots M$ **do**
  $\texttt{pos} \sim \text{IWOR}(N, i)$ // sample a position within r
  $v[i - 1] \leftarrow r[\texttt{pos}]$ // move the element at that position into v
  $r[\texttt{pos}] \leftarrow r[N]$ // move the rightmost entry of r to position $\texttt{pos}$
  $r[N] \leftarrow t_{i+1}$ // update the rightmost entry of r with t_{i+1}
end
return v

SCEPTRE uses a slightly modified, more efficient, and more numerically stable version of Algorithm 6 for constructing the IWOR sample. Suppose there

are p targeting perturbations, with the ith perturbation infecting k_i cells. Let $M = \max_{i \in \{1,\ldots,p\}} k_i$ be the number of cells containing the perturbation that infects the greatest number of cells, and let $m = \min_{i \in \{1,\ldots,p\}} k_i$ be the number of cells containing the perturbation that infects the least number of cells. One can first sample k cells without replacement from the set $c_1, \ldots, c_N, t_1, \ldots, t_k$. Then, one can construct an IWOR sample on the cells that remain from the first step and treatment cells $t_{k+1}, t_{k+2}, \ldots, t_M$. One can use the classical Fisher-Yates shuffle (Ting, 2021) to construct a standard WOR sample in the first step and the IWOR algorithm (6) to construct an IWOR sample in the second step. The resulting algorithm is the the hybrid Fisher-Yates/IWOR sampler (Algorithm 7).

Algorithm 7: Hybrid Fisher-Yates/IWOR sampler

Input Control cells $c_1, \ldots, c_N$, treatment cells $t_1, \ldots, t_M$; the minimum number of cells infected by a given targeting perturbation m.

$x \leftarrow [c_1, \ldots, c_N, t_1, \ldots, t_m]$.
$v \leftarrow \text{vector}(M)$
for $i = 1, \ldots, m$ **do**
  $u \sim \text{Unif}(\{0, 1, \ldots, N + m - i\})$
  $\text{Swap}(x[N + m - i], x[u])$
end
for $i = 0, \ldots, m - 1$ **do**
  $v[i] \leftarrow x[i + N]$
end
$x[N] \leftarrow t_{m+1}$
for $i = m + 1, \ldots, M$ **do**
  `pos` $\sim$ IWOR(N, i)
  $v[i - 1] \leftarrow x[\texttt{pos}]$
  $x[\texttt{pos}] \leftarrow x[N]$
  $x[N] \leftarrow t_{i+1}$
end
return v

Five

Conclusions, future directions, and the next frontier

Single-cell CRISPR screens pose considerable statistical and computational challenges. The broad objective of this thesis was to put single-cell CRIPSR screen analysis onto more solid statistical footing. To this end we conducted extensive empirical and computational analyses to elucidate the most pressing statistical challenges that single-cell CRISPR screen data raise, developed methods to resolve these challenges in theory and in practice, and implemented these methods in efficient and practical software (`katsevich-lab.github.io/sceptre`). Together, these efforts help to bring statistical clarity and rigor to single-cell CRISPR screen analysis.

Future directions

We sketch out two future directions, one statistical and one computational. On the statistical front, we aim to prove several theoretical results about SCEPTRE (low-MOI) and generalize the SCEPTRE (low-MOI) methodology. We have shown in simulation experiments that SCEPTRE (low-MOI) possesses the robustness property of CAMP. Briefly, CAMP implies that SCEPTRE (low-MOI) is valid if one of the following two conditions holds: (i) the GLM underlying SCEPTRE is correctly specified (up to the the conditional first moment) and the sample size is sufficiently large, *or* (ii) the treatment is unconfounded. We have proven this result in the Gaussian linear model case (unpublished report). We believe that we can extend this result to the GLM case, possibly by leveraging recent techniques for deriving asymptotic resampling distributions for conditional randomization tests (Niu et al., 2022). Specifically, we aim to show that the permutation distribution of the GLM score statistic converges to a $N(\mu, \sigma^2)$ distribution in the confounded case (i.e., case (i) above). This result not only would demonstrate validity of the procedure but also justify use of the skew-normal acceleration.

Additionally, we seek to prove a couple power results related to SCEPTRE (low-MOI). The GLM upon which SCEPTRE (low-MOI) relies yields valid inference in the idealized setting of correct model specification and a sufficiently large sample size. First, we aim to show that SCEPTRE (low-MOI) matches the power of the GLM in this idealized setting when the treatment assignments are unbalanced (i.e., when the treatment vector contains relatively few cases and relatively many controls). This is the setting relevant to single-cell CRISPR screens, as within a given perturbation-gene pair, relatively few cells contain a targeting perturbation. (This is true in both low- and high-MOI). Such a result would be fairly striking, as it would establish a "statistical free lunch" phenomenon: SCEPTRE (low-MOI) would match the power of the GLM when GLM-based inference is valid, but SCEPTRE (low-MOI) would be much more robust than the GLM to common calibration threats. Second, we seek to show that SCEPTRE (low-MOI) is asymptotically more powerful than methods that (i) regress the gene expression vector onto the matrix of technical factors, (ii) extract the Pearson or deviance residuals from this regression, and then (iii) conduct a two sample test between the residuals and the treatment vector. We anticipate that we can leverage some version of the Neyman-Pearson lemma to prove this result.

Finally, we hope to extend SCEPTRE (low-MOI) in several methodological directions. We anticipate that the permuting score statistics technique could be extended beyond standard GLMs to penalized GLMs (Friedman et al., 2010), semi-parametric GLMs (that use a spline basis to model the continuous covariates; Wood 2011), and mixed-effects GLMs (Zhou et al., 2020). These extensions would expand the set of problems to which SCEPTRE (low-MOI) could be applied. Second, we anticipate the GLM score statistic upon which SCEPTRE (low-MOI) relies could be used as the test statistic in a conditional randomization test (Candès et al., 2018a); this coupling could give rise to a doubly robust conditional independence testing method. Finally, we hope to apply SCEPTRE (or an extension thereof) to problems beyond single-cell CRISPR screens, such as single-cell eQTL analysis (Yazar et al., 2022), bulk RNA-seq analysis (Love et al., 2014), or rare variant GWAS analysis (Gibson, 2012).

Another future aim is to extend the SCEPTRE software in several directions, thereby improving its performance, statistical robustness, and scalability. SCEPTRE (low-MOI) introduced many improvements not present within the original SCEPTRE (i.e., SCEPTRE high-MOI). Some of these improvements are unique to the low-MOI setting; others, however, likely could be ported into high-MOI, thereby bolstering the performance of the original SCEPTRE. (An example is the adaptive permutation testing scheme that SCEPTRE (low-

MOI) uses.) Second, we have been working over the past couple years on a computational platform, `ondisc`, for out-of-core and distributed computing on large-scale single-cell data. `ondisc` introduces several new algorithms for working efficiently with large, sparse matrices, the relevant data structure for single-cell analysis. We aim to finish the `ondisc` software and unite `ondisc` with SCEPTRE, thereby enabling the seamless deployment of SCEPTRE to analyze massive-scale single-cell CRISPR screen data.

The next frontier of genetics

Genome-wide association studies (GWAS) have played an outsize role in genetics research over the past two decades. GWAS have been successful: over 200,000 genetic variants are now confidently linked to diseases ranging from diabetes to schizophrenia. The vast majority of these variants lie in noncoding regions and remain mechanistically uncharacterized (Lappalainen and MacArthur, 2021), limiting their medical utility. The next frontier in genetics, therefore, is to understand the function of these noncoding variants. This problem is sometimes called the variant-to-function, or V2P, problem. Multiple consortia, including the Impact of Genomic Variation on Function Consortium and the Atlas of Variant Effects, have arisen to attack the V2P problem. New genomic technologies — including CRISPR engineering, single-cell sequencing, and single-cell CRISPR screens — have enabled progress on the V2P problem in recent years. However, to fully unlock their transformative power, these technologies will need to be linked to efficient algorithms and reliable statistical procedures. We hope that the methods, tools, and ideas that we explore within this thesis can make a dent in advancing this effort.

Bibliography

Adamson, B., Norman, T. M., Jost, M., Cho, M. Y., Nuñez, J. K., Chen, Y., Villalta, J. E., Gilbert, L. A., Horlbeck, M. A., Hein, M. Y., Pak, R. A., Gray, A. N., Gross, C. A., Dixit, A., Parnas, O., Regev, A., and Weissman, J. S. (2016). A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell*, 167(7):1867–1882.e21.

Aigner, D. J. (1973). Regression with a binary independent variable subject to errors of observation. *Journal of Econometrics*, 1(1):49–59.

Alda-Catalinas, C., Bredikhin, D., Hernando-Herraez, I., Santos, F., Kubinyecz, O., Eckersley-Maslin, M. A., Stegle, O., and Reik, W. (2020). A Single-Cell Transcriptomics CRISPR-Activation Screen Identifies Epigenetic Regulators of the Zygotic Genome Activation Program. *Cell Systems*, 11(1):25–41.e9.

Andersson, R., Gebhard, C., Miguel-Escalada, I., Hoof, I., Bornholdt, J., Boyd, M., Chen, Y., Zhao, X., Schmidl, C., Suzuki, T., Ntini, E., Arner, E., Valen, E., Li, K., Schwarzfischer, L., Glatz, D., Raithel, J., Lilje, B., Rapin, N., Bagger, F. O., Jørgensen, M., Andersen, P. R., Bertin, N., Rackham, O., Burroughs, A. M., Baillie, J. K., Ishizu, Y., Shimizu, Y., Furuhata, E., Maeda, S., Negishi, Y., Mungall, C. J., Meehan, T. F., Lassmann, T., Itoh, M., Kawaji, H., Kondo, N., Kawai, J., Lennartsson, A., Daub, C. O., Heutink, P., Hume, D. A., Jensen, T. H., Suzuki, H., Hayashizaki, Y., Müller, F., Forrest, A. R., Carninci, P., Rehli, M., and Sandelin, A. (2014). An atlas of active enhancers across human cell types and tissues. *Nature*, 507(7493):455–461.

Ardlie, K. G., Deluca, D. S., Segrè, A. V., Sullivan, T. J., Young, T. R., Gelfand, E. T., Trowbridge, C. A., Maller, J. B., Tukiainen, T., and al. Lek, M. b. (2015). The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science*, 348(6235):648–660.

Barry, T., Katsevich, E., and Roeder, K. (2022). Exponential family measurement error models for single-cell crispr screens. *arXiv preprint arXiv:2201.01879*.

Barry, T., Mason, K., Roeder, K., and Katsevich, E. (2023). Robust differential expression testing for single-cell crispr screens at low multiplicity of infection. *bioRxiv*, pages 2023–05.

Barry, T., Wang, X., Morris, J. A., Roeder, K., and Katsevich, E. (2021a). Sceptre improves calibration and sensitivity in single-cell crispr screen analysis. *Genome Biology*, 22(1):1–19.

Barry, T., Wang, X., Morris, J. A., Roeder, K., and Katsevich, E. (2021b). Sceptre improves calibration and sensitivity in single-cell crispr screen analysis. Github. katsevich-lab.github.io/sceptre/.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(1):289–300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188.

Bock, C., Datlinger, P., Chardon, F., Coelho, M. A., Dong, M. B., Lawson, K. A., Lu, T., Maroc, L., Norman, T. M., Song, B., et al. (2022). High-content crispr screening. *Nature Reviews Methods Primers*, 2(1):1–23.

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420.

Candès, E. et al. (2018a). Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 80(3):551–577.

Candès, E., Fan, Y., Janson, L., and Lv, J. (2018b). Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577.

Carroll, R. J. et al. (2006). *Measurement error in nonlinear models: a modern perspective*. Chapman and Hall/CRC.

Chardon, F. M., McDiarmid, T. A., Page, N. F., Martin, B., Domcke, S., Regalado, S. G., Calderon, D., Starita, L. M., Sanders, S. J., Ahituv, N., et al. (2023). Multiplex, single-cell crispra screening for cell type specific regulatory elements. *bioRxiv*.

Choudhary, S. and Satija, R. (2022). Comparison and evaluation of statistical error models for scrna-seq. *Genome biology*, 23(1):1–20.

Chung, E. and Romano, J. P. (2016). Asymptotically valid and exact permutation tests based on two-sample u-statistics. *Journal of Statistical Planning and Inference*, 168:97–105.

Datlinger, P., Rendeiro, A. F., Boenke, T., Senekowitsch, M., Krausgruber, T., Barreca, D., and Bock, C. (2021). Ultra-high-throughput single-cell rna sequencing and perturbation screening with combinatorial fluidic indexing. *Nature Methods*, 18(6):635–642.

Datlinger, P., Rendeiro, A. F., Schmidl, C., Krausgruber, T., Traxler, P., Klughammer, J., Schuster, L. C., Kuchler, A., Alpar, D., and Bock, C. (2017). Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods*, 14(3):297–301.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., et al. (2016). Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7):1853–1866.

Duan, B., Zhou, C., Zhu, C., Yu, Y., Li, G., Zhang, S., Zhang, C., Ye, X., Ma, H., Qu, S., Zhang, Z., Wang, P., Sun, S., and Liu, Q. (2019). Model-based understanding of single-cell CRISPR screening. *Nature Communications*, 10(1).

Dunham, I., Kundaje, A., Aldred, S. F., Collins, P. J., Davis, C. A., Doyle, F., Epstein, C. B., Frietze, S., Harrow, J., Kaul, R., Khatun, J., Lajoie, B. R., Landt, S. G., Lee, B. K., Pauli, F., Rosenbloom, K. R., Sabo, P., Safi, A., Sanyal, A., Shoresh, N., Simon, J. M., Song, L., Trinklein, N. D., Altshuler, R. C., Birney, E., Brown, J. B., Cheng, C., Djebali, S., Dong, X., Ernst, J., Furey, T. S., Gerstein, M., Giardine, B., Greven, M., Hardison, R. C., Harris, R. S., Herrero, J., Hoffman, M. M., Iyer, S., Kellis, M., Kheradpour, P., Lassmann, T., Li, Q., Lin, X., Marinov, G. K., Merkel, A., Mortazavi, A., Parker, S. C., Reddy, T. E., Rozowsky, J., Schlesinger, F., Thurman, R. E., Wang, J., Ward, L. D., Whitfield, T. W., Wilder, S. P., Wu, W.,

Xi, H. S., Yip, K. Y., Zhuang, J., Bernstein, B. E., Green, E. D., Gunter, C., Snyder, M., Pazin, M. J., Lowdon, R. F., Dillon, L. A., Adams, L. B., Kelly, C. J., Zhang, J., Wexler, J. R., Good, P. J., Feingold, E. A., Crawford, G. E., Dekker, J., Elnitski, L., Farnham, P. J., Giddings, M. C., Gingeras, T. R., Guigó, R., Hubbard, T. J., Kent, W. J., Lieb, J. D., Margulies, E. H., Myers, R. M., Stamatoyannopoulos, J. A., Tenenbaum, S. A., Weng, Z., White, K. P., Wold, B., Yu, Y., Wrobel, J., Risk, B. A., Gunawardena, H. P., Kuiper, H. C., Maier, C. W., Xie, L., Chen, X., Mikkelsen, T. S., Gillespie, S., Goren, A., Ram, O., Zhang, X., Wang, L., Issner, R., Coyne, M. J., Durham, T., Ku, M., Truong, T., Eaton, M. L., Dobin, A., Tanzer, A., Lagarde, J., Lin, W., Xue, C., Williams, B. A., Zaleski, C., Röder, M., Kokocinski, F., Abdelhamid, R. F., Alioto, T., Antoshechkin, I., Baer, M. T., Batut, P., Bell, I., Bell, K., Chakrabortty, S., Chrast, J., Curado, J., Derrien, T., Drenkow, J., Dumais, E., Dumais, J., Duttagupta, R., Fastuca, M., Fejes-Toth, K., Ferreira, P., Foissac, S., Fullwood, M. J., Gao, H., Gonzalez, D., Gordon, A., Howald, C., Jha, S., Johnson, R., Kapranov, P., King, B., Kingswood, C., Li, G., Luo, O. J., Park, E., Preall, J. B., Presaud, K., Ribeca, P., Robyr, D., Ruan, X., Sammeth, M., Sandhu, K. S., Schaeffer, L., See, L. H., Shahab, A., Skancke, J., Suzuki, A. M., Takahashi, H., Tilgner, H., Trout, D., Walters, N., Wang, H., Hayashizaki, Y., Reymond, A., Antonarakis, S. E., Hannon, G. J., Ruan, Y., Carninci, P., Sloan, C. A., Learned, K., Malladi, V. S., Wong, M. C., Barber, G. P., Cline, M. S., Dreszer, T. R., Heitner, S. G., Karolchik, D., Kirkup, V. M., Meyer, L. R., Long, J. C., Maddren, M., Raney, B. J., Grasfeder, L. L., Giresi, P. G., Battenhouse, A., Sheffield, N. C., Showers, K. A., London, D., Bhinge, A. A., Shestak, C., Schaner, M. R., Kim, S. K., Zhang, Z. Z., Mieczkowski, P. A., Mieczkowska, J. O., Liu, Z., McDaniell, R. M., Ni, Y., Rashid, N. U., Kim, M. J., Adar, S., Zhang, Z., Wang, T., Winter, D., Keefe, D., Iyer, V. R., Zheng, M., Wang, P., Gertz, J., Vielmetter, J., Partridge, E. C., Varley, K. E., Gasper, C., Bansal, A., Pepke, S., Jain, P., Amrhein, H., Bowling, K. M., Anaya, M., Cross, M. K., Muratet, M. A., Newberry, K. M., McCue, K., Nesmith, A. S., Fisher-Aylor, K. I., Pusey, B., DeSalvo, G., Parker, S. L., Balasubramanian, S., Davis, N. S., Meadows, S. K., Eggleston, T., Newberry, J. S., Levy, S. E., Absher, D. M., Wong, W. H., Blow, M. J., Visel, A., Pennachio, L. A., Petrykowska, H. M., Abyzov, A., Aken, B., Barrell, D., Barson, G., Berry, A., Bignell, A., Boychenko, V., Bussotti, G., Davidson, C., Despacio-Reyes, G., Diekhans, M., Ezkurdia, I., Frankish, A., Gilbert, J., Gonzalez, J. M., Griffiths, E., Harte, R., Hendrix, D. A., Hunt, T., Jungreis, I., Kay, M., Khurana, E., Leng, J., Lin, M. F., Loveland, J., Lu, Z., Manthravadi, D., Mariotti, M., Mudge, J., Mukherjee, G., Notredame, C., Pei, B., Rodriguez, J. M., Saunders, G., Sboner, A.,

Searle, S., Sisu, C., Snow, C., Steward, C., Tapanari, E., Tress, M. L., Van Baren, M. J., Washietl, S., Wilming, L., Zadissa, A., Zhang, Z., Brent, M., Haussler, D., Valencia, A., Addleman, N., Alexander, R. P., Auerbach, R. K., Balasubramanian, S., Bettinger, K., Bhardwaj, N., Boyle, A. P., Cao, A. R., Cayting, P., Charos, A., Cheng, Y., Eastman, C., Euskirchen, G., Fleming, J. D., Grubert, F., Habegger, L., Hariharan, M., Harmanci, A., Iyengar, S., Jin, V. X., Karczewski, K. J., Kasowski, M., Lacroute, P., Lam, H., Lamarre-Vincent, N., Lian, J., Lindahl-Allen, M., Min, R., Miotto, B., Monahan, H., Moqtaderi, Z., Mu, X. J., O'Geen, H., Ouyang, Z., Patacsil, D., Raha, D., Ramirez, L., Reed, B., Shi, M., Slifer, T., Witt, H., Wu, L., Xu, X., Yan, K. K., Yang, X., Struhl, K., Weissman, S. M., Penalva, L. O., Karmakar, S., Bhanvadia, R. R., Choudhury, A., Domanus, M., Ma, L., Moran, J., Victorsen, A., Auer, T., Centanin, L., Eichenlaub, M., Gruhl, F., Heermann, S., Hoeckendorf, B., Inoue, D., Kellner, T., Kirchmaier, S., Mueller, C., Reinhardt, R., Schertel, L., Schneider, S., Sinn, R., Wittbrodt, B., Wittbrodt, J., Jain, G., Balasundaram, G., Bates, D. L., Byron, R., Canfield, T. K., Diegel, M. J., Dunn, D., Ebersol, A. K., Frum, T., Garg, K., Gist, E., Hansen, R. S., Boatman, L., Haugen, E., Humbert, R., Johnson, A. K., Johnson, E. M., Kutyavin, T. V., Lee, K., Lotakis, D., Maurano, M. T., Neph, S. J., Neri, F. V., Nguyen, E. D., Qu, H., Reynolds, A. P., Roach, V., Rynes, E., Sanchez, M. E., Sandstrom, R. S., Shafer, A. O., Stergachis, A. B., Thomas, S., Vernot, B., Vierstra, J., Vong, S., Wang, H., Weaver, M. A., Yan, Y., Zhang, M., Akey, J. M., Bender, M., Dorschner, M. O., Groudine, M., MacCoss, M. J., Navas, P., Stamatoyannopoulos, G., Beal, K., Brazma, A., Flicek, P., Johnson, N., Lukk, M., Luscombe, N. M., Sobral, D., Vaquerizas, J. M., Batzoglou, S., Sidow, A., Hussami, N., Kyriazopoulou-Panagiotopoulou, S., Libbrecht, M. W., Schaub, M. A., Miller, W., Bickel, P. J., Banfai, B., Boley, N. P., Huang, H., Li, J. J., Noble, W. S., Bilmes, J. A., Buske, O. J., Sahu, A. D., Kharchenko, P. V., Park, P. J., Baker, D., Taylor, J., and Lochovsky, L. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74.

Dunn, P. K. and Smyth, G. K. (2018). *Generalized linear models with examples in R*, chapter 7, pages 286–287. Springer.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless r and c++ integration. *Journal of statistical software*, 40:1–18.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., Slichter, C. K., Miller, H. W., McElrath, M. J., Prlic, M., et al. (2015). Mast: a flexible statistical framework for assessing transcriptional changes

and characterizing heterogeneity in single-cell rna sequencing data. *Genome Biology*, 16(1):1–13.

Finner, H. and Roters, M. (2001). On the false discovery rate and expected type I errors. *Biometrical Journal*, 43(8):985–1005.

Finucane, H. K., Bulik-Sullivan, B., Gusev, A., Trynka, G., Reshef, Y., Loh, P.-R., Anttila, V., Xu, H., Zang, C., Farh, K., Ripke, S., Day, F. R., ReproGen Consortium, Schizophrenia Working Group of the Psychiatric Genomics Consortium, RACI Consortium, Purcell, S., Stahl, E., Lindstrom, S., Perry, J. R. B., Okada, Y., Raychaudhuri, S., Daly, M. J., Patterson, N., Neale, B. M., and Price, A. L. (2015). Partitioning heritability by functional annotation using genome-wide association summary statistics. *Nat Genet*, 47(11):1228–35.

Fishilevich, S., Nudel, R., Rappaport, N., Hadar, R., Plaschkes, I., Iny Stein, T., Rosen, N., Kohn, A., Twik, M., Safran, M., Lancet, D., and Cohen, D. (2017). GeneHancer: genome-wide integration of enhancers and target genes in GeneCards. *Database : the journal of biological databases and curation*, 2017:1–17.

Fitzpatrick, P. (2009). *Advanced calculus*, volume 5. American Mathematical Soc.

Frangieh, C. J., Melms, J. C., Thakore, P. I., Geiger-Schuller, K. R., Ho, P., Luoma, A. M., Cleary, B., Jerby-Arnon, L., Malu, S., Cuoco, M. S., et al. (2021). Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. *Nature Genetics*, 53(3):332–341.

Frangoul, H., Altshuler, D., Cappellini, M. D., Chen, Y.-S., Domm, J., Eustace, B. K., Foell, J., de la Fuente, J., Grupp, S., Handgretinger, R., et al. (2021). Crispr-cas9 gene editing for sickle cell disease and β-thalassemia. *New England Journal of Medicine*, 384(3):252–260.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Gallagher, M. D. and Chen-Plotkin, A. S. (2018). The Post-GWAS Era: From Association to Function. *American Journal of Human Genetics*, 102(5):717–730.

Gasperini, M., Hill, A. J., McFaline-Figueroa, J. L., Martin, B., Kim, S., Zhang, M. D., Jackson, D., Leith, A., Schreiber, J., Noble, W. S., Trapnell, C., Ahituv, N., and Shendure, J. (2019). A Genome-wide Framework for Mapping Gene Regulation via Cellular Genetic Screens. *Cell*, 176(1-2):377–390.e19.

Gasperini, M., Tome, J. M., and Shendure, J. (2020). Towards a comprehensive catalogue of validated and target-linked human enhancers. *Nature Reviews Genetics*.

Genga, R. M., Kernfeld, E. M., Parsi, K. M., Parsons, T. J., Ziller, M. J., and Maehr, R. (2019). Single-Cell RNA-Sequencing-Based CRISPRi Screening Resolves Molecular Drivers of Early Human Endoderm Development. *Cell Reports*, 27(3):708–718.e10.

Gibson, G. (2012). Rare and common variants: twenty arguments. *Nature Reviews Genetics*, 13(2):135–145.

Grün, B. and Leisch, F. (2008). *Finite Mixtures of Generalized Linear Regression Models*, pages 205–230. Physica-Verlag HD, Heidelberg.

Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20(1):1–15.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck III, W. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zager, M., et al. (2021). Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.

Hill, A. J., McFaline-Figueroa, J. L., Starita, L. M., Gasperini, M. J., Matreyek, K. A., Packer, J., Jackson, D., Shendure, J., and Trapnell, C. (2018). On the design of CRISPR-based single-cell molecular screens. *Nature Methods*, 15(4):271–274.

Ibrahim, J. G. (1990). Incomplete Data in Generalized Linear Models. *Journal of the American Statistical Association*, 85(411):765–769.

Jin, X., Simmons, S. K., Guo, A. X., Shetty, A. S., Ko, M., Nguyen, L., Robinson, E. B., Oyler, P., Curry, N., Deangeli, G., Lodato, S., Levin, J. Z., Regev, A., Zhang, F., and Arlotta, P. (2020). In vivo Perturb-Seq reveals neuronal and glial abnormalities associated with Autism risk genes. *Science*, page 791525.

Lalli, M. A., Avey, D., Dougherty, J. D., Milbrandt, J., and Mitra, R. D. (2020). High-throughput single-cell functional elucidation of neurodevelopmental disease-associated genes reveals convergent mechanisms altering neuronal differentiation. *Genome Research*, 30(9):1317–1331.

Lappalainen, T. and MacArthur, D. G. (2021). From variant to function in human disease genetics. *Science*, 373(6562):1464–1468.

Lause, J., Berens, P., and Kobak, D. (2021). Analytic pearson residuals for normalization of single-cell rna-seq umi data. *Genome Biology*, 22(1):1–20.

Li, Y., Ge, X., Peng, F., Li, W., and Li, J. J. (2022). Exaggerated false positives by popular differential expression methods when analyzing human population samples. *Genome Biology*, 23(1):1–13.

Lin, K. Z., Lei, J., and Roeder, K. (2021). Exponential-Family Embedding With Application to Cell Developmental Trajectories for Single-Cell RNA-Seq Data. *Journal of the American Statistical Association*, 0(0):1–32.

Liscovitch-Brauer, N., Montalbano, A., Deng, J., Méndez-Mancilla, A., Wessels, H.-H., Moss, N. G., Kung, C.-Y., Sookdeo, A., Guo, X., Geller, E., et al. (2021). Profiling the genetic determinants of chromatin accessibility with scalable single-cell crispr screens. *Nature Biotechnology*, 39(10):1270–1277.

Liu, M. et al. (2021). Fast and Powerful Conditional Randomization Testing via Distillation. *Biometrika*, pages 1–25.

Liu, M., Katsevich, E., Ramdas, A., and Janson, L. (2020). Fast and Powerful Conditional Randomization Testing via Distillation. *arXiv*.

Louis, T. A. . (1982). Finding the Observed Information Matrix when Using the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 44(2):226–233.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15(12):1–21.

Maurano, M. T., Humbert, R., Rynes, E., Thurman, R. E., Haugen, E., Wang, H., Reynolds, A. P., Sandstrom, R., Qu, H., Brody, J., Shafer, A., Neri, F., Lee, K., Kutyavin, T., Stehling-Sun, S., Johnson, A. K., Canfield, T. K., Giste, E., Diegel, M., Bates, D., Hansen, R. S., Neph, S., Sabo, P. J., Heimfeld, S., Raubitschek, A., Ziegler, S., Cotsapas, C., Sotoodehnia, N.,

Glass, I., Sunyaev, S. R., Kaul, R., and Stamatoyannopoulos, J. A. (2012). Systematic localization of common disease-associated variation in regulatory dna. *Science*, 337(6099):1190–5.

McCullagh, P. and Nelder, J. A. (1990). Generalized Linear Models, 2nd Edn.

McGinnis, C. S., Patterson, D. M., Winkler, J., Conrad, D. N., Hein, M. Y., Srivastava, V., Hu, J. L., Murrow, L. M., Weissman, J. S., Werb, Z., et al. (2019). Multi-seq: sample multiplexing for single-cell rna sequencing using lipid-tagged indices. *Nature Methods*, 16(7):619–626.

Mimitou, E. P., Cheng, A., Montalbano, A., Hao, S., Stoeckius, M., Legut, M., Roush, T., Herrera, A., Papalexi, E., Ouyang, Z., Satija, R., Sanjana, N. E., Koralov, S. B., and Smibert, P. (2019). Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. *Nature Methods*, 16(5):409–412.

Mimitou, E. P. et al. (2021). *Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells*, volume 39. Springer US.

Min, Y.-L., Bassel-Duby, R., and Olson, E. N. (2019). Crispr correction of duchenne muscular dystrophy. *Annual review of medicine*, 70:239–255.

Morris, J. A., Caragine, C., Daniloski, Z., Domingo, J., Barry, T., Lu, L., Davis, K., Ziosi, M., Glinos, D. A., Hao, S., et al. (2023). Discovery of target genes and pathways at gwas loci by pooled single-cell crispr screens. *Science*, page eadh7699.

Mullins, N., Forstner, A. J., O'Connell, K. S., Coombes, B., Coleman, J. R., Qiao, Z., Als, T. D., Bigdeli, T. B., Børte, S., Bryois, J., et al. (2021). Genome-wide association study of more than 40,000 bipolar disorder cases provides new insights into the underlying biology. *Nature genetics*, 53(6):817–829.

Niu, Z., Chakraborty, A., Dukes, O., and Katsevich, E. (2022). Reconciling model-x and doubly robust approaches to conditional independence testing. *arXiv preprint arXiv:2211.14698*.

Norman, T. M., Horlbeck, M. A., Replogle, J. M., Ge, A. Y., Xu, A., Jost, M., Gilbert, L. A., and Weissman, J. S. (2019). Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793.

Nystrom, N. A., Levine, M. J., Roskies, R. Z., and Scott, J. R. (2015). Bridges: A Uniquely Flexible HPC Resource for New Communities and Data Analytics. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, XSEDE '15, pages 30:1—-30:8, New York, NY, USA. ACM.

Papalexi, E., Mimitou, E. P., Butler, A. W., Foster, S., Bracken, B., Mauck, W. M., Wessels, H.-H., Hao, Y., Yeung, B. Z., Smibert, P., et al. (2021). Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. *Nature Genetics*, 53(3):322–331.

Pierce, S. E., Granja, J. M., and Greenleaf, W. J. (2020). High-throughput single-cell chromatin accessibility crispr screens enable unbiased identification of regulatory networks in cancer. *bioRxiv*.

Przybyla, L. and Gilbert, L. A. (2021). A new era in functional genomics screens. *Nature Reviews Genetics*, pages 1–15.

Qiao, Y., Giannopoulou, E. G., Chan, C. H., ho Park, S., Gong, S., Chen, J., Hu, X., Elemento, O., and Ivashkiv, L. B. (2013). Synergistic activation of inflammatory cytokine genes by interferon-γ-induced chromatin remodeling and toll-like receptor signaling. *Immunity*, 39(3):454–469.

Qiu, X., Hill, A., Packer, J., Lin, D., Ma, Y. A., and Trapnell, C. (2017). Single-cell mRNA quantification and differential analysis with Census. *Nature Methods*, 14(3):309–315.

Rao, S. S., Huntley, M. H., Durand, N. C., Stamenova, E. K., Bochkov, I. D., Robinson, J. T., Sanborn, A. L., Machol, I., Omer, A. D., Lander, E. S., and Aiden, E. L. (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680.

Replogle, J. M., Norman, T. M., Xu, A., Hussmann, J. A., Chen, J., Cogan, J. Z., Meer, E. J., Terry, J. M., Riordan, D. P., Srinivas, N., et al. (2020). Combinatorial single-cell crispr screens by direct guide rna capture and targeted sequencing. *Nature Biotechnology*, 38(8):954–961.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., Wagner, E. J., Adelman, K., Lithwick-Yanai, G., Iremadze, N., Oberstrass, F., Lipson, D., Bonnar, J. L., Jost, M., Norman, T. M., and Weissman, J. S. (2022). Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28.

Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A., Firth, D., and Ripley, M. B. (2013). Package 'mass'. *Cran r*, 538:113–120.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Sarkar, A. and Stephens, M. (2021). Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53(6):770–777.

Savoca, E. (2000). Measurement errors in binary regressors: An application to measuring the effects of specific psychiatric diseases on earnings. *Health Services and Outcomes Research Methodology*, 1(2):149–164.

Schnitzler, G. R., Kang, H., Lee-Kim, V. S., Ma, R. X., Zeng, T., Angom, R. S., Fang, S., Vellarikkal, S. K., Zhou, R., Guo, K., et al. (2022). Mapping the convergence of genes for coronary artery disease onto endothelial cell programs. *bioRxiv*, pages 2022–11.

Schraivogel, D., Gschwind, A. R., Milbank, J. H., Leonce, D. R., Jakob, P., Mathur, L., Korbel, J. O., Merten, C. A., Velten, L., and Steinmetz, L. M. (2020). Targeted perturb-seq enables genome-scale genetic screens in single cells. *Nature Methods*, 17(6):629–635.

Stefanski, L. A. (2000). Measurement Error Models. *Journal of the American Statistical Association*, 95(452):1353–1358.

Tam, V., Patel, N., Turcotte, M., Bossé, Y., Paré, G., and Meyre, D. (2019). Benefits and limitations of genome-wide association studies. *Nature Reviews Genetics*, 20(8):467–484.

Ting, D. (2021). Simple, optimal algorithms for random sampling without replacement. *arXiv preprint arXiv:2104.05091*.

Townes, F. W., Hicks, S. C., Aryee, M. J., and Irizarry, R. A. (2019). Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. *Genome Biology*, 20(1):1–16.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., Roskies, R., Scott, J., and Wilkins-Diehr, N. (2014). XSEDE: Accelerating Scientific Discovery. *Computing in Science & Engineering*, 16(05):62–74.

Trapnell, C. et al. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386.

Tuano, N. K., Beesley, J., Manning, M., Shi, W., Perlaza-Jimenez, L., Malaver-Ortega, L. F., Paynter, J. M., Black, D., Civitarese, A., McCue, K., et al. (2023). Crispr screens identify gene targets at breast cancer risk loci. *Genome biology*, 24(1):1–23.

Uffelmann, E., Huang, Q. Q., Munung, N. S., De Vries, J., Okada, Y., Martin, A. R., Martin, H. C., Lappalainen, T., and Posthuma, D. (2021). Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59.

Ursu, O., Neal, J. T., Shea, E., Thakore, P. I., Jerby-arnon, L., Nguyen, L., Dionne, D., Diaz, C., Bauman, J., Mosaad, M. M., Fagre, C., Lo, A., Mcsharry, M., Giacomelli, A. O., Ly, S. H., Rozenblatt-rosen, O., Hahn, W. C., Aguirre, A. J., Berger, A. H., Regev, A., and Boehm, J. S. (2022). Massively parallel phenotyping of coding variants in cancer with Perturb-seq. *Nature Biotechnology*.

Wang, L. (2021). Single-cell normalization and association testing unifying crispr screen and gene co-expression analyses with normalisr. *Nature Communications*, 12(1):1–13.

Ward, L. D. and Kellis, M. (2012). Evidence of abundant purifying selection in humans for recently acquired regulatory functions. *Science*, 337(6102):1675–8.

Wessels, H.-h., Méndez-mancilla, A., Papalexi, E., William, M., Lu, L., Morris, J. A., Mimitou, E., Smibert, P., Sanjana, N. E., and Satija, R. (2022). Efficient combinatorial targeting of RNA transcripts in single cells with Cas13 RNA Perturb-seq. *Nature Methods*.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36.

Xie, S., Armendariz, D., Zhou, P., Duan, J., and Hon, G. C. (2019a). Global Analysis of Enhancer Targets Reveals Convergent Enhancer-Driven Regulatory Modules. *Cell Reports*, 29(9):2570–2578.e5.

Xie, S., Duan, J., Li, B., Zhou, P., and Hon, G. C. (2017). Multiplexed Engineering and Analysis of Combinatorial Enhancer Activity in Single Cells. *Molecular Cell*, 66:285–299.

Xie, S. et al. (2019b). Global Analysis of Enhancer Targets Reveals Convergent Enhancer-Driven Regulatory Modules. *Cell Reports*, 29(9):2570–2578.e5.

Yang, L., Zhu, Y., Yu, H., Chen, S., Chu, Y., Huang, H., Zhang, J., and Li, W. (2020). Linking genotypes with multiple phenotypes in single-cell CRISPR screens. *Genome Biology*, 21.

Yao, D., Binan, L., Bezne, J., Simonton, B., Freedman, J., Frangieh, C. J., Dey4, K., Geiger-Schuller, K., Eraslan, B., Gusev, A., Regev, A., and Cleary, B. (2023). Compressed Perturb-seq: highly efficient screens for regulatory circuits using random composite perturbations. *bioRxiv*.

Yazar, S., Alquicira-Hernandez, J., Wing, K., Senabouth, A., Gordon, M. G., Andersen, S., Lu, Q., Rowson, A., Taylor, T. R., Clarke, L., et al. (2022). Single-cell eqtl mapping identifies cell type–specific genetic control of autoimmune disease. *Science*, 376(6589):eabf3041.

Yin, H., Xue, W., and Anderson, D. G. (2019). Crispr–cas: a tool for cancer research and therapeutics. *Nature Reviews Clinical Oncology*, 16(5):281–295.

Zamanighomi, M., Jain, S. S., Ito, T., Pal, D., Daley, T. P., and Sellers, W. R. (2019). GEMINI: A variational Bayesian approach to identify genetic interactions from combinatorial CRISPR screens. *Genome Biology*, 20(1):1–10.

Zhang, L. and Janson, L. (2020). Floodgate : inference for model-free variable importance. *arXiv*, pages 1–67.

Zhou, W., Zhao, Z., Nielsen, J. B., Fritsche, L. G., LeFaive, J., Gagliano Taliun, S. A., Bi, W., Gabrielsen, M. E., Daly, M. J., Neale, B. M., et al. (2020). Scalable generalized linear mixed model for region-based association tests in large biobanks and cohorts. *Nature genetics*, 52(6):634–639.

Zhou, Y., Luo, K., Chen, M., and He, X. (2022). A novel bayesian factor analysis method improves detection of genes and biological processes affected by perturbations in single-cell crispr screening. *bioRxiv*.

Zhu, C., Preissl, S., and Ren, B. (2020). Single-cell multimodal omics: the power of many. *Nature Methods*, 17(1):11–14.

A

Supplementary materials for Chapter 2

This section contains supplementary figures for Chapter 2.

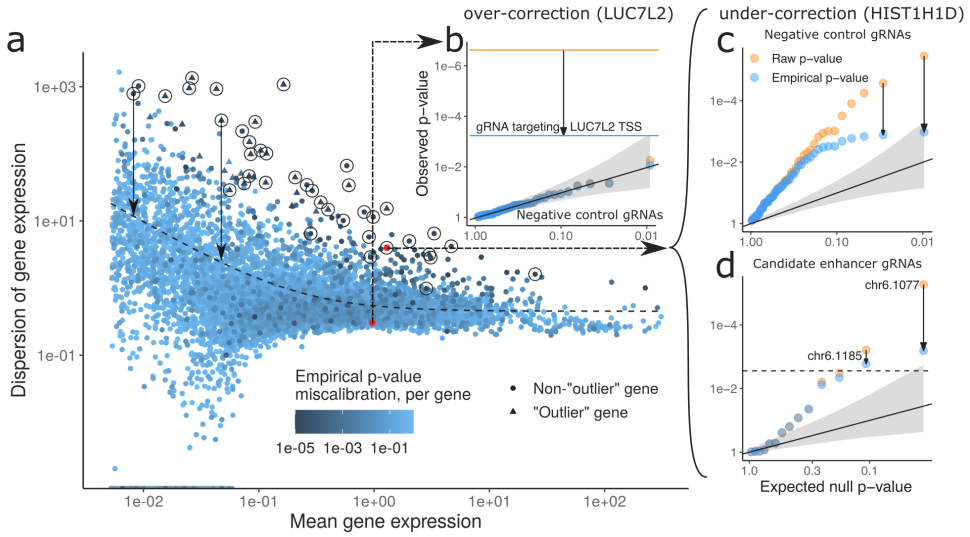


Figure A.1: (Caption on next page.)

Figure A.1: **Gasperini et al.'s empirical correction is insufficient to correct for miscalibration**. **a**, Dispersion estimation procedure employed leads to miscalibration for high-dispersion genes, which the empirical correction does not adequately correct for, as measured by KS test applied to empirical p-values per gene (point colors). **b**, Raw p-values already well-calibrated for *LUC7L2* gene, so empirical correction unnecessarily shrinks the significance of the association with TSS-targeting gRNA, depicted by horizontal lines, by three orders of magnitude. **c**, Empirical correction not strong enough for *HIST1H1D*, which is among circled genes in panel a, which have an NTC-based miscalibration p-value smaller than the Bonferroni threshold. **d**, Under-correction leads to two potential false discoveries for *HIST1H1D*. Dashed horizontal line represents the multiple testing threshold.

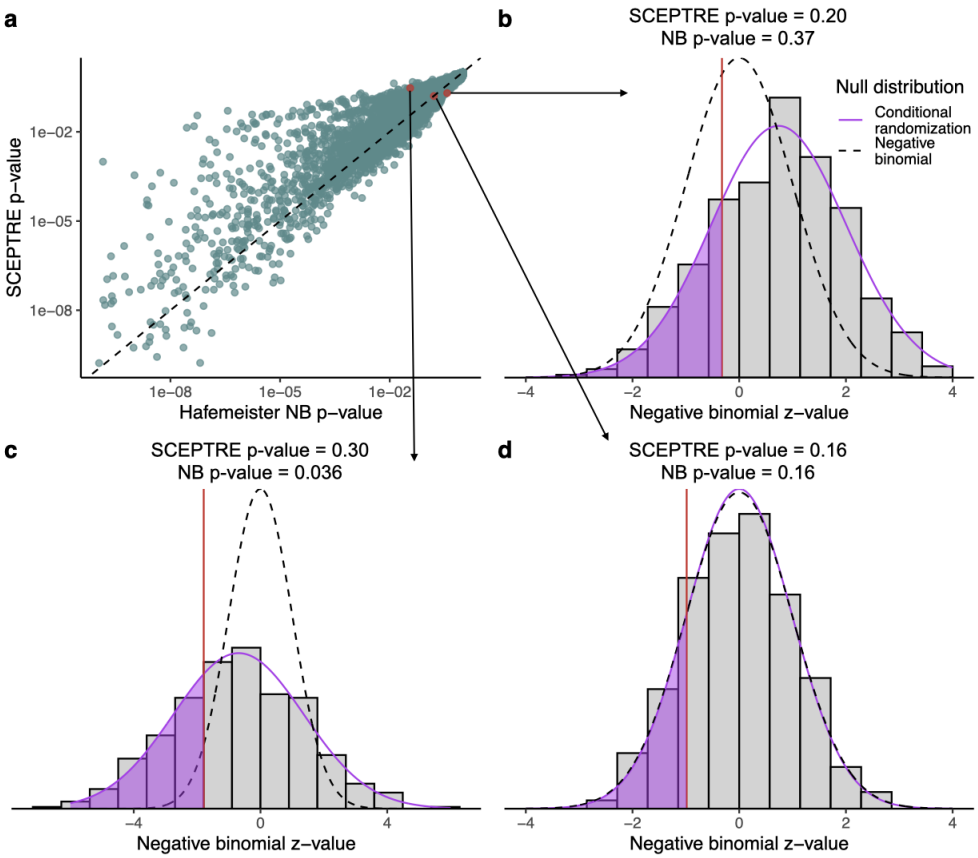


Figure A.2: **Comparison of negative binomial and conditional resampling p-values based on the same test statistic. a**, The standard parametric negative binomial p-value versus that obtained from the same test statistic by conditional resampling, for each gene-enhancer pair (both truncated at 10^{-10} for visualization). The two can diverge fairly substantially. **b-d**, Parametric and conditional resampling null distributions for the negative binomial z-value in three cases: the conditional resampling p-value is more significant (b), the parametric p-value is more significant (c), the two p-values are about the same (d).

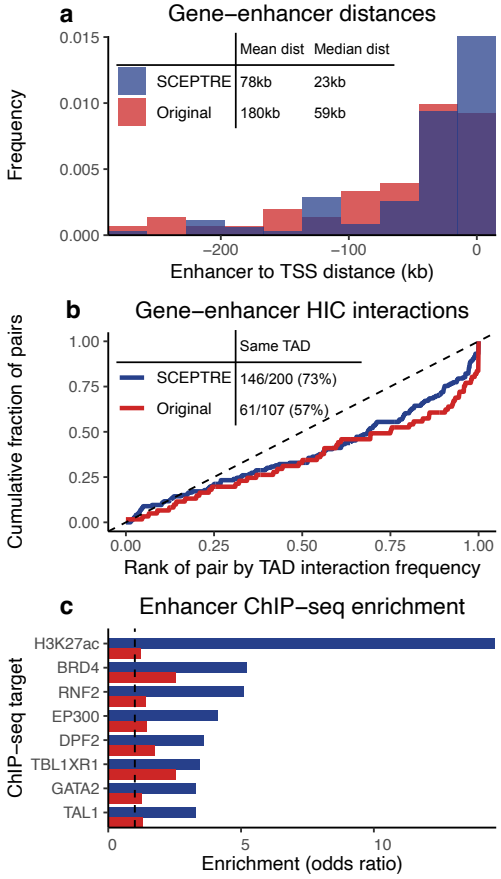


Figure A.3: **Discoveries unique to SCEPTRE on the Gasperini data exhibited greater enrichment for biological signal than those unique to the original method.** This figure shows the 200 discoveries unique to SCEPTRE and the 107 discoveries unique to the original method (in contrast to Figure 2.4, which shows the entire discovery set of both methods). **a,** On average, enhancers discovered by SCEPTRE were less than half the distance to their target genes than those discovered by Gasperini et al. **b,** 73% of the gene-enhancer pairs discovered by SCEPTRE fell within the same TAD, in contrast to 57% of those discovered by the original method. HI-C interaction frequency was similar across methods (though slightly higher for the original), despite the fact that SCEPTRE found 85 more same-TAD pairs. **c,** Enhancers returned by SCEPTRE showed significantly greater enrichment across all ChIP-seq targets, especially H3K27ac.

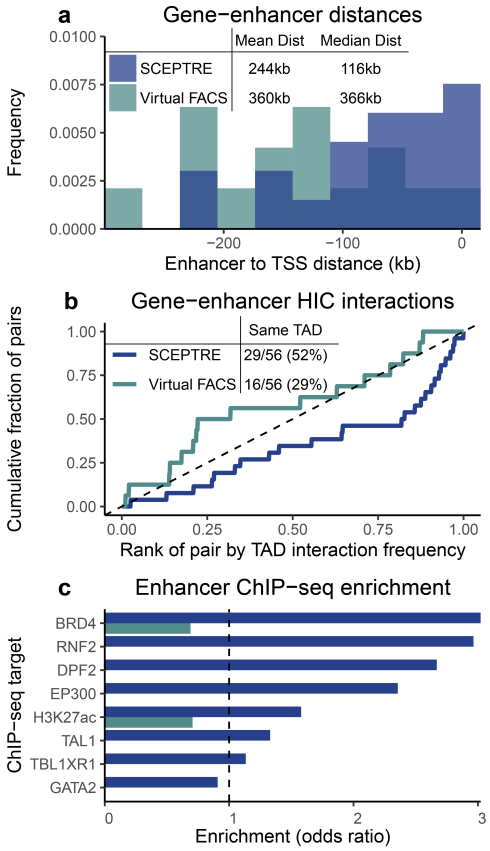


Figure A.4: **SCEPTRE-specific discoveries on the Xie et al. data were more enriched for biological signals of regulation than those specific to the original method.** This Figure is analogous to Figure A.3 but corresponds to the Xie et al. data; only the discoveries unique to SCEPTRE ($n = 56$) and Virtual FACS ($n = 56$) are depicted. **a,** Gene-enhancer links discovered by SCEPTRE were physically closer (median = 116 kb) to one another than those discovered by Virtual FACS (median = 366 kb). **b,** SCEPTRE pairs exhibited higher HI-C interaction frequency and were more likely to fall within the same TAD (52%) than Virtual FACS pairs (29%). **c,** SCEPTRE pairs showed greater enrichment across all eight cell type-relevant ChIP-seq targets. Virtual FACS odds ratios were exactly equal to zero for six of eight targets.

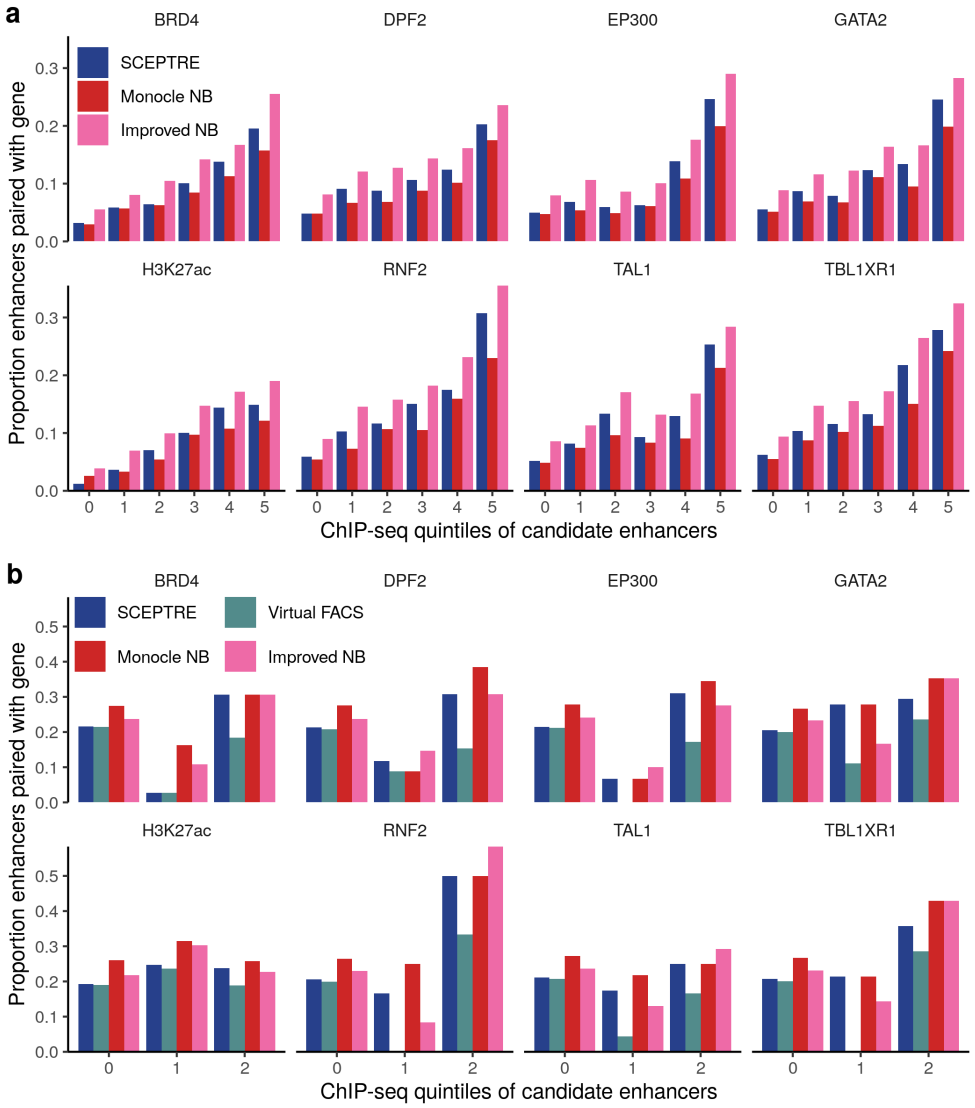


Figure A.5: **Details on ChIP-seq enrichment analysis.** Fraction of candidate enhancers linked to gene, broken down by quantile of ChIP-seq signal for (a) Gasperini et al. data (five quantiles used) and (b) Xie et al. data (two quantiles used). "0" indicates that the candidate enhancer did not overlap a ChIP-seq peak. **a**, On the Gasperini et al. data, methods generally paired candidate enhancers in higher ChIP-seq quantiles more frequently. This enrichment was most pronounced for SCEPTRE across all eight ChIP-seq targets. **b**, Trends were less monotonic on the Xie et al. data, possibly due to the fact that Xie et al. used a different strategy for selecting candidate enhancers.Xie et al. (2019a)

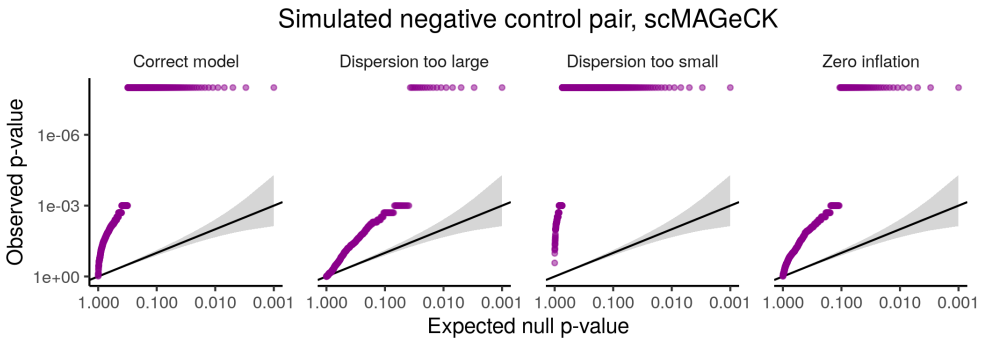


Figure A.6: **scMAGeCK simulation study.** Application of an in-house, specialized version of scMAGeCK to the simulated data. Compared to SCEPTRE (Figure 2.3), scMAGeCK produced inflated p-values.

Quintile	***N* genes**	***N* pairs**	**Mean expression**	**Percent rejected**
1	2108	16253	0.33	0.13%
2	2108	16702	1.5	0.33%
3	2108	16452	3.8	0.62%
4	2108	16692	8.9	0.88%
5	2108	18496	105.0	1.30%

Table A.1: To investigate the impact of gene expression level on the sensitivity of SCEPTRE, we binned candidate *cis* genes into five groups based on their mean expression (i.e., mean number of UMIs per cell). Gene-enhancer pairs consisting of genes in higher expression quintiles were more likely to be rejected than pairs consisting of genes in lower expression quintiles, suggesting that SCEPTRE was better able to detect gene-enhancer links for highly-expressed genes. Results reported for Gasperini et al. data.

Tertile	***N* genes**	***N* pairs**	**Mean expression**	**Percent rejected**
1	249	1867	0.001	1.66%
2	249	1868	0.16	3.32%
3	248	1474	68.2	3.12%

Table A.2: We replicated the analysis reported in table A.1 on the Xie et al. data, binning genes into tertiles rather than quintiles. We observed a similar pattern: genes in the second and third tertile were more likely to be rejected than genes in the first tertile.

B

Supplementary materials for Chapter 3

B.1 Theoretical analysis of the thresholding method

We study the thresholding method from a theoretical perspective, recovering in a simplified Gaussian setting phenomena revealed in the empirical analysis. Suppose we observe gRNA expression and gene expression data $(g_1, m_1), \ldots, (g_n, m_n)$ on n cells from the following linear model:

$$\begin{cases} m_i = \beta_0^m + \beta_1^m p_i + \epsilon_i \\ g_i = \beta_0^g + \beta_1^g p_i + \tau_i \\ p_i \sim \text{Bern}(\pi) \\ \epsilon_i, \tau_i \sim N(0,1), \end{cases} \tag{B.1}$$

where p_i, τ_i, and ϵ_i are independent. For a given threshold $c \in \mathbb{R}$, the imputed perturbation assignment $\hat{p}_i$ is $\hat{p}_i = \mathbb{I}(g_i \geq c)$. The thresholding estimator $\hat{\beta}_1^m$ is the OLS solution, i.e. $\hat{\beta}_1^m = \left[\sum_{i=1}^n (\hat{p}_i - \overline{\hat{p}})^2\right]^{-1} \left[\sum_{i=1}^n (\hat{p}_i - \overline{\hat{p}})(m_i - \overline{m})\right]$. We derive the almost sure limit of $\hat{\beta}_1^m$:

Proposition 1. *The almost sure limit (as $n \to \infty$) of $\hat{\beta}_1^m$ is*

$$\hat{\beta}_1^m \xrightarrow{a.s.} \beta_1^m \left(\frac{\pi(\omega - \mathbb{E}[\hat{p}_i])}{\mathbb{E}[\hat{p}_i](1 - \mathbb{E}[\hat{p}_i])} \right) \equiv \beta_1^m \gamma(\beta_1^g, \pi, c, \beta_0^g), \tag{B.2}$$

where $\mathbb{E}[\hat{p}_i] = \zeta(1-\pi) + \omega\pi$, $\omega \equiv \Phi\left(\beta_1^g + \beta_0^g - c\right)$, and $\zeta \equiv \Phi\left(\beta_0^g - c\right)$.

The function $\gamma : \mathbb{R}^4 \to \mathbb{R}$ does not depend on the gene expression parameters β_1^m or β_0^m. The asymptotic relative bias $b : \mathbb{R}^4 \to \mathbb{R}$ of $\hat{\beta}_1^m$ is given by

$$b(\beta_1^g, \pi, c, \beta_0^g) \equiv \frac{1}{\beta_1^m} \left(\beta_1^m - \lim_{\text{a.s.}} \hat{\beta}_1^m \right) = 1 - \gamma(\beta_1^g, \pi, c, \beta_0^g).$$

Having derived an exact expression for the asymptotic relative bias of $\hat{\beta}_1^m$, we can prove several results about this quantity. We fix π to $1/2$ for simplicity.

(In reality, π is smaller, but the relevant statistical phenomena emerge for $\pi = 1/2$.) We start with informal proposition statements; we follow up with formal proposition statements below. First, the thresholding estimator strictly underestimates (in absolute value) the true value of β_1^m over all choices of the threshold c and over all values of the regression coefficients (β_0^m, β_1^m) and (β_0^g, β_1^g). This phenomenon, called attenuation bias, is a common attribute of estimators that ignore measurement error in errors-in-variables models (Stefanski, 2000). Second, the magnitude of the bias decreases monotonically in β_1^g, comporting with the intuition that the problem becomes easier as the gRNA mixture distribution becomes increasingly well-separated. Third, the Bayes-optimal decision boundary $c_{\text{bayes}} \in \mathbb{R}$ (i.e., the most accurate decision boundary for classifying cells) is a critical value of the bias function. Finally, and most subtly, there is no universally applicable rule for selecting a threshold that yields minimal bias: when β_1^g is small, setting the threshold to an arbitrarily large number yields smaller bias than setting the threshold to the Bayes decision boundary; when β_1^g is large, the reverse is true.

We state state five propositions labeled 2 – 6 corresponding to the informal claims above; these propositions are depicted visually in Figure B.1.

Proposition 2. *Fix $\pi = 1/2$. For all $(\beta_1^g, c, \beta_0^g) \in \mathbb{R}^3$, the asymptotic relative bias is positive, i.e.*

$$b(\beta_1^g, 1/2, c, \beta_0^g) > 0.$$

Proposition 3. *Fix $\pi = 1/2$. The asymptotic relative bias b decreases monotonically in β_1^g, i.e.*

$$\frac{\partial b}{\partial(\beta_1^g)}\,(\beta_1^g, 1/2, c, \beta_0^g) \leq 0.$$

Let c_{bayes} denote the Bayes-optimal decision boundary for classifying cells as perturbed or unperturbed, i.e. $c_{\text{bayes}} = (1/2)(\beta_0^g + \beta_1^g)$ for $\pi = 1/2$. We have that c_{bayes} is a critical value of the bias function:

Proposition 4. *For $\pi = 1/2$ and given $(\beta_1^g, \beta_0^g) \in \mathbb{R}^2$, the Bayes-optimal decision boundary c_{bayes} is a critical value of the bias function b, i.e.*

$$\frac{\partial b}{\partial c}\,(\beta_1^g, 1/2, c_{bayes}, \beta_0^g) = 0.$$

Furthermore, as the threshold tends to infinity, the asymptotic relative bias b tends to π:

Proposition 5. *Assume without loss of generality that $\beta_1^g > 0$. As the threshold c tends to infinity, the asymptotic relative bias b tends to π, i.e.*

$$\lim_{c\to\infty} b(\beta_1^g, \pi, c, \beta_0^g) = \pi.$$

As a corollary, when $\pi = 1/2$, asymptotic relative bias tends to $1/2$ as c tends to infinity. Finally, we compare two threshold selection strategies head-to-head: setting the threshold to an arbitrarily large number, and setting the threshold to the Bayes-optimal decision boundary:

Proposition 6. *Assume without loss of generality that $\beta_1^g > 0$. For $\beta_1^g \in [0, 2\Phi^{-1}(3/4))$, we have that*

$$b(\beta_1^g, 1/2, c_{bayes}, \beta_0^g) > b(\beta_1^g, 1/2, \infty, \beta_0^g).$$

For $\beta_1^g = 2\Phi^{-1}(3/4)$, we have that

$$b(\beta_1^g, 1/2, c_{bayes}, \beta_0^g) = b(\beta_1^g, 1/2, \infty, \beta_0^g).$$

Finally, for $\beta_1^g \in (2\Phi^{-1}(3/4), \infty)$, we have that

$$b(\beta_1^g, 1/2, c_{bayes}, \beta_0^g) < b(\beta_1^g, 1/2, \infty, \beta_0^g).$$

In other words, setting the threshold to a large number yields a smaller bias when β_1^g is small (i.e., $\beta_1^g < 2\Phi^{-1}(3/4) \approx 1.35$; Figure B.2a, left); setting the threshold to the Bayes-optimal decision boundary yields a smaller bias when β_1^g is large (i.e., $\beta_1^g > 2\Phi^{-1}(3/4)$; Figure B.2a, right); and the two approaches coincide when β_1^g is intermediate (i.e., $\beta_1^g = 2\Phi^{-1}(3/4)$; Figure B.2a, middle).

Next, we study the variance of the thresholding estimator, considering a slightly simpler model for this purpose. Suppose the intercepts in (B.1) are fixed at 0 (i.e., $\beta_0^m = \beta_0^g = 0$). For notational simplicity we write $\beta_m = \beta_1^m$ and $\beta_g = \beta_1^g$. The thresholding estimator $\hat{\beta}_m$ is the no-intercept OLS solution $\hat{\beta}_m = \left[\sum_{i=1}^n \hat{p}_i^2\right]^{-1} \left[\sum_{i=1}^n \hat{p}_i m_i\right]$. The following proposition derives the scaled, asymptotic distribution of $\hat{\beta}_m$:

Proposition 7. *The limiting distribution of $\hat{\beta}_m$ is*

$$\sqrt{n}(\hat{\beta}_m - l) \xrightarrow{d} N\left(0, \frac{\beta_m \omega\pi(\beta_m - 2l) + \mathbb{E}[\hat{p}_i](1 + l^2)}{(\mathbb{E}[\hat{p}_i])^2}\right),$$

where

$$l \equiv \beta_m\omega\pi/[\zeta(1-\pi)+\omega\pi]; \quad \mathbb{E}[\hat{p}_i] = \pi\omega+(1-\pi)\zeta; \quad \omega \equiv \Phi(\beta_g - c); \quad \zeta \equiv \Phi(-c).$$

This proposition yields an asymptotically exact bias-variance decomposition for $\hat{\beta}_m$: as the threshold tends to infinity, the bias decreases and the variance increases. Figure B.2 plots the bias-variance decomposition as a function of the threshold.

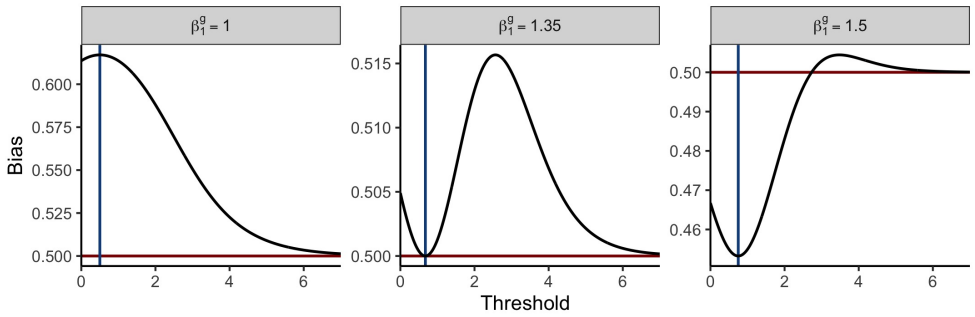


Figure B.1: **Bias as a function of threshold**. This figure visually depicts Propositions 2-6, which were stated informally above. Asymptotic relative bias is plotted on the vertical axis, and the threshold is plotted on the horizontal axis. Panels correspond to different values of β_1^g. Vertical blue lines indicate the Bayes-optimal decision boundary. Observe that (a) bias is strictly nonzero (proposition 2); (b) bias decreases monotonically in β_1^g (Proposition 3); (c) the Bayes-optimal decision boundary is a critical value of the bias function (Proposition 4), in some cases a maximum and in other cases a minumum; (d) as the threshold tends to infinity, the bias converges to 1/2 (Proposition 5); and (e) when $\beta_1^g < 1.35$, an arbitrarily large number yields a smaller bias; by contrast, when $\beta_1^g > 1.35$, the Bayes-optimal decision boundary yields a smaller bias (Proposition 6). Together, these results illustrate that selecting a good threshold is deceptively challenging in the setting of high background contamination.

B.1.1 Organization

The following subsections prove all propositions. Section B.1.2 introduces some notation. Section B.1.3 establishes almost sure convergence of the thresholding estimator in the model (B.1), proving Proposition 1. Section B.1.4 simplifies the expression for the attenuation function γ, and section B.1.5 computes derivatives of γ to be used throughout the proofs. Section B.1.6 establishes the limit in c of γ, proving Proposition 5. Section B.1.7 establishes that the Bayes-optimal decision boundary is a critical value of γ, proving Proposition 4, and section B.1.8 compares the competing threshold selection strategies head-to-head, proving Proposition 6. Section B.1.9 demonstrates that γ is monotone in β_1^g, proving Proposition 3, and Section B.1.10 establishes attenuation bias of the thresholding estimator, proving Proposition 2. Finally, Section B.1.11 derives the bias-variance decomposition of the thresholding estimator in the no-intercept version of B.1, proving Proposition 7.

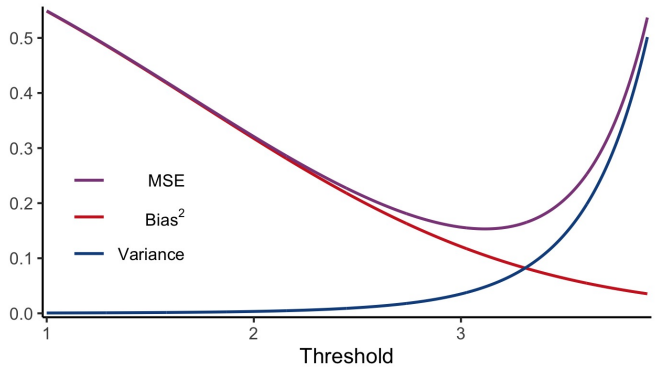


Figure B.2: **Thresholding method bias-variance decomposition.** Bias decreases and variance increases as the threshold tends to infinity. $\beta_1^g = 1, \beta_1^m = 1$, and $\pi = 0.1$ in this plot.

B.1.2 Notation

All notation introduced in this subsection (i.e., B.1.2) pertains to the Gaussian model with intercepts (B.1). Recall that the attenuation function $\gamma : \mathbb{R}^4 \to \mathbb{R}$ is defined by

$$\gamma(\beta_1^g, c, \pi, \beta_0^g) = \frac{\pi(\omega - \mathbb{E}[\hat{p}_i])}{\mathbb{E}[\hat{p}_i](1 - \mathbb{E}[\hat{p}_i])},$$

where

$$\mathbb{E}[\hat{p}_i] = \zeta(1-\pi) + \omega\pi; \quad \omega = \Phi\left(\beta_1^g + \beta_0^g - c\right); \quad \zeta = \Phi\left(\beta_0^g - c\right).$$

Additionally, recall that the asymptotic relative bias function $b : \mathbb{R}^4 \to \mathbb{R}$ is $b(\beta_1^g, c, \pi, \beta_0^g) = 1 - \gamma(\beta_1^g, c, \pi, \beta_0^g)$. Next, we define the functions g and $h : \mathbb{R}^4 \to \mathbb{R}$ by

$$g(\beta_1^g, c, \pi, \beta_0^g) = (1-\pi)\left(\Phi(\beta_0^g + \beta_1^g - c)\right) - (1-\pi)\left(\Phi(\beta_0^g - c)\right) \tag{B.3}$$

and

$$h(\beta_1^g, c, \pi, \beta_0^g) = \left[(1-\pi)\left(\Phi(\beta_0^g - c)\right) + \pi\left(\Phi(\beta_0^g + \beta_1^g - c)\right)\right] \times \left[(1-\pi)\left(\Phi(c - \beta_0^g)\right) + \pi\left(\Phi(c - \beta_0^g - \beta_1^g)\right)\right]. \tag{B.4}$$

We use $f : \mathbb{R} \to \mathbb{R}$ to denote the $N(0,1)$ density, and we denote the right-tail probability probability of f by $\bar{\Phi}$, i.e.,

$$\bar{\Phi}(x) = \int_x^\infty f = \Phi(-x).$$

The parameter β_0^g is a given, fixed constant throughout the proofs. Therefore, to minimize notation, we typically use $\gamma(\beta_1^g, c, \pi)$ (resp., $b(\beta_1^g, c, \pi)$, $g(\beta_1^g, c, \pi)$, $h(\beta_1^g, c, \pi)$) to refer to the function γ (resp., b, g, h) evaluated at $(\beta_1^g, c, \pi, \beta_0^g)$. Finally, for a given function $r : \mathbb{R}^p \to \mathbb{R}$, point $x \in \mathbb{R}^p$, and index $i \in \{1, \ldots, p\}$, we use the symbol $D_i r(x)$ to refer to the derivative of the ith argument of r evaluated at x (*sensu* Fitzpatrick 2009). For example, $D_1\gamma(\beta_1^g, c, 1/2)$ is the derivative of the first argument of γ (the argument corresponding to β_1^g) evaluated at $(\beta_1^g, c, 1/2)$. Likewise, $D_2 g(\beta_1^g, c, \pi)$ is the derivative of the second argument of g (the argument corresponding to c) evaluated at (β_1^g, c, π).

B.1.3 Almost sure limit of $\hat{\beta}_1^m$

We derive the limit in probability of $\hat{\beta}_1^m$ for the Gaussian model with intercepts (B.1). Dividing by n in (B.2), we can express $\hat{\beta}_1^m$ as

$$\hat{\beta}_1^m = \frac{\frac{1}{n}\sum_{i=1}^n (\hat{p}_i - \overline{\hat{p}_i})(m_i - \overline{m})}{\frac{1}{n}\sum_{i=1}^n (\hat{p}_i - \overline{\hat{p}})}.$$

By weak LLN, $\hat{\beta}_1^m \xrightarrow{P} \mathrm{Cov}(\hat{p}_i, m_i)/\mathbb{V}(\hat{p}_i)$. To compute this quantity, we first compute several simpler quantities:

1. Expectation of m_i: $\mathbb{E}[m_i] = \beta_0^m + \beta_1^m \pi$.

2. Expectation of $\hat{p}_i$:

$$\begin{aligned}
&\mathbb{E}[\hat{p}_i] = \mathbb{P}[\hat{p}_i = 1] = \mathbb{P}[\beta_0^g + \beta_1^g p_i + \tau_i \geq c] = \\
&\text{(By LOTP) } \mathbb{P}[\beta_0^g + \tau_i \geq c]\,\mathbb{P}[p_i = 0] + \mathbb{P}[\beta_0^g + \beta_1^g + \tau_i \geq c]\,\mathbb{P}[p_i = 1] \\
&\qquad = \mathbb{P}[\tau_i \geq c - \beta_0^g](1 - \pi) + \mathbb{P}[\tau_i \geq c - \beta_1^g - \beta_0^g](\pi) \\
&\qquad = \left(\bar{\Phi}(c - \beta_0^g)\right)(1 - \pi) + \left(\bar{\Phi}(c - \beta_1^g - \beta_0^g)\right)(\pi) = \\
&\qquad\qquad \Phi(\beta_0^g - c)(1 - \pi) + \Phi(\beta_1^g + \beta_0^g - c)\pi = \zeta(1 - \pi) + \omega\pi.
\end{aligned}$$

3. Expectation of $\hat{p}_i p_i$:

$$\mathbb{E}[\hat{p}_i p_i] \;=\; \mathbb{E}[\hat{p}_i | p_i = 1]\,\mathbb{P}[p_i = 1] \;=\; \mathbb{P}[\beta_0^g + \beta_1^g + \tau_i \geq c]\,\pi \;=\; \omega\pi.$$

4. Expectation of $\hat{p}_i m_i$:

$$\begin{aligned}
\mathbb{E}[\hat{p}_i m_i] &= \mathbb{E}[\hat{p}_i(\beta_0^m + \beta_1^m p_i + \epsilon_i)] = \beta_0^m \mathbb{E}[\hat{p}_i] + \beta_1^m \mathbb{E}[\hat{p}_i p_i] + \mathbb{E}[\hat{p}_i \epsilon_i] \\
&= \beta_0^m \mathbb{E}[\hat{p}_i] + \beta_1^m \omega\pi + \mathbb{E}[\hat{p}_i]\mathbb{E}[\epsilon_i] = \beta_0^m \mathbb{E}[\hat{p}_i] + \beta_1^m \omega\pi.
\end{aligned}$$

5. Variance of $\hat{p}_i$: Because $\hat{p}_i$ is binary, we have that $\mathbb{V}[\hat{p}_i] = \mathbb{E}[\hat{p}_i]\left(1 - \mathbb{E}[\hat{p}_i]\right)$.

6. Covariance of $\hat{p}_i, m_i$:

$$\begin{aligned}\mathrm{Cov}\left(\hat{p}_i, m_i\right) = \mathbb{E}\left[\hat{p}_i m_i\right] - \mathbb{E}[\hat{p}_i]\mathbb{E}[m_i] &= \beta_0^m \mathbb{E}[\hat{p}_i] + \beta_1^m \omega\pi - \mathbb{E}[\hat{p}_i](\beta_0^m + \beta_1^m \pi) \\ &= \beta_1^m \omega\pi - \mathbb{E}[\hat{p}_i]\beta_1^m \pi = \beta_1^m \pi\left(\omega - \mathbb{E}[\hat{p}_i]\right).\end{aligned}$$

Combining these expressions, we have that

$$\hat{\beta}_1^m \xrightarrow{P} \frac{\beta_1^m \pi(\omega - \mathbb{E}[\hat{p}_i])}{\mathbb{E}[\hat{p}_i](1 - \mathbb{E}[\hat{p}_i])} = \beta_1^m \gamma(\beta_1^g, c, \pi).$$

B.1.4 Re-expressing γ in a simpler form

We rewrite the attenuation fraction γ in a way that makes it more amenable to theoretical analysis. We leverage the fact that f integrates to unity and is even. We have that

$$\mathbb{E}\left[\hat{p}_i\right] = (1-\pi)\bar{\Phi}(c - \beta_0^g) + \pi\bar{\Phi}(c - \beta_0^g - \beta_1^g) = (1-\pi)\Phi(\beta_0^g - c) + \pi\Phi(\beta_0^g + \beta_1^g - c), \tag{B.5}$$

and so

$$\begin{aligned}1 - \mathbb{E}\left[\hat{p}_i\right] = (1-\pi) + \pi - \mathbb{E}[\hat{p}_i] &= (1-\pi)\left(1 - \bar{\Phi}(c - \beta_0^g)\right) + \pi\left(1 - \bar{\Phi}(c - \beta_0^g - \beta_1^g)\right) \\ &= (1-\pi)\Phi(c - \beta_0^g) + \pi\Phi(c - \beta_0^g - \beta_1^g). \quad \text{(B.6)}\end{aligned}$$

Next,

$$\omega = \Phi(\beta_1^g + \beta_0^g - c), \tag{B.7}$$

and so

$$\begin{aligned}\omega - \mathbb{E}[\hat{p}_i] = \Phi(\beta_1^g + \beta_0^g - c) - (1-\pi)\Phi(\beta_0^g - c) - \pi\Phi(\beta_0^g + \beta_1^g - c) \\ (1-\pi)\Phi(\beta_1^g + \beta_0^g - c) - (1-\pi)\Phi(\beta_0^g - c). \quad \text{(B.8)}\end{aligned}$$

Combining (B.5, B.6, B.7, B.8), we find that

$$\begin{aligned}\gamma(\beta_1^g, c, \pi) &= \frac{\pi(\omega - \mathbb{E}[\hat{p}_i])}{\mathbb{E}[\hat{p}_i](1 - \mathbb{E}[\hat{p}_i])} \\ &= \frac{\pi\left[(1-\pi)\Phi(\beta_0^g + \beta_1^g - c) - (1-\pi)\Phi(\beta_0^g - c)\right]}{\left[(1-\pi)\Phi(\beta_0^g - c) + \pi\Phi(\beta_0^g + \beta_1^g - c)\right]\left[(1-\pi)\Phi(c - \beta_0^g) + \pi\Phi(c - \beta_0^g - \beta_1^g)\right]}.\end{aligned} \tag{B.9}$$

As a corollary, when $\pi = 1/2$,

$$\gamma(\beta_1^g, c, 1/2) = \frac{\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c)}{[\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)]\,[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)]}. \tag{B.10}$$

Recalling the definitions of g (B.3) and h (B.4), we can write γ as

$$\gamma(\beta_1^g, c, \pi) = \frac{\pi g(\beta_1^g, c, \pi)}{h(\beta_1^g, c, \pi)}.$$

The special case (B.10) is identical to

$$\gamma(\beta_1^g, c, 1/2) = \frac{(4)(1/2)g(\beta_1^g, c, 1/2)}{4h(\beta_1^g, c, 1/2)} = \frac{2g(\beta_1^g, c, 1/2)}{4h(\beta_1^g, c, 1/2)}, \tag{B.11}$$

i.e., the numerator and denominator of (B.11) coincide with those of (B.10). We sometimes will use the notation $2 \cdot g$ and $4 \cdot h$ to refer to the numerator and denominator of (B.10), respectively.

B.1.5 Derivatives of g and h in c

We compute the derivatives of g and h in c, which we will need to prove subsequent results. First, by the fundamental theorem of calculus and the evenness of f, we have that

$$\begin{aligned} D_2 g(\beta_1^g, c, \pi) &= -(1-\pi) f(\beta_0^g + \beta_1^g - c) + (1-\pi) f(\beta_0^g - c) \\ &= (1-\pi) f(c - \beta_0^g) - (1-\pi) f(c - \beta_0^g - \beta_1^g). \end{aligned} \tag{B.12}$$

Second, we have that

$$\begin{aligned} D_2 h(\beta_1^g, c, \pi) &= -[(1-\pi) f(\beta_0^g - c) + \pi f(\beta_0^g + \beta_1^g - c)]\,[(1-\pi)\Phi(c - \beta_0^g) + \pi\Phi(c - \beta_0^g - \beta_1^g)] \\ &+ [(1-\pi) f(c - \beta_0^g) + \pi f(c - \beta_0^g - \beta_1^g)]\,[(1-\pi)\Phi(\beta_0^g - c) + \pi\Phi(\beta_0^g + \beta_1^g - c)] \\ &= [(1-\pi) f(c - \beta_0^g) + \pi f(c - \beta_0^g - \beta_1^g)] \times \\ &\Big[(1-\pi)\Phi(\beta_0^g - c) + \pi\Phi(\beta_0^g + \beta_1^g - c) - (1-\pi)\Phi(c - \beta_0^g) - \pi\Phi(c - \beta_0^g - \beta_1^g)\Big]. \end{aligned} \tag{B.13}$$

B.1.6 Limit of γ in c

Assume (without loss of generality) that $\beta_1^g > 0$. We compute $\lim_{c\to\infty} \gamma(\beta_1^g, c, \pi)$. Observe that

$$\lim_{c\to\infty} g(\beta_1^g, c, \pi) = \lim_{c\to\infty} h(\beta_1^g, c, \pi) = 0.$$

Therefore, we can apply L'Hôpital's rule. We have by (B.12) and (B.13) that

$$\lim_{c\to\infty}\gamma(\beta_1^g,c,\pi)=\lim_{c\to\infty}\frac{\pi D_2 g(\beta_1^g,c,\pi)}{D_2 h(\beta_1^g,c,\pi)}$$
$$=\lim_{c\to\infty}\left\{\frac{(1-\pi)f(c-\beta_0^g)+\pi f(c-\beta_0^g-\beta_1^g)}{\pi(1-\pi)f(c-\beta_0^g)-\pi(1-\pi)f(c-\beta_0^g-\beta_1^g)}\times\right.$$
$$\left.\left[(1-\pi)\Phi(\beta_0^g-c)+\pi\Phi(\beta_0^g+\beta_1^g-c)-(1-\pi)\Phi(c-\beta_0^g)-\pi\Phi(c-\beta_0^g-\beta_1^g)\right]\right\}^{-1}. \tag{B.14}$$

We evaluate the two terms in the product (B.14) separately. Dividing by $f(c-\beta_0^g-\beta_1^g)>0$, we see that

$$\frac{(1-\pi)f(c-\beta_0^g)+\pi f(c-\beta_0^g-\beta_1^g)}{\pi(1-\pi)f(c-\beta_0^g)-\pi(1-\pi)f(c-\beta_0^g-\beta_1^g)}=\frac{\frac{(1-\pi)f(c-\beta_0^g)}{f(c-\beta_0^g-\beta_1^g)}+\pi}{\frac{\pi(1-\pi)f(c-\beta_0^g)}{f(c-\beta_0^g-\beta_1^g)}-\pi(1-\pi)}. \tag{B.15}$$

To evaluate the limit of (B.15), we first evaluate the limit of

$$\frac{f(c-\beta_0^g)}{f(c-\beta_0^g-\beta_1^g)}=\frac{\exp\left[-(1/2)(c-\beta_0^g)^2\right]}{\exp\left[-(1/2)(c-\beta_0^g-\beta_1^g)^2\right]}$$
$$=\frac{\exp[-(1/2)(c^2-2c\beta_0^g+(\beta_0^g)^2)]}{\exp\left[-(1/2)(c^2-2c\beta_0^g-2c\beta_1^g+(\beta_0^g)^2+2(\beta_0^g\beta_1^g)+(\beta_1^g)^2)\right]}$$
$$=\exp\big[-c^2/2+c\beta_0^g-(\beta_0^g)^2/2$$
$$+c^2/2-c\beta_0^g-c\beta_1^g+(\beta_0^g)^2/2+\beta_0^g\beta_1^g+(\beta_1^g)^2/2\big]$$
$$=\exp[-c\beta_1^g+\beta_0^g\beta_1^g+(\beta_1^g)^2/2]=\exp[\beta_0^g\beta_1^g+(\beta_1^g)^2/2]\exp[-c\beta_1^g]. \tag{B.16}$$

Taking the limit in (B.16), we obtain

$$\lim_{c\to\infty}\frac{f(c-\beta_0^g)}{f(c-\beta_0^g-\beta_1^g)}=\exp[\beta_0^g\beta_1^g+(\beta_1^g)^2/2]\lim_{c\to\infty}\exp[-c\beta_1^g]=0$$

for $\beta_1^g>0$. We now can evaluate the limit of (B.15):

$$\lim_{c\to\infty}\frac{(1-\pi)f(c-\beta_0^g)+\pi f(c-\beta_0^g-\beta_1^g)}{\pi(1-\pi)f(c-\beta_0^g)-\pi(1-\pi)f(c-\beta_0^g-\beta_1^g)}=\frac{-\pi}{\pi(1-\pi)}=-\frac{1}{1-\pi}.$$

Next, we compute the limit of the other term in the product (B.14):

$$\lim_{c\to\infty}\Big[(1-\pi)\Phi(\beta_0^g-c)+\pi\Phi(\beta_0^g+\beta_1^g-c) - (1-\pi)\Phi(c-\beta_0^g)-\pi\Phi(c-\beta_0^g-\beta_1^g)\Big]=-(1-\pi)-\pi=-1. \quad \text{(B.17)}$$

Combining (B.15) and (B.17), the limit (B.14) evaluates to

$$\lim_{c\to\infty}\gamma(\beta_1^g,c,\pi)=\left(\frac{1}{1-\pi}\right)^{-1}=1-\pi.$$

It follows that the limit in c of the asymptotic relative bias b is

$$\lim_{c\to\infty} b(\beta_1^g,c,\pi)=1-\lim_{c\to\infty}\gamma(\beta_1^g,c,\pi)=\pi.$$

A corollary is that $\lim_{c\to\infty} b(\beta_1^g,c,1/2)=1/2$.

B.1.7 Bayes-optimal decision boundary as a critical value of γ

Let $c_{\text{bayes}}=\beta_0^g+(1/2)\beta_1^g$. We show that $c=c_{\text{bayes}}$ is a critical value of γ for $\pi=1/2$ and given β_1^g, i.e, $D_2\gamma(\beta_1^g,c_{\text{bayes}},1/2)=0$. Differentiating (B.11), the quotient rule implies that

$$D_2\gamma(\beta_1^g,c,1/2)=\frac{D_2[2g(\beta_1^g,c,1/2)]4h(\beta_1^g,c,1/2)-2g(\beta_1^g,c,1/2)D_2[4h(\beta_1^g,c,1/2)]}{[4h(\beta_1^g,c,\pi)]^2}. \quad \text{(B.18)}$$

We have by (B.12) that

$$D_2[2g(\beta_1^g,c_{\text{bayes}},1/2)]=f(\beta_1^g/2)-f(-\beta_1^g/2)=f(\beta_1^g/2)-f(\beta_1^g/2)=0. \quad \text{(B.19)}$$

Similarly, we have by (B.13) that

$$D_2[4h(\beta_1^g,c_{\text{bayes}},\pi)]=[f(\beta_1^g/2)+f(-\beta_1^g/2)]\cdot [\Phi(-\beta_1^g/2)+\Phi(\beta_1^g/2)-\Phi(\beta_1^g/2)-\Phi(-\beta_1^g/2)]=0. \quad \text{(B.20)}$$

Plugging in (B.20) and (B.19) to (B.18), we find that $D_2[\gamma(\beta_1^g,c_{\text{bayes}},1/2)]=0$. Finally, because

$$b(\beta_1^g,c,1/2)=1-\gamma(\beta_1^g,c,1/2),$$

it follows that

$$D_2[b(\beta_1^g,c_{\text{bayes}},1/2)]=-D_2[\gamma(\beta_1^g,c_{\text{bayes}},1/2)]=0.$$

B.1.8 Comparing Bayes boundary versus large threshold

We compare the bias produced by setting the threshold to a large number to the bias produced by setting the threshold to the Bayes-optimal decision boundary. Let $r : \mathbb{R}^{\geq 0} \to \mathbb{R}$ be the value of attenuation function evaluated at the Bayes-optimal decision boundary $c_{\text{bayes}} = \beta_0^g + (1/2)\beta_1^g$, i.e.

$$r(\beta_1^g) = \gamma(\beta_1^g, \beta_0^g + (1/2)\beta_1^g, 1/2) = \frac{\Phi(\beta_1^g/2) - \Phi(-\beta_1^g/2)}{[\Phi(-\beta_1^g/2) + \Phi(\beta_1^g/2)]\,[\Phi(\beta_1^g/2) + \Phi(-\beta_1^g/2)]}$$

$$= \frac{\int_{-\beta_1^g/2}^{\beta_1^g/2} f}{[1 - \Phi(\beta_1^g/2) + \Phi(\beta_1^g/2)]\,[\Phi(\beta_1^g/2) + 1 - \Phi(\beta_1^g/2)]} = 2\int_0^{\beta_1^g/2} f = 2\Phi(\beta_1^g/2) - 1.$$

We set r to $1/2$ and solve for β_1^g:

$$r(\beta_1^g) = 1/2 \iff 2\Phi(\beta_1^g/2) - 1 = 1/2$$
$$\iff \Phi(\beta_1^g/2) = 3/4 \iff \beta_1^g = 2\Phi^{-1}(3/4) \approx 1.35.$$

Because r is a strictly increasing function, it follows that $r(\beta_1^g) < 1/2$ for $\beta_1^g < 2\Phi^{-1}(3/4)$ and $r(\beta_1^g) > 1/2$ for $\beta_1^g > 2\Phi^{-1}(3/4)$. Next, because

$$b(\beta_1^g, c_{\text{bayes}}, 1/2) = 1 - \gamma(\beta_1^g, c_{\text{bayes}}, 1/2) = 1 - r(\beta_1^g),$$

we have that $b(\beta_1^g, c_{\text{bayes}}, 1/2) > 1/2$ for $\beta_1^g < 2\Phi^{-1}(3/4)$ and $b(\beta_1^g, c_{\text{bayes}}, 1/2) < 1/2$ for $\beta_1^g > 2\Phi^{-1}(3/4)$. Recall that the bias induced by sending the threshold to infinity (as stated in Proposition 5 and proven in Section B.1.6) is $1/2$, i.e.

$$b(\beta_1^g, \infty, 1/2) = 1/2.$$

We conclude that $b(\beta_1^g, c_{\text{bayes}}, 1/2) > b(\beta_1^g, \infty, 1/2)$ on $\beta_1^g \in [0, 2\Phi^{-1}(3/4))$; $b(\beta_1^g, c_{\text{bayes}}, 1/2) = b(\beta_1^g, \infty, 1/2)$ for $\beta_1^g = 2\Phi^{-1}(3/4)$; and $b(\beta_1^g, c_{\text{bayes}}, 1/2) < b(\beta_1^g, \infty, 1/2)$ on $\beta_1^g \in (2\Phi^{-1}(3/4), \infty)$.

B.1.9 Monotonicity in β_1^g

We show that γ is monotonically increasing in β_1^g for $\pi = 1/2$ and given threshold c. We begin by stating and proving two lemmas. The first lemma establishes an inequality that will serve as the basis for the proof.

Lemma B.1. *The following inequality holds:*

$$[\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)] \cdot [\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c) + \Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)]$$
$$\geq [\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c)]\,[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)]\,. \quad \text{(B.21)}$$

Proof: We take cases on the sign on β_1^g.

Case 1: $\beta_g^1 < 0$. Then $\beta_1^g + (\beta^g - c) < (\beta_0^g - c)$, implying $\Phi(\beta_0^g + \beta_1^g - c) < \Phi(\beta_0^g - c)$, or $[\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c)] < 0$. Moreover, $[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)]$ is positive. Therefore, the right-hand side of (B.21) is negative.

Turning our attention of the left-hand side of (B.21), we see that

$$\Phi(\beta_0^g + \beta_1^g - c) + \Phi(c - \beta_0^g - \beta_1^g) = 1 - \Phi(\beta_0^g + \beta_1^g - c) + \Phi(c - \beta_0^g - \beta_1^g) = 1. \quad \text{(B.22)}$$

Additionally, $\Phi(\beta_0^g - c) < 1$ and $\Phi(c - \beta_0^g) > 0$. Combining these facts with (B.22), we find that

$$[\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c) + \Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)] > 0.$$

Finally, because $[\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)] > 0$, the entire left-hand side of (B.21) is positive. The inequality holds for $\beta_1^g < 0$.

Case 2: $\beta_g^1 \geq 0$. We will show that the first term on the LHS of (B.21) is greater than the first term on the RHS of (B.21), and likewise that the second term on the LHS is greater than the second term on the RHS, implying the truth of the inequality. Focusing on the first term, the positivity of $\Phi(\beta_0^g - c)$ implies that $\Phi(\beta_0^g - c) \geq -\Phi(\beta_0^g - c)$, and so

$$\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c) \geq \Phi(\beta_0^g - \beta_1^g - c) - \Phi(\beta_0^g - c).$$

Next, focusing on the second term, $\beta_1^g \geq 0$ implies that

$$\beta_1^g + \beta_0^g - c \geq \beta_0^g - c \implies \Phi(\beta_1^g + \beta_0^g - c) - \Phi(\beta_0^g - c) \geq 0. \quad \text{(B.23)}$$

Adding $\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)$ to both sides of (B.23) yields

$$\Phi(\beta_1^g + \beta_0^g - c) - \Phi(\beta_0^g - c) + \Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g) \geq \Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g).$$

The inequality holds for $\beta_1^g \geq 0$. Combining the cases, the inequality holds for all $\beta_1^g \in \mathbb{R}$. □

The second lemma establishes the derivatives of the functions $2 \cdot g$ and $4 \cdot h$ in β_1^g.

Lemma B.2. *The derivatives in β_1^g of $2 \cdot g$ and $4 \cdot h$ are*

$$D_1[2g(\beta_1^g, c, 1/2)] = f(\beta_0^g + \beta_1^g - c), \quad \text{(B.24)}$$

$$\begin{aligned} D_1[4h(\beta_1^g, c, 1/2)] = {} & f(\beta_0^g + \beta_1^g - c)\,[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)] \\ & - f(\beta_0^g + \beta_1^g - c)\,[\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)]. \quad \text{(B.25)} \end{aligned}$$

Proof: Apply the fundamental theorem of calculus and product rule. □

We are ready to prove the monotonicity of γ in β_1^g. Subtracting

$$[\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)]\,[\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c)]$$

from both sides of (B.21) and multiplying by $f(\beta_0^g + \beta_1^g - c) > 0$ yields

$$\begin{aligned} &f(\beta_0^g + \beta_1^g - c)[\Phi(\beta_0^g - c) + \Phi\left(\beta_0^g + \beta_1^g - c\right)]\,[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)] \\ &\geq f(\beta_0^g + \beta_1^g - c)\,[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)][\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c)] \\ &- f(\beta_0^g + \beta_1^g - c)\,[\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)][\Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c)]. \end{aligned} \tag{B.26}$$

Next, recall that

$$2g(\beta_1^g, c, 1/2) = \Phi(\beta_0^g + \beta_1^g - c) - \Phi(\beta_0^g - c). \tag{B.27}$$

and

$$4h(\beta_1^g, c, 1/2) = [\Phi(\beta_0^g - c) + \Phi(\beta_0^g + \beta_1^g - c)]\,[\Phi(c - \beta_0^g) + \Phi(c - \beta_0^g - \beta_1^g)]. \tag{B.28}$$

Substituting (B.24, B.25, B.27, B.28) into (B.26) produces

$$D_1[2g(\beta_1^g, c, 1/2)]4h(\beta_1^g, c, 1/2) \geq 2g(\beta_1^g, c, 1/2)D_1[4h(\beta_1^g, c, 1/2)],$$

or

$$D_1[2g(\beta_1^g, c, 1/2)]4h(\beta_1^g, c, 1/2) - 2g(\beta_1^g, c, 1/2)D_1[4h(\beta_1^g, c, 1/2)] \geq 0. \tag{B.29}$$

The quotient rule implies that

$$D_1\gamma(\beta_1^g, c, 1/2) = \frac{D_1[2g(\beta_1^g, c, 1/2)]4h(\beta_1^g, c, 1/2) - 2g(\beta_1^g, c, 1/2)D_1[4h(\beta_1^g, c, 1/2)]}{[4h(\beta_1^g, c, 1/2)]^2}. \tag{B.30}$$

We conclude by (B.29) and (B.30) that γ is monotonically increasing in β_1^g. Finally, $b(\beta_1^g, c, \pi) = 1 - \gamma(\beta_1^g, c, \pi)$ is monotonically decreasing in β_1^g.

B.1.10 Strict attenuation bias

We begin by computing the limit of γ in β_1^g given $\pi = 1/2$. First,

$$\begin{aligned} \lim_{\beta_1^g \to \infty} \gamma(\beta_1^g, c, 1/2) &= \frac{1 - \Phi(\beta_0^g - c)}{[1 + \Phi(\beta_0^g - c)]\,[\Phi(c - \beta_0^g)]} \\ &= \frac{\Phi(c - \beta_0^g)}{[1 + \Phi(\beta_0^g - c)]\,[\Phi(c - \beta_0^g)]} = \frac{1}{1 + \Phi(\beta_0^g - c)} < 1. \end{aligned}$$

Similarly,

$$\lim_{\beta_1^g \to -\infty} \gamma(\beta_1^g, c, 1/2) = \frac{-\Phi(\beta_0^g - c)}{[\Phi(\beta_0^g - c)]\,[\Phi(c - \beta_0^g) + 1]} = \frac{-1}{1 + \Phi(c - \beta_0^g)} > -1.$$

The function $\gamma(\beta_1^g, c, 1/2, \beta_0^g)$ is monotonically increasing in β_1^g (as stated in Proposition 3 and proven in section B.1.9). It follows that

$$-1 < -\frac{1}{1 + \Phi(c - \beta_0^g)} \leq \gamma(\beta_1^g, c, 1/2, \beta_0^g) \leq \frac{1}{1 - \Phi(\beta_0^g - c)} < 1$$

for all $\beta_1^g \in \mathbb{R}$. But β_0^g and c were chosen arbitrarily, and so

$$-1 < \gamma(\beta_1^g, c, 1/2, \beta_0^g) < 1$$

for all $(\beta_1^g, c, \beta_0^g) \in \mathbb{R}^3$. Finally, because $b(\beta_1^g, c, 1/2, \beta_0^g) = 1 - \gamma(\beta_1^g, c, 1/2, \beta_0^g)$, it follows that

$$0 < b(\beta_1^g, c, 1/2, \beta_0^g) < 2$$

for all $(\beta_1^g, c, \beta_0^g) \in \mathbb{R}^3$

B.1.11 Bias-variance decomposition in no-intercept model

We prove the bias-variance decomposition for the no-intercept version of (B.1). Define l (for "limit") by

$$l = \beta_m \left(\frac{\omega\pi}{\zeta(1-\pi) + \omega\pi} \right),$$

where

$$\omega = \bar{\Phi}(c - \beta_g) = \Phi(\beta_g - c); \quad \zeta = \bar{\Phi}(c) = \Phi(-c).$$

We have that

$$\hat{\beta}_m - l = \frac{\sum_{i=1}^n \hat{p}_i m_i}{\sum_{i=1}^n \hat{p}_i^2} - l = \frac{\sum_{i=1}^n \hat{p}_i m_i}{\sum_{i=1}^n \hat{p}_i^2} - \frac{l \sum_{i=1}^n \hat{p}_i^2}{\sum_{i=1}^n \hat{p}_i^2} = \frac{\sum_{i=1}^n \hat{p}_i (m_i - l\hat{p}_i)}{\sum_{i=1}^n \hat{p}_i^2}.$$

Therefore,

$$\sqrt{n}(\hat{\beta}_m - l) = \frac{(1/\sqrt{n}) \sum_{i=1}^n \hat{p}_i (m_i - l\hat{p}_i)}{(1/n) \sum_{i=1}^n \hat{p}_i^2}. \tag{B.31}$$

Next, we compute the expectation and variance of $\hat{p}_i(m_i - l\hat{p}_i)$. To do so, we first compute several simpler quantities:

1. Expectation of $\hat{p}_i$: $\mathbb{E}[\hat{p}_i] = \mathbb{P}(p_i\beta_g + \tau_i \geq c) = \mathbb{P}(\beta_g + \tau_i \geq c)\pi + \mathbb{P}(\tau_i \geq c)(1 - \pi) = \pi\omega + (1 - \pi)\zeta$.

2. Expectation of $\hat{p}_i p_i$: $\mathbb{E}\left[\hat{p}_i p_i\right] = \mathbb{E}\left[\hat{p}_i | p_i = 1\right] \mathbb{P}\left[p_i = 1\right] = \omega\pi$.

3. Expectation of $\hat{p}_i m_i$:

$$\begin{aligned}\mathbb{E}[\hat{p}_i m_i] = \mathbb{E}\left[\hat{p}_i(\beta_m p_i + \epsilon_i)\right] &= \mathbb{E}\left[\beta_m \hat{p}_i p_i + \hat{p}_i \epsilon_i\right] \\ &= \beta_m \mathbb{E}\left[\hat{p}_i p_i\right] + \mathbb{E}[\hat{p}_i]\mathbb{E}[\epsilon_i] = \beta_m \omega\pi + 0 = \beta_m \omega\pi.\end{aligned}$$

4. Expectation of $\hat{p}_i m_i^2$:

$$\begin{aligned}&\mathbb{E}\left[\hat{p}_i m_i^2\right] = \mathbb{E}\left[\hat{p}_i(\beta_m p_i + \epsilon_i)^2\right] = \mathbb{E}\left[\hat{p}_i\left(\beta_m^2 p_i^2 + 2\beta_m p_i \epsilon_i + \epsilon_i^2\right)\right] \\ &= \mathbb{E}\left[\hat{p}_i p_i \beta_m^2 + 2\beta_m p_i \hat{p}_i \epsilon_i + \hat{p}_i \epsilon_i^2\right] = \beta_m^2 \mathbb{E}[\hat{p}_i p_i] + 2\beta_m \mathbb{E}[p_i \hat{p}_i]\mathbb{E}[\epsilon_i] + \mathbb{E}[\hat{p}_i]\mathbb{E}[\epsilon_i^2] \\ &= \beta_m^2 \mathbb{E}[\hat{p}_i p_i] + \mathbb{E}[\hat{p}_i] = \beta_m^2 \omega\pi + \mathbb{E}[\hat{p}_i].\end{aligned}$$

Now, we can compute the expectation and variance of $\hat{p}_i(m_i - l\hat{p}_i)$. First,

$$\mathbb{E}\left[\hat{p}_i(m_i - l\hat{p}_i)\right] = \mathbb{E}[\hat{p}_i m_i] - l\mathbb{E}[\hat{p}_i] = \beta_m \omega\pi - \left(\frac{\beta_m \omega\pi}{\zeta(1-\pi) + \omega\pi}\right)\left[\zeta(1-\pi) + \omega\pi\right] = 0. \tag{B.32}$$

Additionally,

$$\begin{aligned}&\mathbb{V}\left[\hat{p}_i(m_i - l\hat{p}_i)\right] = \mathbb{E}\left[\hat{p}_i^2(m_i - l\hat{p}_i)^2\right] - \left(\mathbb{E}\left[\hat{p}_i(m_i - l\hat{p}_i)\right]\right)^2 \\ &= \mathbb{E}\left[\hat{p}_i m_i^2\right] - 2l\mathbb{E}[m_i \hat{p}_i] + l^2 \mathbb{E}[\hat{p}_i] = \beta_m^2 \omega\pi + \mathbb{E}[\hat{p}_i] - 2l\beta_m \omega\pi + l^2 \mathbb{E}[\hat{p}_i] \\ &= \beta_m \omega\pi(\beta_m - 2l) + \mathbb{E}[\hat{p}_i](1 + l^2). \quad \text{(B.33)}\end{aligned}$$

Therefore, by CLT, (B.32), and (B.33),

$$(1/\sqrt{n}) \sum_{i=1}^{n} \hat{p}_i(m_i - l\hat{p}_i) \xrightarrow{d} N\left(0, \beta_m \omega\pi(\beta_m - 2l) + \mathbb{E}[\hat{p}_i](1 + l^2)\right). \tag{B.34}$$

Next, by weak LLN,

$$(1/n) \sum_{i=1}^{n} \hat{p}_i^2 = (1/n) \sum_{i=1}^{n} \hat{p}_i \xrightarrow{P} \mathbb{E}[\hat{p}_i]. \tag{B.35}$$

Finally, by (B.31), (B.34), (B.35), and Slutsky's Theorem,

$$\sqrt{n}(\hat{\beta}_m - l) \xrightarrow{d} N\left(0, \frac{\beta_m \omega\pi(\beta_m - 2l) + \mathbb{E}[\hat{p}_i](1 + l^2)}{\left(\mathbb{E}[\hat{p}_i]\right)^2}\right).$$

Thus, for large $n \in \mathbb{N}$, we have that

$$\mathbb{E}[\hat{\beta}_m] \approx l; \quad \mathbb{V}[\hat{\beta}_m] \approx \left[\beta_m \omega\pi(\beta_m - 2l) + \mathbb{E}[\hat{p}_i](1 + l^2)\right] / [n\mathbb{E}^2[\hat{p}_i]],$$

completing the bias-variance decomposition.

B.2 Estimation and inference in the GLM-EIV model

Estimation

We estimate the parameters of the GLM-EIV model using an EM algorithm.

E step

The E step entails computing the membership probability of each cell. Let $\theta^{(t)} = (\beta_m^{(t)}, \beta_g^{(t)}, \pi^{(t)})$ be the parameter estimate at the t-th iteration of the algorithm. For $k \in \{0,1\}$, let $[\eta_i^m(k)]^{(t)}$ be the ith canonical parameter at the t-th iteration of the algorithm of the gene expression distribution that results from setting p_i to k, i.e. $[\eta_i^m(k)]^{(t)} \equiv h_m\left(\langle \tilde{x}_i(k), \beta_m^{(t)}\rangle + o_i^m\right)$. Similarly, let $[\eta_i^g(k)]^{(t)}$ be defined by $[\eta_i^g(k)]^{(t)} \equiv h_g\left(\langle \tilde{x}_i(k), \beta_g^{(t)}\rangle + o_i^g\right)$. Next, for $k \in \{0,1\}$, define $\alpha_i^{(t)}(k)$ by

$$\begin{aligned}
\alpha_i^{(t)}(k) &\equiv \mathbb{P}\left(M_i = m_i, G_i = g_i | P_i = k, \theta^{(t)}\right) \\
&= \mathbb{P}\left(M_i = m_i | P_i = k, \theta^{(t)}\right) \mathbb{P}\left(G_i = g_i | P_i = k, \theta^{(t)}\right) \text{ (because } G_i \perp\!\!\!\perp M_i | P_i) \\
&= f_m\left(m_i; [\eta_i^m(k)]^{(t)}\right) f_g\left(g_i; [\eta_i^g(k)]^{(t)}\right).
\end{aligned}$$

Finally, let $\pi^{(t)}(1) \equiv \pi^{(t)} = \mathbb{P}\left(P_i = 1 | \theta^{(t)}\right)$ and $\pi^{(t)}(0) \equiv 1 - \pi^{(t)} = \mathbb{P}\left(P_i = 0 | \theta^{(t)}\right)$. The ith membership probability $T_i^{(t)}(1)$ is

$$\begin{aligned}
T_i^{(t)}(1) &= \mathbb{P}(P_i = 1 | M_i = m_i, G_i = g_i, \theta^{(t)}) = \frac{\pi^{(t)}(1)\alpha_i^{(t)}(1)}{\sum_{k=0}^{1} \pi^{(t)}(k)\alpha_i^{(t)}(k)} \text{ (by Bayes rule)} \\
&= \frac{1}{\frac{\pi^{(t)}(0)\alpha_i(0)}{\pi^{(t)}(1)\alpha_i(1)} + 1} = \frac{1}{\exp\left(\log\left(\frac{\pi^{(t)}(0)\alpha_i(0)}{\pi^{(t)}(1)\alpha_i(1)}\right)\right) + 1} = \frac{1}{\exp\left(q_i^{(t)}\right) + 1}, \quad \text{(B.36)}
\end{aligned}$$

where we set

$$q_i^{(t)} := \log\left(\frac{\pi^{(t)}(0)\alpha_i^{(t)}(0)}{\pi^{(t)}(1)\alpha_i^{(t)}(1)}\right). \quad \text{(B.37)}$$

Next, we have that

$$\begin{aligned}
q_i^{(t)} = &\log\left[\pi^{(t)}(0)\right] + \log\left[f_m\left(m_i; [\eta_i^m(0)]^{(t)}\right)\right] + \log\left[f_g\left(g_i; [\eta_i^g(0)]^{(t)}\right)\right] \\
&- \log\left[\pi^{(t)}(1)\right] - \log\left[f_m\left(m_i; [\eta_i^m(1)]^{(t)}\right)\right] - \log\left[f_g\left(g_i; [\eta_i^g(1)]^{(t)}\right)\right],
\end{aligned}$$

We therefore conclude that $T_i^{(t)} = 1/\left(\exp\left(q_i^{(t)}\right) + 1\right)$, which is easily computable.

M step

The complete-data log-likelihood of the GLM-EIV model is

$$\mathcal{L}(\theta; m, g, p) = \sum_{i=1}^{n} \left[p_i \log(\pi) + (1 - p_i) \log(1 - \pi) \right] + \sum_{i=1}^{n} \log\left(f_m(m_i; \eta_i^m) \right) + \sum_{i=1}^{n} \log\left(f_g(g_i; \eta_i^g) \right). \quad \text{(B.38)}$$

Define $Q(\theta|\theta^{(t)}) = \mathbb{E}_{\left(P|M=m,G=g,\theta^{(t)}\right)} \left[\mathcal{L}(\theta; m, g, p) \right]$. We have that

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{n} \left[T_i^{(t)}(1) \log(\pi) + T_i^{(t)}(0) \log(1 - \pi) \right] + \sum_{k=0}^{1} \sum_{i=1}^{n} T_i^{(t)}(k) \log\left[f_m(m_i; \eta_i^m(k)) \right] + \sum_{k=0}^{1} \sum_{i=1}^{n} T_i^{(t)}(k) \log\left[f_g(g_i; \eta_i^{g,b}(k)) \right]. \quad \text{(B.39)}$$

The three terms of (B.39) are functions of different parameters: the first is a function of π, the second is a function of β_m, and the third is a function of β_g. Therefore, to find the maximizer $\theta^{(t+1)}$ of (B.39), we maximize the three terms separately. Differentiating the first term with respect to π, we find that

$$\frac{\partial}{\partial \pi} \sum_{i=1}^{n} \left[T_i^{(t)}(1) \log(\pi) + T_i^{(t)}(0) \log(1 - \pi) \right] = \frac{\sum_{i=1}^{n} T_i^{(t)}(1)}{\pi} - \frac{\sum_{i=1}^{n} T_i^{(t)}(0)}{1 - \pi}.$$

Setting the derivative equal to 0 and solving for π,

$$\frac{\sum_{i=1}^{n} T_i^{(t)}(1)}{\pi} - \frac{\sum_{i=1}^{n} T_i^{(t)}(0)}{1 - \pi} = 0 \iff \sum_{i=1}^{n} T_i^{(t)}(1) - \pi \sum_{i=1}^{n} T_i^{(t)}(1) = \pi \sum_{i=1}^{n} T_i(0)$$
$$\iff \sum_{i=1}^{n} T_i^{(t)}(1) - \pi \sum_{i=1}^{n} T_i^{(t)}(1) = \pi n - \pi \sum_{i=1}^{n} T_i(1) \iff \pi = \frac{\sum_{i=1}^{n} T_i^{(t)}(1)}{n}.$$

Thus, the maximizer $\pi^{(t+1)}$ of (B.39) in π is $\pi^{(t+1)} = (1/n) \sum_{i=1}^{n} T_i^{(t)}(1)$. Next, define $w^{(t)} = [T_1^{(t)}(0), \ldots, T_n^{(t)}(0), T_1^{(t)}(1), \ldots, T_n^{(t)}(1)]^T \in \mathbb{R}^{2n}$. We can view the second term of (B.39) as the log-likelihood of a GLM – call it $\text{GLM}_m^{(t)}$ – that has exponential family density f_m, link function r_m, responses $[m, m]^T$, offsets $[o^m, o^m]^T$, weights $w^{(t)}$, and design matrix $[\tilde{X}(0)^T, \tilde{X}(1)^T]^T$. Therefore,

the maximizer $\beta_m^{(t+1)}$ of the second term of (B.39) is the maximizer of $\text{GLM}_m^{(t)}$, which we can compute using the iteratively reweighted least squares (IRLS) procedure, as implemented in R's GLM function. Similarly, the maximizer $\beta_g^{(t+1)}$ of the third term of (B.39) is the maximizer of the GLM with exponential family density f_g, link function r_g, responses $[g, g]^T$, offsets $[o^g, o^g]^T$, weights $w^{(t)}$, and design matrix $[\tilde{X}(0)^T, \tilde{X}(1)^T]^T$.

Inference

We derive the asymptotic observed information matrix of the GLM-EIV log likelihood, enabling us to perform inference on the parameters. First, we define some notation. For $i \in \{1, \ldots, n\}$, $j \in \{0, 1\}$, and $\theta = (\pi, \beta_m, \beta_g)$, let $T_i^\theta(j)$ be defined by

$$T_i^\theta(j) = \mathbb{P}_\theta \left(P_i = j | M_i = m_i, G_i = g_i \right).$$

Let the $n \times n$ matrix $T^\theta(j)$ be given by $T^\theta(j) = \text{diag}\left\{ T_1^\theta(j), \ldots, T_n^\theta(j) \right\}$. Next, define the diagonal $n \times n$ matrices Δ^m, $[\Delta']^m$, V^m, and H^m by

$$\begin{cases} \Delta^m = \text{diag}\{h'_m(l_1^m), \ldots, h'_m(l_n^m)\} \\ [\Delta']^m = \text{diag}\{h''_m(l_1^m), \ldots, h''_m(l_n^m)\} \\ V^m = \text{diag}\{\psi''_m(\eta_1^m), \ldots, \psi''_m(\eta_n^m)\} \\ H^m = \text{diag}\{m_1 - \mu_1^m, \ldots, m_n - \mu_n^m\}. \end{cases}$$

Define the $n \times n$ matrices $\Delta^g, [\Delta']^g, V^g$, and H^g analogously. These matrices are *unobserved*, as they depend on $\{p_1, \ldots, p_n\}$. Next, for $j \in \{0, 1\}$, let the diagonal $n \times n$ matrices $\Delta^m(j), [\Delta']^m(j), V^m(j)$, and $H^m(j)$ be given by

$$\begin{cases} \Delta^m(j) = \text{diag}\{h'_m(l_1^m(j)), \ldots, h'_m(l_n^m(j))\} \\ [\Delta']^m(j) = \text{diag}\{h''_m(l_1^m(j)), \ldots, h''_m(l_n^m(j))\} \\ V^m(j) = \text{diag}\{\psi''_m(\eta_1^m(j)), \ldots, \psi''_m(\eta_n^m(j))\} \\ H^m(j) = \text{diag}\{m_1 - \mu_1^m(j), \ldots, m_n - \mu_n^m(j)\}. \end{cases}$$

Define the matrices $\Delta^g(j)$, $[\Delta']^g(j)$, $V^g(j)$, and $H^g(j)$ analogously. Finally, define the vectors $s^m(j), w^m(j) \in \mathbb{R}^n$ by

$$\begin{cases} s^m(j) = [m_1 - \mu_1^m(j), \ldots, m_n - \mu_n^m(j)]^T \\ w^m(j) = [T_1(0)T_1(1)\Delta_1^m(j)H_1^m(j), \ldots, T_n(0)T_n(1)\Delta_n^m(j)H_n^m(j)]^T, \end{cases}$$

and let the vectors $s^g(j)$ and $w^g(j)$ be defined analogously. The quantities $\Delta^m(j), [\Delta']^m(j), V^m(j)$, $H^m(j)$, $s^m(j)$, $w^m(j)$, $\Delta^g(j), [\Delta']^g(j), V^g(j)$, $H^g(j)$, $s^g(j)$, and $w^g(j)$ are all *observed*.

The observed information matrix $J(\theta; m, g)$ evaluated at $\theta = (\pi, \beta_m, \beta_g)$ is the negative Hessian of the log likelihood (3.4) evaluated at θ, i.e. $J(\theta; m, g) = -\nabla^2 \mathcal{L}(\theta; m, g)$. This quantity, unfortunately, is hard to compute, as the log likelihood (3.4) is a complicated mixture. Louis (1982) showed that $J(\theta; m, g)$ is equivalent to the following quantity:

$$\begin{aligned} J(\theta; m, g) = &-\mathbb{E}\left[\nabla^2 \mathcal{L}(\theta; m, g, p) | G = g, M = m\right] \\ &+ \mathbb{E}\left[\nabla \mathcal{L}(\theta; m, g, p) | G = g, M = m\right] \mathbb{E}\left[\nabla \mathcal{L}(\theta; m, g, p) | G = g, M = m\right]^T \\ &- \mathbb{E}\left[\nabla \mathcal{L}(\theta; m, g, p) \nabla \mathcal{L}(\theta; m, g, p)^T | G = g, M = m\right]. \end{aligned} \quad \text{(B.40)}$$

The observed information matrix $J(\theta; m, g)$ has dimension $(2d + 1) \times (2d + 1)$. Recall that the complete-data log-likelihood (B.38) is the sum of three terms. The first term depends only on π, the second on β_m, and the third on β_g. Therefore, the observed information matrix can be viewed as block matrix consisting of nine submatrices (Figure B.3; only six submatrices labelled). Submatrix I depends on π, submatrix II on β_m, submatrix III on β_g, submatrix IV on β_m and β_g, submatrix V on π and β_m, and submatrix VI on π and β_g. We only need to compute these six submatrices to compute the entire matrix, as the matrix is symmetric. The following sections derive formulas for submatrices I-VI. All expectations are understood to be *conditional* on m and g. The notation ∇_v and ∇_v^2 represent the gradient and Hessian, respectively, with respect to the vector v.

Submatrix I

Denote submatrix I by $J_\pi(\theta; m, g)$. The formula for $J_\pi(\theta; m, g)$ is

$$J_\pi(\theta; m, g) = -\mathbb{E}\left[\nabla_\pi^2 \mathcal{L}(\theta; m, g, p)\right] + \left(\mathbb{E}\left[\nabla_\pi \mathcal{L}(\theta; m, g, p)\right]\right)^2 - \mathbb{E}\left[\left(\nabla_\pi \mathcal{L}(\theta; m, g, p)\right)^2\right]. \quad \text{(B.41)}$$

We begin by calculating the first and second derivatives of the log-likelihood $\mathcal{L}$ with respect to π. The first derivative is

$$\begin{aligned} \nabla_\pi \mathcal{L}(\theta; m, g, p) &= \frac{\partial}{\partial \pi}\left(\sum_{i=1}^n p_i \log(\pi) + \sum_{i=1}^n (1 - p_i) \log(1 - \pi)\right) \\ &= \frac{\sum_{i=1}^n p_i}{\pi} - \frac{\sum_{i=1}^n (1 - p_i)}{1 - \pi} = \frac{\sum_{i=1}^n p_i}{\pi} - \frac{n - \sum_{i=1}^n p_i}{1 - \pi} \\ &= \left(\frac{1}{\pi} + \frac{1}{1 - \pi}\right) \sum_{i=1}^n p_i - \frac{n}{1 - \pi}. \end{aligned} \quad \text{(B.42)}$$

The second derivative is

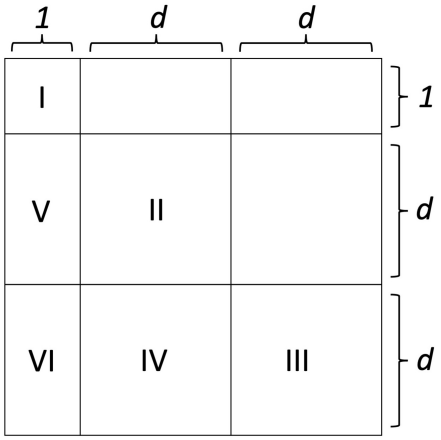


Figure B.3: Block structure of the observed information matrix $J(\theta; m, g) = -\nabla^2 \mathcal{L}(\theta; m, g)$. The matrix is symmetric, and so we only need to compute submatrices I-VI to compute the entire matrix.

$$\nabla^2_\pi \mathcal{L}(\theta; m, g, p) = \frac{\partial^2}{\partial^2 \pi} \left(\frac{\sum_{i=1}^n p_i}{\pi} - \frac{n - \sum_{i=1}^n p_i}{1-\pi} \right) = \frac{(\sum_{i=1}^n p_i) - n}{(1-\pi)^2} - \frac{\sum_{i=1}^n p_i}{\pi^2}.$$

We compute the expectation of the first term of (B.41):

$$\begin{aligned}
\mathbb{E}\left[-\nabla^2_\pi \mathcal{L}(\theta; m, g, p)\right] &= -\mathbb{E}\left[\frac{(\sum_{i=1}^n p_i) - n}{(1-\pi)^2} - \frac{\sum_{i=1}^n p_i}{\pi^2} \right] \\
&= -\mathbb{E}\left\{ \left[\frac{1}{(1-\pi)^2} - \frac{1}{\pi^2} \right] \sum_{i=1}^n p_i - \frac{n}{(1-\pi)^2} \right\} \\
&= -\left\{ \left[\frac{1}{(1-\pi)^2} - \frac{1}{\pi^2} \right] \sum_{i=1}^n T_i^\theta(1) - \frac{n}{(1-\pi)^2} \right\} \\
&= \left[\frac{1}{\pi^2} - \frac{1}{(1-\pi)^2} \right] \sum_{i=1}^n T_i^\theta(1) + \frac{n}{(1-\pi)^2}. \quad \text{(B.43)}
\end{aligned}$$

Next, we compute the difference of the second two pieces of (B.41). To this end, define $a \equiv 1/(1-\pi) + 1/\pi$ and $b \equiv n/(1-\pi)$. We have that

$$\mathbb{E}\left[\nabla_\pi \mathcal{L}(\theta; m, g, p)^2\right] = \mathbb{E}\left[\left(a \sum_{i=1}^n p_i - b \right)^2 \right] = \mathbb{E}\left[a^2 \left(\sum_{i=1}^n p_i \right)^2 - 2ab \sum_{i=1}^n p_i + b^2 \right]$$

$$= a^2 \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[p_i p_j] - 2ab \sum_{i=1}^{n} \mathbb{E}[p_i] + b^2.$$

Next,

$$\left(\mathbb{E}\left[\nabla_\pi \mathcal{L}(\theta; m, g, x)\right]\right)^2 = \left(a \sum_{i=1}^{n} \mathbb{E}[p_i] - b\right)^2 = a^2 \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[p_i]E[p_j] - 2ab \sum_{i=1}^{n} \mathbb{E}[p_i] + b^2.$$

Therefore,

$$\begin{aligned}
&\left(\mathbb{E}[\nabla_\pi \mathcal{L}(\theta; m, g, p)]\right)^2 - \mathbb{E}\left[\nabla_\pi \mathcal{L}(\theta; m, g, p)^2\right] \\
&\quad = a^2 \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[p_i]\mathbb{E}[p_j] - a^2 \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[p_i p_j] = a^2 \left(\sum_{i=1}^{n} \mathbb{E}[p_i]^2 - \mathbb{E}[p_i^2]\right) \\
&= a^2 \left(\sum_{i=1}^{n} [T_i^\theta(1)]^2 - T_i^\theta(1)\right) = \left(\frac{1}{(1-\pi)} + \frac{1}{\pi}\right)^2 \left(\sum_{i=1}^{n} [T_i^\theta(1)]^2 - T_i^\theta(1)\right).
\end{aligned} \tag{B.44}$$

Stringing (B.41), (B.43) and (B.44) together, we obtain

$$\begin{aligned}
J_\pi(\theta; m, g) = &\left[\frac{1}{\pi^2} - \frac{1}{(1-\pi)^2}\right] \sum_{i=1}^{n} T_i^\theta(1) + \frac{n}{(1-\pi)^2} \\
&+ \left(\frac{1}{(1-\pi)} + \frac{1}{\pi}\right)^2 \left(\sum_{i=1}^{n} [T_i^\theta(1)]^2 - T_i^\theta(1)\right).
\end{aligned} \tag{B.45}$$

Submatrix II

Denote submatrix II by $J_{\beta^m}(\theta; m, g)$. The formula for $J_{\beta^m}(\theta; m, g)$ is

$$\begin{aligned}
J_{\beta^m}(\theta; m, g) = &-\mathbb{E}\left[\nabla^2_{\beta^m} \mathcal{L}(\theta; m, g, p)\right] \\
&+ \mathbb{E}\left[\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p)\right] \mathbb{E}\left[\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p)\right]^T \\
&- \mathbb{E}\left[\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p) \nabla_{\beta^m} \mathcal{L}(\theta; m, g, p)^T\right].
\end{aligned} \tag{B.46}$$

Standard GLM results imply that $-\nabla^2_{\beta^m} \mathcal{L}(\theta; m, g, p) = \tilde{X}^T(\Delta^m V^m \Delta^m - [\Delta']^m H^m)\tilde{X}$ and $\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p) = \tilde{X}^T \Delta^m s^m$. We compute the first term of (B.46). The (k, l)th entry of this matrix is

$$\left(\mathbb{E}\left[-\nabla^2_{\beta^m} \mathcal{L}(\theta; m, g, p)\right]\right)[k, l] = \mathbb{E}\left\{\tilde{X}[, k]^T(\Delta^m V^m \Delta^m - [\Delta']^m H^m)\tilde{X}[, l]\right\}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} \mathbb{E}\left\{\tilde{x}_{i,k}(\Delta_i^m V_i^m \Delta_i^m - [\Delta']_i^m H_i^m)\tilde{x}_{i,l}\right\} \\
&= \sum_{i=1}^{n} \tilde{x}_{i,k}(0) T_i^\theta(0)[\Delta_i^m(0) V_i^m(0) \Delta_i^m(0) - [\Delta']_i^m(0) H_i^m(0)]\tilde{x}_{i,l}(0) \\
&+ \sum_{i=1}^{n} \tilde{x}_{i,k}(1) T_i^\theta(1)[\Delta_i^m(1) V_i^m(1) \Delta_i^m(1) - [\Delta']_i^m(1) H_i^m(1)]\tilde{x}_{i,l}(1) \\
&= \sum_{s=0}^{1} \tilde{X}(s)[,k]^T T^\theta(s) \left[\Delta^m(s) V^m(s) \Delta^m(s) - [\Delta']^m(s) H^m(s)\right] \tilde{X}(s)[,l].
\end{aligned}
$$

We therefore have that

$$
\mathbb{E}\left[-\nabla^2_{\beta^m} \mathcal{L}(\theta; m, g, p)\right] = \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s) \left[\Delta^m(s) V^m(s) \Delta^m(s) - [\Delta']^m(s) H^m(s)\right] \tilde{X}(s). \tag{B.47}
$$

Next, we compute the difference of the last two terms of (B.46). The (k,l)th entry is

$$
\begin{aligned}
&\Big[\mathbb{E}\left[\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p)\right] \mathbb{E}\left[\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p)\right]^T \\
&\qquad - \mathbb{E}\left[\nabla_{\beta^m} \mathcal{L}(\theta; m, g, p) \nabla_{\beta^m} \mathcal{L}(\theta; m, g, p)^T\right]\Big][k,l] \\
&= \left[\mathbb{E}\left[\tilde{X}^T \Delta^m s^m\right] \mathbb{E}\left[\tilde{X}^T \Delta^m s^m\right]^T\right][k,l] - \mathbb{E}\left[\tilde{X}^T \Delta^m s^m (s^m)^T \Delta^m \tilde{X}\right][k,l] \\
&= \mathbb{E}\left[\tilde{X}[,k]^T \Delta^m s^m\right] \mathbb{E}\left[\tilde{X}[,l]^T \Delta^m s^m\right] - \mathbb{E}\left[\tilde{X}[,k]^T \Delta^m s^m (s^m)^T \Delta^m \tilde{X}[,l]\right] \\
&= \mathbb{E}\left(\sum_{i=1}^{n} \tilde{x}_{ik} \Delta_i^m s_i^m\right) \mathbb{E}\left(\sum_{j=1}^{n} \tilde{x}_{jl} \Delta_j^m s_j^m\right) - \mathbb{E}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{x}_{ik} \Delta_i^m s_i^m s_j^m \Delta_j^m \tilde{x}_{jl}\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[\tilde{x}_{ik} \Delta_i^m s_i^m] \mathbb{E}[\tilde{x}_{jl} \Delta_j^m s_j^m] - \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[\tilde{x}_{ik} \Delta_i^m s_i^m s_j^m \Delta_j^m \tilde{x}_{jl}] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[\tilde{x}_{ik} \Delta_i^m s_i^m] \mathbb{E}\left[\tilde{x}_{jl} \Delta_j^m s_j^m\right] - \sum_{i \neq j} \mathbb{E}[\tilde{x}_{ik} \Delta_i^m s_i^m] \mathbb{E}[s_j^m \Delta_j^m \tilde{x}_{jl}] \\
&\qquad - \sum_{i=1}^{n} \mathbb{E}[\tilde{x}_{ik} \Delta_i^m s_i^m s_i^m \Delta_i^m \tilde{x}_{il}]
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} \mathbb{E}[\tilde{x}_{ik}\Delta_i^m s_i^m]\mathbb{E}[\tilde{x}_{il}\Delta_i^m s_i^m] - \sum_{i=1}^{n} \mathbb{E}[\tilde{x}_{ik}(\Delta_i^m)^2(H_i^m)^2\tilde{x}_{il}] \\
&= \sum_{i=1}^{n} \Big[\tilde{x}_{ik}(0)\Delta_i^m(0)T_i^\theta(0)H_i^m(0) + \tilde{x}_{ik}(1)\Delta_i^m(1)T_i^\theta(1)H_i^m(1)\Big] \\
&\quad \cdot \Big[\tilde{x}_{il}(0)\Delta_i^m(0)T_i^\theta(0)H_i^m(0) + \tilde{x}_{il}(1)\Delta_i^m(1)T_i^\theta(1)H_i^m(1)\Big] \\
&- \sum_{i=1}^{n} \Big[\tilde{x}_{ik}(0)T_i^\theta(0)(\Delta_i^m(0))^2(H_i^m(0))^2\tilde{x}_{il}(0) + \tilde{x}_{ik}(1)T_i^\theta(1)(\Delta_i^m(1))^2(H_i^m(1))^m\tilde{x}_{il}(1)\Big] \\
&= \sum_{s=0}^{1}\sum_{t=0}^{1} \left[\sum_{i=1}^{n} \tilde{x}_{ik}(s)T_i^\theta(s)\Delta_i^m(s)H_i^m(t)T_i^\theta(t)\Delta_i^m(t)H_i^m(t)\tilde{x}_{il}(t)\right] \\
&\quad - \sum_{s=0}^{1} \left[\sum_{i=1}^{n} \tilde{x}_{ik}(s)T_i^\theta(s)(\Delta_i^m(s))^2(H_i^m(s))^2\tilde{x}_{il}(s)\right] \\
&= \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)[,k]^T T^\theta(s)\Delta^m(s)H^m(s)T^\theta(t)\Delta^m(t)H^m(t)\tilde{X}(k)[,l] \\
&\quad - \sum_{s=0}^{1} X(s)[,k]^T T^\theta(s)(\Delta^m(s))^2(H^m(s))^2\tilde{X}(s)[,l].
\end{aligned}
$$

The sum of the last two terms on the right-hand side of (B.46) is therefore

$$
\begin{aligned}
&\mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta;m,g,p)\right]\mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta;m,g,p)\right]^T - \mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta;m,g,p)\nabla_{\beta^m}\mathcal{L}(\theta;m,g,p)^T\right] \\
&= \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s)\Delta^m(s)H^m(s)T^\theta(t)\Delta^m(t)H^m(t)\tilde{X}(t) \\
&\quad - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s)(\Delta^m(s))^2(H^m(s))^2\tilde{X}(s). \quad \text{(B.48)}
\end{aligned}
$$

Combining (B.46), (B.47), (B.48), we find that

$$
\begin{aligned}
J_{\beta^m}(\theta;m,g) &= \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s)\left[\Delta^m(s)V^m(s)\Delta^m(s) - [\Delta']^m(s)H^m(s)\right]\tilde{X}(s) \\
&\quad + \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s)\Delta^m(s)H^m(s)T^\theta(t)\Delta^m(t)H^m(t)\tilde{X}(t) \\
&\quad - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s)(\Delta^m(s))^2(H^m(s))^2\tilde{X}(s). \quad \text{(B.49)}
\end{aligned}
$$

Submatrix III

Denote submatrix III by $J_{\beta^g}(\theta; m, g)$. The formula for sub-matrix III is similar to that of sub-matrix II (B.49). Substituting g for m in this equation yields

$$J_{\beta^g}(\theta; m, g) = \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s) \left[\Delta^g(s) V^g(s) \Delta^g(s) - [\Delta']^g(s) H^g(s)\right] \tilde{X}(s) + \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s) \Delta^g(s) H^g(s) T^\theta(t) \Delta^g(t) H^g(t) \tilde{X}(t) - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s) (\Delta^g(s))^2 (H^g(s))^2 \tilde{X}(s). \quad \text{(B.50)}$$

Submatrix IV

Denote sub-matrix IV by $J_{(\beta^g, \beta^m)}(\theta; m, g)$. The formula for $J_{(\beta^g, \beta^m)}(\theta; m, g)$ is

$$J_{(\beta^g, \beta^m)}(\theta; m, g) = \mathbb{E}\left[-\nabla_{\beta^g}\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)\right] + \mathbb{E}\left[\nabla_{\beta^g}\mathcal{L}(\theta; m, g, p)\right]\mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)\right]^T - \mathbb{E}\left[\nabla_{\beta^g}\mathcal{L}(\theta; m, g, p)\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)^T\right]. \quad \text{(B.51)}$$

First, we have that

$$\mathbb{E}\left[-\nabla_{\beta^g}\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)\right] = 0, \quad \text{(B.52)}$$

as differentiating $\mathcal{L}$ with with respect to β^g yields a vector that is a function of β^g, and differentiating this vector with respect to β^m yields 0. Next, recall from GLM theory that $\nabla_{\beta^g}\mathcal{L}(\theta; m, g, p) = \tilde{X}^T \Delta^g s^g$ and $\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p) = \tilde{X}^T \Delta^m s^m$. The (k, l)th entry of the last two terms of (B.51) is

$$\begin{aligned}
&\Big[\mathbb{E}\left[\nabla_{\beta^g}\mathcal{L}(\theta; m, g, p)\right]\mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)\right]^T \\
&\qquad - \mathbb{E}\left[\nabla_{\beta^g}\mathcal{L}(\theta; m, g, p)\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)^T\right]\Big][k, l] \\
&= \left[\mathbb{E}\left[\tilde{X}^T\Delta^g s^g\right]\mathbb{E}\left[\tilde{X}^T\Delta^m s^m\right]^T\right][k, l] - \mathbb{E}\left[\tilde{X}^T\Delta^g s^g (s^m)^T \Delta^m \tilde{X}\right][k, l] \\
&= \mathbb{E}\left[\tilde{X}[, k]^T\Delta^g s^g\right]\mathbb{E}\left[\tilde{X}[, l]^T\Delta^m s^m\right] - \mathbb{E}\left[\tilde{X}[, k]^T\Delta^g s^g (s^m)^T\Delta^m \tilde{X}[, l]\right] \\
&= \mathbb{E}\left(\sum_{i=1}^{n} \tilde{x}_{ik}\Delta_i^g s_i^g\right)\mathbb{E}\left(\sum_{j=1}^{n} \tilde{x}_{jl}\Delta_j^m s_j^m\right) - \mathbb{E}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{x}_{ik}\Delta_i^g s_i^g s_j^m \Delta_j^m \tilde{x}_{jl}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[\tilde{x}_{ik}\Delta_i^g s_i^g]\mathbb{E}[\tilde{x}_{jl}\Delta_j^m s_j^m] - \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[\tilde{x}_{ik}\Delta_i^g s_i^g s_j^m \Delta_j^m \tilde{x}_{jl}] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[\tilde{x}_{ik}\Delta_i^g s_i^g]\mathbb{E}\left[\tilde{x}_{jl}\Delta_j^m s_j^m\right] - \sum_{i\neq j} \mathbb{E}[\tilde{x}_{ik}\Delta_i^g s_i^g]\mathbb{E}[\tilde{x}_{jl}\Delta_j^m s_j^m] \\
&\qquad - \sum_{i=1}^{n} \mathbb{E}[\tilde{x}_{ik}\Delta_i^g s_i^g s_i^m \Delta_i^m \tilde{x}_{il}] \\
&= \sum_{i=1}^{n} \mathbb{E}[\tilde{x}_{ik}\Delta_i^g H_i^g]\mathbb{E}[\tilde{x}_{il}\Delta_i^m H_i^m] - \sum_{i=1}^{n} \mathbb{E}[\tilde{x}_{ik} H_i^g \Delta_i^g \Delta_i^m H_i^m \tilde{x}_{il}] \\
&= \sum_{i=1}^{n} \left[\tilde{x}_{ik}(0)\Delta_i^g(0)T_i^\theta(0)H_i^g(0) + \tilde{x}_{ik}(1)\Delta_i^g(1)T_i^\theta(1)H_i^g(1)\right] \\
&\quad \cdot \left[\tilde{x}_{il}(0)\Delta_i^m(0)T_i^\theta(0)H_i^m(0) + \tilde{x}_{il}(1)\Delta_i^m(1)T_i^\theta(1)H_i^m(1)\right] \\
&\qquad - \sum_{i=1}^{n} [\tilde{x}_{ik}(0)T_i^\theta(0)\Delta_i^g(0)H_i^g(0)\Delta_i^m(0)H_i^m(0)\tilde{x}_{il}(0) \\
&\qquad\quad + \tilde{x}_{ik}(1)T_i^\theta(1)\Delta_i^g(1)H_i^g(1)\Delta_i^m(1)H_i^m(1)\tilde{x}_{il}(1)] \\
&= \sum_{s=0}^{1}\sum_{t=0}^{1} \left[\sum_{i=1}^{n} \tilde{x}_{ik}(s)T_i^\theta(s)\Delta_i^g(s)H_i^g(s)T_i^\theta(t)\Delta_i^m(t)H_i^m(t)\tilde{x}_{il}(t)\right] \\
&\qquad - \sum_{s=0}^{1} \left[\sum_{i=1}^{n} \tilde{x}_{ik}(s)T_i^\theta(s)\Delta_i^g(s)H_i^g(s)\Delta_i^m(s)H_i^m(s)\tilde{x}_{il}(s)\right] \\
&= \sum_{s=0}^{1}\sum_{t=0}^{1} \left[\tilde{X}(s)[,k]^T T^\theta(s)\Delta^g(s)H^g(s)T^\theta(t)\Delta^m(t)H^m(t)\tilde{X}(t)[,l]\right] \\
&\qquad - \sum_{s=0}^{1} \left[\tilde{X}[,k]^T T^\theta(s)\Delta^g(s)H^g(s)\Delta^m(s)H^m(s)\tilde{X}[,l](s)\right]. \quad \text{(B.53)}
\end{aligned}
$$

Combining (B.51), (B.52), and (B.53) produces

$$
\begin{aligned}
J_{(\beta^g,\beta^m)}(\theta; m, g) &= \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s)\Delta^g(s)H^g(s)T^\theta(t)\Delta^m(t)H^m(t)\tilde{X}(t) \\
&\qquad - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s)\Delta^g(s)H^g(s)\Delta^m(s)H^m(s)\tilde{X}(s). \quad \text{(B.54)}
\end{aligned}
$$

Submatrix V

Denote submatrix V by $J_{(\beta^m,\pi)}(\theta; m, g)$. The formula for $J_{(\beta^m,\pi)}(\theta; m, g)$ is

$$\begin{aligned} &J_{(\beta^m,\pi)}(\theta; m, g) = \mathbb{E}\left[-\nabla_{\beta^m}\nabla_{\pi}\mathcal{L}(\theta; m, g, p)\right] \\ &+\mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)\right]\mathbb{E}\left[\nabla_{\pi}\mathcal{L}(\theta; m, g, p)\right]^T - \mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)\nabla_{\pi}\mathcal{L}(\theta; m, g, p)^T\right]. \end{aligned} \tag{B.55}$$

We have that

$$\mathbb{E}\left[-\nabla_{\beta^m}\nabla_{\pi}\mathcal{L}(\theta; m, g, p)\right] = 0, \tag{B.56}$$

as β^m and π separate in the log likelihood. Next, set $a \equiv 1/\pi + 1/(1-\pi)$ and $b \equiv n/(1-\pi)$. Recall from GLM theory that $\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p) = \tilde{X}^T\Delta^m s^m$ and from (B.42) that $a\sum_{i=1}^n p_i - b$. The kth entry of the last two terms of (B.55) is

$$\begin{aligned} &\mathbb{E}\left[\nabla_{\pi}\mathcal{L}(\theta; m, g, p)\right]\mathbb{E}\left[\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)[k]\right] - \mathbb{E}\left[\nabla_{\pi}\mathcal{L}(\theta; m, g, p)\nabla_{\beta^m}\mathcal{L}(\theta; m, g, p)[k]\right] \\ &= \left(\mathbb{E}\left[a\sum_{i=1}^n p_i - b\right]\right)\left(\mathbb{E}\left[\tilde{X}[,k]^T\Delta^m s^m\right]\right) - \mathbb{E}\left[\left(a\sum_{i=1}^n p_i - b\right)\tilde{X}[,k]^T\Delta^m s^m\right] \\ &= \left(a\sum_{i=1}^n \mathbb{E}[p_i] - b\right)\left(\sum_{j=1}^n \mathbb{E}[\tilde{x}_{jk}\Delta_j^m s_j^m]\right) - \mathbb{E}\left[\left(a\sum_{i=1}^n p_i - b\right)\left(\sum_{j=1}^n \tilde{x}_{jk}\Delta_j^m s_j^m\right)\right] \\ &= a\sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[p_i]\mathbb{E}[\tilde{x}_{jk}\Delta_j^m s_j^m] - b\sum_{j=1}^n \mathbb{E}[\tilde{x}_{jk}\Delta_j^m s_j^m] \\ &\quad - \left[a\sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[p_i\tilde{x}_{jk}\Delta_j^m s_j^m] - b\sum_{j=1}^n \mathbb{E}[\tilde{x}_{jk}\Delta_j^m s_j^m]\right] \\ &= a\sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[p_i]\mathbb{E}[\tilde{x}_{jk}\Delta_j^m s_j^m] - a\sum_{i\neq j} \mathbb{E}[p_i]\mathbb{E}[\tilde{x}_{jk}\Delta_j^m s_j^m] - a\sum_{i=1}^n \mathbb{E}[p_i\tilde{x}_{ik}\Delta_i^m s_i^m] \\ &= a\sum_{i=1}^n \mathbb{E}[p_i]\mathbb{E}[\tilde{x}_{ik}\Delta_i^m s_i^m] - a\sum_{i=1}^n \mathbb{E}[p_i\tilde{x}_{ik}\Delta_i^m s_i^m] \\ &= a\sum_{i=1}^n T_i^\theta(1)[T_i^\theta(0)\Delta_i^m(0)s_i^m(0)\tilde{x}_{ik}(0) + T_i^\theta(1)\Delta_i^m(1)s_i^m(1)\tilde{x}_{ik}(1)] - a\sum_{i=1}^n T_i^\theta(1)\Delta_i^m(1)s_i^m(1)\tilde{x}_{ik}(1) \\ &= a\sum_{i=1}^n T_i^\theta(0)T_i^\theta(1)\Delta_i^m(0)H_i^m(0)\tilde{x}_{ik}(0) \\ &\quad + a\sum_{i=1}^n \left([T_i^\theta(1)]^2\Delta_i^m(1)H_i^m(1) - T_i^\theta(1)\Delta_i^m(1)H_i^m(1)\right)\tilde{x}_{ik}(1) \end{aligned}$$

$$= a\left[\sum_{i=1}^{n} T_i^\theta(0)T_i^\theta(1)\Delta_i^m(0)H_i^m(0)\tilde{x}_{ik}(0) + \sum_{i=1}^{n} T_i^\theta(1)\Delta_i^m(1)H_i^m(1)[T_i^\theta(1)-1]\tilde{x}_{ik}(1)\right]$$
$$= a\left[\sum_{i=1}^{n} T_i^\theta(0)T_i^\theta(1)\Delta_i^m(0)H_i^m(0)\tilde{x}_{ik}(0) - \sum_{i=1}^{n} T_i^\theta(0)T_i^\theta(1)\Delta_i^m(1)H_i^m(1)\tilde{x}_{ik}(1)\right]$$
$$= a\left(\tilde{X}(0)[,k]^T w^m(0) - \tilde{X}(1)[,k]^T w^m(1)\right). \quad \text{(B.57)}$$

Combining (B.55), (B.56), and (B.57), we conclude that

$$J_{(\beta^m,\pi)}(\theta; m, g, p) = \left(\frac{1}{\pi} + \frac{1}{1-\pi}\right)\left(\tilde{X}(0)^T w^m(0) - \tilde{X}(1)^T w^m(1)\right). \quad \text{(B.58)}$$

Submatrix VI

Denote submatrix VI by $J_{(\beta^g,\pi)}(\theta; m, g)$. Calculations similar to those for submatrix V show that

$$J_{(\beta^g,\pi)}(\theta; m, g, p) = \left(\frac{1}{\pi} + \frac{1}{1-\pi}\right)\left(\tilde{X}(0)^T w^g(0) - \tilde{X}(1)^T w^g(1)\right). \quad \text{(B.59)}$$

Combining submatrices

To summarize, the formulas for submatrices I-VI are as follows:

I

$$J_\pi(\theta; m, g) = \left[\frac{1}{\pi^2} - \frac{1}{(1-\pi)^2}\right]\sum_{i=1}^{n} T_i^\theta(1) + \frac{n}{(1-\pi)^2} + \left(\frac{1}{(1-\pi)} + \frac{1}{\pi}\right)^2\left(\sum_{i=1}^{n}[T_i^\theta(1)]^2 - T_i^\theta(1)\right).$$

II

$$J_{\beta^m}(\theta; m, g) = \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s)\left[\Delta^m(s)V^m(s)\Delta^m(s) - [\Delta']^m(s)H^m(s)\right]\tilde{X}(s) + \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s)\Delta^m(s)H^m(s)T^\theta(t)\Delta^m(t)H^m(t)\tilde{X}(t) - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s)(\Delta^m(s))^2(H^m(s))^2\tilde{X}(s).$$

III

$$J_{\beta^g}(\theta; m, g) = \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s) \left[\Delta^g(s) V^g(s) \Delta^g(s) - [\Delta']^g(s) H^g(s)\right] \tilde{X}(s) + \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s) \Delta^g(s) H^g(s) T^\theta(t) \Delta^g(t) H^g(t) \tilde{X}(t) - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s) (\Delta^g(s))^2 (H^g(s))^2 \tilde{X}(s).$$

IV

$$J_{(\beta^g, \beta^m)}(\theta; m, g) = \sum_{s=0}^{1}\sum_{t=0}^{1} \tilde{X}(s)^T T^\theta(s) \Delta^g(s) H^g(s) T^\theta(t) \Delta^m(t) H^m(t) \tilde{X}(t) - \sum_{s=0}^{1} \tilde{X}(s)^T T^\theta(s) \Delta^g(s) H^g(s) \Delta^m(s) H^m(s) \tilde{X}(s).$$

V

$$J_{(\beta^m, \pi)}(\theta; m, g, p) = \left(\frac{1}{\pi} + \frac{1}{1-\pi}\right) \left(\tilde{X}(0)^T w^m(0) - \tilde{X}(1)^T w^m(1)\right).$$

VI

$$J_{(\beta^g, \pi)}(\theta; m, g, p) = \left(\frac{1}{\pi} + \frac{1}{1-\pi}\right) \left(\tilde{X}(0)^T w^g(0) - \tilde{X}(1)^T w^g(1)\right).$$

We stitch these pieces together and transpose submatrices IV, V, and VI to produce the whole information matrix $J(\theta; m, g)$. Evaluating this matrix at the EM estimate θ^{EM} and inverting yields the asymptotic covariance matrix, which we can use to compute standard errors.

Implementation

To evaluate the observed information matrix, we need to compute the matrices $\Delta^m(j)$, $[\Delta']^m(j)$, $V^m(j)$, and $H^m(j)$ and the vectors $s^m(j)$ and $w^m(j)$ for $j \in \{0, 1\}$. We likewise need to compute the analogous gRNA quantities. The procedure that we propose for this purpose is general, but for concreteness, we describe how to implement this procedure using the `glm` function in R by extending base family objects. We implicitly condition on p_i, z_i^m, and o_i^m.

An `R` family object contains several functions, including `linkinv`, `variance`, and `mu.eta`. `linkinv` is the inverse link function r_m^{-1}. `variance` takes as an argument the mean μ_i^m of the ith example and returns its variance $[\sigma_i^m]^2$. `mu.eta` is the derivative of the inverse link function $[r_m^{-1}]'$. We extend the `R` family object by adding two additional functions: `skewness` and `mu.eta.prime`. `skewness` returns the skewness γ_i^m of the distribution as a function of the mean μ_i, i.e.

$$\texttt{skewness}\,(\mu_i) = \mathbb{E}\left[\left(\frac{m_i - \mu_i^m}{\sigma_i^m}\right)^3\right] := \gamma_i^m.$$

Finally, `mu.eta.prime` is the second derivative of the inverse link function $[r_m^{-1}]''$. Algorithm 8 computes the matrices $\Delta^m(j)$, $[\Delta']^m(j)$, $V^m(j)$, and $H^m(j)$ and vector $s^m(j)$ for given β_m and given family object. (The vector $w^m(j)$ can be computed in terms of $\Delta^m(j)$ and $H^m(j)$.) We use $\sigma_i^m(j)$ (resp. $\gamma_i^m(j)$) to refer to the standard deviation (resp. skewness) of the gene expression distribution the ith cell when the perturbation p_i is set to j.

All steps of the algorithm are obvious except the calculation of $h'_m(l_i^m(j))$ (line 6), $h''(l_i^m(j))$ (line 9), and $V_i^m(j)$ (line 12). We omit the (j) notation for compactness. First, we prove the correctness of the expression for $h'_m(l_i^m)$. Recall the basic GLM identities

$$\psi''_m(\eta_i^m) = [\sigma_i^m]^2 \tag{B.60}$$

and, for all $t \in \mathbb{R}$,

$$r_m^{-1}(t) = \psi'_m(h_m(t)). \tag{B.61}$$

Differentiating (B.61) in t, we find that

$$(r_m^{-1})'(t) = \psi''_m(h_m(t))h'_m(t) \iff h'_m(t) = \frac{(r_m^{-1})'(t)}{\psi''_m(h_m(t))}. \tag{B.62}$$

Finally, plugging in l_i^m for t,

$$h'_m(l_i) = \frac{(r_m^{-1})'(l_i^m)}{\psi''(h_m(l_i^m))} = \frac{(r_m^{-1})'(l_i^m)}{\psi''_m(\eta_i^m)} = \text{ by (B.60) } \frac{(r_m^{-1})'(l_i^m)}{[\sigma_i^m]^2}.$$

Next, we prove the correctness for the expression for $h''_m(l_i^m)$. Recall the exponential family identity

$$\psi'''_m(\eta_i^m) = \gamma_i^m([\sigma_i^m]^2)^{3/2}. \tag{B.63}$$

Differentiating (B.62) in t, we obtain

$$(r_m^{-1})''(t) = \psi_m'''(h_m(t))[h_m'(t)]^2 + \psi_m''(h_m(t))h_m''(t)$$
$$\iff h_m''(t) = \frac{(r_m^{-1})''(t) - \psi_m'''(h_m(t))[h_m'(t)]^2}{\psi_m''(h_m(t))}.$$

Plugging in l_i^m for t, we find that

$$h_m''(l_i^m) = \frac{(r_m^{-1})''(l_i^m) - \psi_m'''(\eta_i^m)[h_m'(l_i^m)]^2}{[\sigma_i^m]^2}$$
$$= \text{(by B.63)} \frac{(r_m^{-1})''(l_i^m) - ([\sigma_i^m]^2)^{3/2}(\gamma_i^m)[h_m'(l_i^m)]^2}{[\sigma_i^m]^2}.$$

Finally, the expression for V_i^m follows from (B.60). We can apply a similar algorithm to compute the analogous matrices for the gRNA modality. Table B.1 shows the `linkinv`, `variance`, `mu.eta`, `skewness`, and `mu.eta.prime` functions for several common family objects (which are defined by a distribution and link function).

B.3 Zero-inflated model

In this section we introduce the "zero-inflated" GLM-EIV model. The zero-inflated GLM-EIV model is appropriate to use when the unperturbed cells do not transcribe *any* gRNA molecules (i.e., when there are no background reads). Let $x_i = [1, z_i]^T \in \mathbb{R}^{d-1}$ be the vector of observed covariates, including an intercept term. (x_i is the same as $\tilde{x}_i$, but with the perturbation indicator p_i removed.) Let $\beta_{g,z} = [\beta_0^g, \gamma_g] \in \mathbb{R}^{d-1}$ be an unknown coefficient vector. ($\beta_{g,z}$ is the same as β_g, but with the perturbation effect β_1^g removed). Let the linear component $l_i^{g,z}$, mean $\mu_i^{g,z}$, and canonical parameter $\eta_i^{g,z}$ of gRNA count distribution of the ith cell be given by

$$l_i^{g,z} = \langle x_i, \beta_{g,z} \rangle + o_i^g; \quad r_g(\mu_i^{g,z}) = l_i^{g,z}; \quad \eta_i^{g,z} = ([\psi_g']^{-1} \circ r_g^{-1})(l_i^{g,z}) := h_g(l_i^{g,z}).$$

The density $f_{g,z}$ of gRNA counts in the zero-inflated model is as follows:

$$f_{g,z}(g_i; \eta_i^{g,z}, p_i) = [f_g(g_i; \eta_i^{g,z})]^{p_i} \mathbb{I}(g_i = 0)^{1-p_i}.$$

In other words, when the cell is *perturbed* (i.e., $p_i = 1$), the zero-inflated density $f_{g,z}$ coincides with the background-read density f_g; by contrast, when the cell is *unperturbed* (i.e., $p_i = 0$), the zero-inflated density $f_{g,z}$ is a point mass at zero. The gene expression density f_m and perturbation indicator density f_p are the same across the background read and zero-inflated models. We assume

Algorithm 8: Computing the matrices $\Delta^m(j)$, $[\Delta']^m(j)$, $V^m(j)$, $H^m(j)$, and $s^m(j)$ given given β_m.

Input A coefficient vector β_m; data $[m_1, \ldots, m_n]$, $[o_1^m, \ldots, o_n^m]$, and $[z_1, \ldots, z_n]$; and a family object containing functions `linkinv`, `variance`, `mu.eta`, `mu.eta.prime`, and `skewness`.

for $j \in \{0, 1\}$ **do**

  for $i \in \{1, \ldots, n\}$ **do**

    $l_i^m(j) \leftarrow \langle \beta_m, \tilde{x}_i(j) \rangle + o_i^m$

    $\mu_i^m(j) \leftarrow \texttt{linkinv}(l_i^m(j))$

    $[\sigma_i^m(j)]^2 \leftarrow \texttt{variance}(\mu_i^m(j))$

    $h'_m(l_i^m(j)) \leftarrow \texttt{mu.eta}(l_i^m(j))/[\sigma_i^m(j)]^2$

    $\gamma_i^m(j) \leftarrow \texttt{skewness}(\mu_i^m(j))$

    $[r_m^{-1}]''(l_i^m(j)) \leftarrow \texttt{mu.eta.prime}(l_i^m(j))$

    $$h''_m(l_i^m(j)) \leftarrow \frac{[r^{-1}]''(l_i^m(j)) - [([\sigma_i^m(j)]^2)^{3/2}][\gamma_i^m(j)][h'_m(l_i^m(j))]^2}{[\sigma_i^m(j)]^2}$$

    // Assign quantities to matrices

    $\Delta_i^m(j) \leftarrow h'_m(l_i^m(j))$

    $[\Delta']_i^m(j) \leftarrow h''(l_i^m(j))$

    $V_i^m(j) \leftarrow [\sigma_i^m(j)]^2$

    $H_i^m(j) \leftarrow s_i^m(j) \leftarrow m_i - \mu_i^m(j)$

  end

end

Table B.1: `linkinv`, `variance`, `mu.eta`, `skewness`, `mu.eta.prime` for common family objects (i.e., pairs of distributions and link functions).

	Gaussian response, identity link	Poisson response, log link	NB response ($s > 0$ fixed), log link
`linkinv`	x	$\exp(x)$	$\exp(x)$
`variance`	x	x	$x + x^2/s$
`mu.eta`	1	x	$\exp(x)$
`skewness`	0	$x^{-1/2}$	$\frac{2x+s}{\sqrt{sx}\sqrt{x+s}}$
`mu.eta.prime`	0	$\exp(x)$	$\exp(x)$

that the gene expression m_i and gRNA count g_i are conditionally independent given the perturbation indicator p_i. The joint density f_z of (m_i, p_i, z_i) is

$$\begin{aligned} f_z(m_i, g_i, p_i) &= f_m(m_i|p_i) f_{g,z}(g_i|p_i) f_p(p_i) \\ &= \pi^{p_i}(1-\pi)^{1-p_i} f_m(m_i; \eta_i^m)[f_g(g_i; \eta_i^{g,z})]^{p_i} \mathbb{I}(g_i = 0)^{1-p_i}. \end{aligned}$$

The complete-data log-likelihood $\mathcal{L}_z$ is

$$\begin{aligned} \mathcal{L}_z(\theta; m, g, p) = &\sum_{i=1}^n \log\left[\pi^{p_i}(1-\pi)^{1-p_i}\right] + \sum_{i=1}^n \log\left[f_m(m_i; \eta_i^m)\right] \\ &+ \sum_{i=1}^n p_i \log\left[f_g(g_i; \eta_i^{g,z})\right] + \sum_{i=1}^n (1-p_i)\log\left[\mathbb{I}(g_i = 0)\right], \end{aligned}$$

where $\theta = [\pi, \beta_m, \beta_{g,z}]$ is the vector of unknown parameters. Integrating over the unobserved variable p_i, the marginal f_z of (m_i, g_i) is

$$f_z(m_i, g_i; \theta) = (1-\pi) f_m(m_i; \eta_i^m(0)) \mathbb{I}(g_i = 0) + \pi f_m(m_i; \eta_i^m(1)) f_g(g_i; \eta_i^{g,z}).$$

Finally, the log-likelihood is

$$\mathcal{L}_z(\theta; m_i, g_i) = \sum_{i=1}^n \log\left[(1-\pi) f_m(m_i; \eta_i^m(0)) \mathbb{I}(g_i = 0) + \pi f_m(m_i; \eta_i^m(1)) f_g(g_i; \eta_i^{g,z})\right].$$

Estimation

To estimate the parameters of the zero-inflated GLM-EIV model, we use an EM algorithm similar to Algorithm 1 but with two changes. First, we use a different formula for the ith membership probability at the t-th step of the algorithm $T_i^{(t)}(1)$. (We use $T_i^{(t)}(1)$ to denote the ith membership probability in *both* the background read and zero inflated cases; the difference should be clear from context.) Let $\theta^{(t)} = (\pi^{(t)}, \beta_m^{(t)}, \beta_{g,z}^{(t)})$ be the parameter estimate at the t-th iteration of the algorithm. Arguing in a manner similar to the background read case, we have that

$$T_i^{(t)}(1) = \frac{1}{\exp(q_i^{(t,z)}) + 1},$$

where

$$q_i^{(t,z)} = \log\left(\frac{(1-\pi^{(t)})\mathbb{P}(M_i = m_i | P_i = 0, \theta^{(t)})\mathbb{P}(G_i = g_i | P_i = 0, \theta^{(t)})}{(\pi^{(t)})\mathbb{P}(M_i = m_i | P_i = 1, \theta^{(t)})\mathbb{P}(G_i = g_i | P_i = 1, \theta^{(t)})}\right).$$

The expression for $q_i^{(t,z)}$ is

$$q_i^{(t,z)} = \log\left[1-\pi^{(t)}\right] + \log\left[f_m\left(m_i; [\eta_i^m(0)]^{(t)}\right)\right] + \log\left[\mathbb{I}(g_i=0)\right] \\ - \log\left[\pi^{(t)}\right] - \log\left[f_m\left(m_i; [\eta_i^m(1)]^{(t)}\right)\right] - \log\left[f_g\left(g_i; [\eta_i^{g,z}]^{(t)}\right)\right],$$

where $[\eta_i^{g,z}]^{(t)} = h_g(\langle x_i, \beta_{g,z}^{(t)}\rangle + o_i^g)$. Notice that if $g_i \geq 1$, then $T_i^{(t)}(1) = 1$. This comports with our intuition that a nonzero gRNA count indicates the presence of a perturbation.

Next, we consider the M step of the EM algorithm, which is similar to the background read case. Define $Q_z(\theta|\theta^{(t)}) = \mathbb{E}_{(P|M=m,G=g,\theta^{(t)})}\left[\mathcal{L}_z(\theta; m, g, p)\right]$. We have that

$$Q_z(\theta|\theta^{(t)}) = \sum_{i=1}^n \left[T_i^{(t)}(1)\log(\pi) + T_i^{(t)}(0)\log(1-\pi)\right] + \\ \sum_{i=1}^n \sum_{j=0}^1 T_i^{(t)}(j)\log\left[f_m(m_i; \eta_i^m(j))\right] + \sum_{i=1}^n T_i^{(t)}(1)\left[\log(f_g(g_i; \eta_i^{g,z}))\right] + C. \quad \text{(B.64)}$$

The three terms of (B.64) are functions of π, β_m, and $\beta_{g,z}$, respectively. The maximizer $\pi^{(t)}$ and $\beta_m^{(t+1)}$ of the first and second term are the same as in the background read case. The maximizer $\beta_{g,z}^{(t+1)}$ of the third term is the maximizer of the GLM with exponential family density f_g, link function r_g, responses g, weights $T^{(t)}(1)$, design matrix X, offsets o^g.

Inference

Next, we derive the asymptotic observed information matrix for the zero-inflated model, allowing us to perform inference. Again, let $T^\theta(1) := \text{diag}\{T_1^\theta(1), \ldots, T_n^\theta(1)\}$, but note that $T_i^\theta(1) = \mathbb{P}(P_i = 1 | G_i = g_i, M_i = m_i, \theta)$ is computed differently than in the background read case. Define the $n \times n$ matrices $\Delta^{(g,z)}$, $[\Delta']^{(g,z)}$, $V^{(g,z)}$, and $H^{(g,z)}$ by

$$\begin{cases} \Delta^{(g,z)} = \text{diag}\{h_g'(l_1^{g,z}), \ldots, h_g'(l_n^{g,z})\} \\ [\Delta']^{(g,z)} = \text{diag}\{h_g''(l_1^{g,z}), \ldots, h_g''(l_n^{g,z})\} \\ V^{(g,z)} = \text{diag}\{\psi_g(\eta_1^{g,z}), \ldots, \psi_g(\eta_n^{g,z})\} \\ H^{(g,z)} = \text{diag}\{m_1 - \mu_1^{g,z}, \ldots, m_n - \mu_n^{g,z}\}. \end{cases}$$

Also, define the $\mathbb{R}^n$ vectors $s^{(g,z)}$ and $w^{(g,z)}$ by

$$s^{(g,z)} = [g_1 - \mu_1^{g,z}, \ldots, g_n - \mu_n^{g,z}]^T,$$

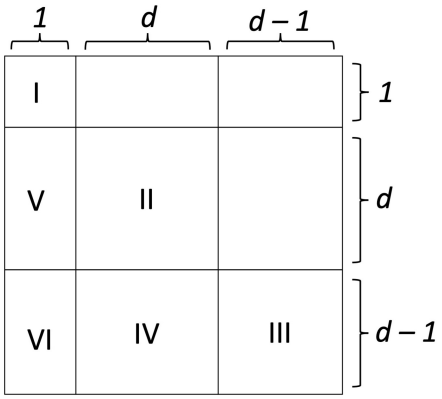


Figure B.4: Block structure of the observed information matrix $J_z(\theta; m, g) = -\nabla^2 \mathcal{L}_z(\theta; m, g)$ for the zero-inflated model. Submatrices I, II, and VI are the same as in the background read model; therefore, we only need to compute submatrices III, VI, and V.

and

$$w^{(g,z)} = [T_1^\theta(0)T_1^\theta(1)\Delta_1^{(g,z)}H_1^{(g,z)}, \ldots, T_n^\theta(0)T_n^\theta(1)\Delta_n^{(g,z)}H_n^{(g,z)}].$$

These quantities are computable, as they do not depend on the unobserved variables $p_1, \ldots, p_n$. Finally, let the unobserved, $n \times n$ matrix P be defined by $P = \text{diag}\{p_1, \ldots, p_n\}$.

The observed information matrix $J_z(\theta; m, g)$ is given by $J_z(\theta; m, g) = -\nabla^2 \mathcal{L}_z(\theta; m, g)$. Louis's theorem implies that

$$\begin{aligned} J_z(\theta; m, g) = &-\mathbb{E}\left[\nabla^2 \mathcal{L}_z(\theta; m, g, p) | G = g, M = m\right] \\ &+ \mathbb{E}\left[\nabla \mathcal{L}_z(\theta; m, g, p) | G = g, M = m\right] \mathbb{E}\left[\nabla \mathcal{L}_z(\theta; m, g, p) | G = g, M = m\right]^T \\ &- \mathbb{E}\left[\nabla \mathcal{L}_z(\theta; m, g, p) \nabla \mathcal{L}_z(\theta; m, g, p)^T | G = g, M = m\right]. \end{aligned}$$

The matrix $J_z(\theta; m, g)$ has dimension $d \times d$ and consists of nine submatrices (Figure B.4). Three of these submatrices (i.e., I, II, and V) are the same as the corresponding submatrices in the background read case. We therefore must compute the remaining submatrices (i.e., III, IV, and VI) to compute the entire matrix $J_z(\theta; m, g)$. Again, in the following, all expectations are understood to be conditional on m and g.

Submatrix III (zero-inflated)

Denote submatrix III by $J_{\beta_{(g,z)}}(\theta; m, g)$ The formula for $J_{\beta_{(g,z)}}(\theta; m, g)$ is

$$J_{\beta_{(g,z)}}(\theta; m, g) = -\mathbb{E}\left[\nabla^2_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)\right] + \mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)\right] \mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)\right]^T - \mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) \nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)^T\right]. \quad \text{(B.65)}$$

GLM theory indicates that $-\nabla^2_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) = X^T P(\Delta^{(g,z)} V^{(g,z)} \Delta^{(g,z)} - (\Delta')^{(g,z)} H^{(g,z)})X$ and $\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) = X^T P \Delta^{(g,z)} s^{(g,z)}$. We begin by computing the first term of (B.65). The only random matrix among X, P, $\Delta^{(g,z)}$, $V^{(g,z)}$, $(\Delta')^{(g,z)}$, and $H^{(g,z)}$ is P. Therefore, by the linearity of expectation,

$$-\mathbb{E}\left[\nabla^2_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)\right] = \mathbb{E}\left[X^T P(\Delta^{(g,z)} V^{(g,z)} \Delta^{(g,z)} - (\Delta')^{(g,z)} H^{(g,z)}\right] = X^T T^\theta(1)(\Delta^{(g,z)} V^{(g,z)} \Delta^{(g,z)} - (\Delta')^{(g,z)} H^{(g,z)})X. \quad \text{(B.66)}$$

Next, we compute the difference of the last two terms of (B.65). The (k, l)th entry of this matrix is

$$\begin{aligned}
&\Big[\mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)\right] \mathbb{E}\left[\nabla_{\beta(g,z)} \mathcal{L}_z(\theta; m, g, p)\right]^T \\
&\qquad - \mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) \nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p)^T\right]\Big][k, l] \\
&= \left[\mathbb{E}\left[X^T P \Delta^{(g,z)} s^{(g,z)}\right] \mathbb{E}\left[X^T P \Delta^{(g,z)} s^{(g,z)}\right]^T\right][k, l] \\
&\qquad - \mathbb{E}\left[X^T P \Delta s^{(g,z)} (s^{(g,z)})^T \Delta^{(g,z)} P X^T\right][k, l] \\
&= \mathbb{E}\left[X[, k]^T P \Delta^{(g,z)} s^{(g,z)}\right] \mathbb{E}\left[X[, l]^T P \Delta^{(g,z)} s^{(g,z)}\right] \\
&\qquad - \mathbb{E}\left[X[, k]^T P \Delta^{(g,z)} s^{(g,z)} (s^{(g,z)})^T \Delta^{(g,z)} P X[, l]\right] \\
&= \mathbb{E}\left(\sum_{i=1}^n x_{ik} P_i \Delta_i^{(g,z)} s_i^{(g,z)}\right) \mathbb{E}\left(\sum_{j=1}^n x_{jl} P_j \Delta_j^{(g,z)} s_j^{(g,z)}\right) \\
&\qquad - \mathbb{E}\left(\sum_{i=1}^n \sum_{j=1}^n x_{ik} P_i \Delta_i^{(g,z)} s_i^{(g,z)} s_j^{(g,z)} \Delta_j^{(g,z)} P_j x_{jl}\right) \\
&= \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[x_{ik} P_i \Delta_i^{(g,z)} s_i^{(g,z)}] \mathbb{E}[x_{jl} P_j \Delta_j^{(g,z)} s_j^{(g,z)}]
\end{aligned}$$

$$
\begin{aligned}
&- \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}s_j^{(g,z)}\Delta_j^{(g,z)}P_jx_{jl}] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}]\mathbb{E}\left[x_{jl}P_j\Delta_j^{(g,z)}s_j^{(g,z)}\right] \\
&\quad - \sum_{i\neq j} \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}]\mathbb{E}[s_j^{(g,z)}P_j\Delta_j^{(g,z)}x_{jl}] \\
&\quad - \sum_{i=1}^{n} \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}s_i^{(g,z)}\Delta_i^{(g,z)}P_ix_{il}] \\
&= \sum_{i=1}^{n} \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}H_i^{(g,z)}]\mathbb{E}[x_{il}P_i\Delta_i^{(g,z)}H_i^{(g,z)}] - \sum_{i=1}^{n} \mathbb{E}[x_{ik}P_i^2(\Delta_i^{(g,z)})^2(H_i^{(g,z)})^2x_{il}] \\
&= \sum_{i=1}^{n} x_{ik}T_i^{\theta}(1)^2(\Delta_i^{(g,z)})^2(H_i^{(g,z)})^2x_{il} - \sum_{i=1}^{n} x_{ik}T_i^{\theta}(1)(\Delta_i^{(g,z)})^2(H_i^{(g,z)})^2x_{il} \\
&= X[,k]^TT^{\theta}(1)^2(\Delta^{(g,z)})^2(H^{(g,z)})^2X[,l] - X[,k]^TT^{\theta}(1)(\Delta^{(g,z)})^2(H^{(g,z)})^2X[,l]
\end{aligned}
$$

Therefore, we have that

$$
\begin{aligned}
&\mathbb{E}\left[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta;m,g,p)\right]\mathbb{E}\left[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta;m,g,p)\right]^T \\
&\qquad - \mathbb{E}\left[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta;m,g,p)\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta;m,g,p)^T\right] \\
&= X^TT^{\theta}(1)^2(\Delta^{(g,z)})^2(H^{(g,z)})^2X - X^TT^{\theta}(1)(\Delta^{(g,z)})^2(H^{(g,z)})^2X \\
&= -X^TT^{\theta}(1)\left(\Delta^{(g,z)}\right)^2\left(H^{(g,z)}\right)^2\left(I - T^{\theta}(1)\right)X. \quad \text{(B.67)}
\end{aligned}
$$

Combining (B.65), (B.66), and (B.67), we conclude that

$$
\begin{aligned}
J_{\beta_{(g,z)}} = (\theta;m,g) &= X^TT^{\theta}(1)(\Delta^{(g,z)}V^{(g,z)}\Delta^{(g,z)} - (\Delta')^{(g,z)}H^{(g,z)})X \\
&\quad - X^TT^{\theta}(1)\left(\Delta^{(g,z)}\right)^2\left(H^{(g,z)}\right)^2\left(I - T^{\theta}(1)\right)X. \quad \text{(B.68)}
\end{aligned}
$$

Submatrix IV (zero-inflated)

Denote submatrix IV by $J_{(\beta_{(g,z)},\beta_m)}(\theta;m,g)$. The formula for submatrix IV is

$$
\begin{aligned}
J_{(\beta_{(g,z)},\beta_m)}(\theta;m,g) &= -\mathbb{E}\left[\nabla_{\beta_{(g,z)}}\nabla_{\beta_m}\mathcal{L}_z(\theta;m,g,p)\right] \\
&\quad + \mathbb{E}\left[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta;m,g,p)\right]\mathbb{E}\left[\nabla_{\beta_m}\mathcal{L}_z(\theta;m,g,p)\right]^T
\end{aligned}
$$

$$- \mathbb{E}\left[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta; m, g, p)\nabla_{\beta_m}\mathcal{L}_z(\theta; m, g, p)]^T\right]. \quad \text{(B.69)}$$

First, we have that

$$-\mathbb{E}\left[\nabla_{\beta_{(g,z)}}\nabla_{\beta_m}\mathcal{L}_z(\theta; m, g, p)\right] = 0, \quad \text{(B.70)}$$

as the derivative in β_m of $\mathcal{L}_z(\theta; m, g, p)$ is a function of β_m, and the derivative in $\beta_{(g,z)}$ of this term is 0. Next, we compute the difference of the last two terms of (B.69). Entry (k, l) of this matrix is

$$\begin{aligned}
&[\mathbb{E}[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta; m, g, p)]\mathbb{E}[\nabla_{\beta_m}\mathcal{L}_z(\theta; m, g, p)]^T \\
&\qquad - \mathbb{E}[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta; m, g, p)\nabla_{\beta_m}\mathcal{L}_z(\theta; m, g, p)^T]][k, l] \\
&\qquad = \left[\mathbb{E}\left[X^T P\Delta^{(g,z)}s^{(g,z)}\right]\mathbb{E}\left[\tilde{X}^T\Delta^m s^m\right]^T\right][k, l] \\
&\qquad\quad - \mathbb{E}\left[X^T P\Delta^{(g,z)}s^{(g,z)}(s^m)^T\Delta^m\tilde{X}\right][k, l] \\
&\qquad = \left[\mathbb{E}\left[X[, k]^T P\Delta^{(g,z)}s^{(g,z)}\right]\mathbb{E}\left[\tilde{X}[, l]^T\Delta^m s^m\right]^T\right] \\
&\qquad\quad - \mathbb{E}\left[X[, k]^T P\Delta^{(g,z)}s^{(g,z)}(s^m)^T\Delta^m\tilde{X}[, l]\right] \\
&\qquad = \mathbb{E}\left(\sum_{i=1}^n x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}\right)\mathbb{E}\left(\sum_{j=1}^n \tilde{x}_{jl}\Delta_j^m s_j^m\right) \\
&\qquad\quad - \mathbb{E}\left(\sum_{i=1}^n\sum_{j=1}^n x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}\Delta_j^m s_j^m\tilde{x}_{jl}\right) \\
&= \sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}]\mathbb{E}[\Delta_j^m s_j^m\tilde{x}_{jl}] - \sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}\Delta_j^m s_j^m\tilde{x}_{jl}] \\
&\qquad = \sum_{i=1}^n\sum_{j=1}^n \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}]\mathbb{E}[\Delta_j^m s_j^m\tilde{x}_{jl}] \\
&\qquad\quad - \sum_{i\neq j} \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}]\mathbb{E}[\Delta_j^m s_j^m\tilde{x}_{jl}] \\
&\qquad\quad - \sum_{i=1}^n \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}s_i^{(g,z)}\Delta_j^m s_j^m\tilde{x}_{jl}] \\
&= \sum_{i=1}^n \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}H_i^{(g,z)}]\mathbb{E}[\tilde{x}_{il}\Delta_i^m H_i^m] - \sum_{i=1}^n \mathbb{E}[x_{ik}P_i\Delta_i^{(g,z)}H_i^{(g,z)}\Delta_i^m H_i^m\tilde{x}_{il}]
\end{aligned}$$

$$= \sum_{i=1}^{n} \left[x_{ik} T_i^{\theta}(1) \Delta_i^{(g,z)} H_i^{(g,z)} \right] \cdot \left[\Delta_i^m(0) T_i^{\theta}(0) H_i^m(0) \tilde{x}_{il}(0) + \Delta_i^m(1) T_i^{\theta}(1) H_i^m(1) \tilde{x}_{il}(1) \right]$$

$$- \sum_{i=1}^{n} \left[x_{ik} T_i^{\theta}(1) \Delta_i^{(g,z)} H_i^{(g,z)} \Delta_i^m(1) H_i^m(1) \tilde{x}_{il}(1) \right]$$

$$= \sum_{s=0}^{1} \sum_{i=1}^{n} x_{ik} T_i^{\theta}(1) H_i^{(g,z)} \Delta_i^{(g,z)} T_i^{\theta}(s) \Delta_i^m(s) H^m(s) \tilde{x}_{il}(s)$$

$$- \sum_{i=1}^{n} \left[x_{il} T_i^{\theta}(1) \Delta_i^{(g,z)} H_i^{(g,z)} \Delta_i^m(1) H_i^m(1) \tilde{x}_{ik}(1) \right]$$

$$= \sum_{s=0}^{1} X[,k]^T T^{\theta}(1) H^{(g,z)} \Delta^{(g,z)} T^{\theta}(s) \Delta^m(s) H^m(s) \tilde{X}(s)[,l]$$

$$- X[,k]^T \Delta^{(g,z)} H^{(g,z)} T^{\theta}(1) \Delta^m(1) H^m(1) \tilde{X}[,l]. \quad \text{(B.71)}$$

Combining (B.65), (B.66), and (B.67) yields

$$J_{(\beta_{(g,z)},\beta_m)}(\theta; m, g) = \left(\sum_{s=0}^{1} X^T T^{\theta}(1) H^{(g,z)} \Delta^{(g,z)} T^{\theta}(s) \Delta^m(s) H^m(s) \tilde{X}(s) \right)$$

$$- X^T \Delta^{(g,z)} H^{(g,z)} T^{\theta}(1) \Delta^m(1) H^m(1) \tilde{X}(1). \quad \text{(B.72)}$$

Submatrix VI (zero-inflated)

Denote submatrix VI by $J_{(\beta_{(g,z)},\pi)}(\theta; m, g)$. The formula for $J_{(\beta_{(g,z)},\pi)}(\theta; m, g)$ is

$$J_{(\beta_{(g,z)},\pi)}(\theta; m, g) = \mathbb{E}\left[-\nabla_{\beta_{(g,z)}} \nabla_{\pi} \mathcal{L}_z(\theta; m, g, p) \right]$$

$$+ \mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) \right] \mathbb{E}\left[\nabla_{\pi} \mathcal{L}_z(\theta; m, g, p) \right]$$

$$- \mathbb{E}\left[\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) \nabla_{\pi} \mathcal{L}_z(\theta; m, g, p) \right]. \quad \text{(B.73)}$$

Recall that $\nabla_{\beta_{(g,z)}} \mathcal{L}_z(\theta; m, g, p) = X^T P \Delta^{(g,z)} s^{(g,z)}$ and $\nabla_{\pi} \mathcal{L}_z(\theta; m, g, p) = a\left(\sum_{i=1}^{n} p_i\right) - b$, where $a = 1/\pi + 1/(1-\pi)$ and $b = n/(1-\pi)$. We have that

$$\mathbb{E}\left[-\nabla_{\beta_{(g,z)}} \nabla_{\pi} \mathcal{L}_z(\theta; m, g, p) \right] = 0, \quad \text{(B.74)}$$

as the derivative in π of $\mathcal{L}_z(\theta; m, g, p)$ is a function of π, and the derivative in $\beta_{(g,z)}$ of this term is 0. Next, we compute the difference of the second two terms of (B.73). The kth entry of this vector is

$$
\begin{aligned}
&\mathbb{E}\left[\nabla_{\pi}\mathcal{L}_z(\theta;m,g,p)\right]\mathbb{E}\left[\nabla_{\beta_{(g,z)}}\mathcal{L}_z(\theta;m,g,x)[k]\right]- \\
&\qquad\mathbb{E}\left[\nabla_{\pi}\mathcal{L}(\theta;m,g,p)\nabla_{\beta_{(g,z)}}\mathcal{L}(\theta;m,g,p)[k]\right] \\
&\quad=\left(\mathbb{E}\left[a\sum_{i=1}^{n}p_i-b\right]\right)\left(\mathbb{E}\left[X[,k]^T P\Delta^{(g,z)}s^{(g,z)}\right]\right)- \\
&\qquad\mathbb{E}\left[\left(a\sum_{i=1}^{n}p_i-b\right)X[,k]^T P\Delta^{(g,z)}s^{(g,z)}\right] \\
&\quad=\left(a\sum_{i=1}^{n}\mathbb{E}[p_i]-b\right)\left(\sum_{j=1}^{n}\mathbb{E}[x_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}]\right) \\
&\qquad-\mathbb{E}\left[\left(a\sum_{i=1}^{n}p_i-b\right)\left(\sum_{j=1}^{n}\tilde{x}_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}\right)\right] \\
&=a\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}[p_i]\mathbb{E}[x_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}]-b\sum_{j=1}^{n}\mathbb{E}[x_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}] \\
&-\left[a\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}[p_ix_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}]-b\sum_{j=1}^{n}\mathbb{E}[x_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}]\right] \\
&\qquad=a\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}[p_i]\mathbb{E}[x_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}]- \\
&\quad a\sum_{i\neq j}\mathbb{E}[p_i]\mathbb{E}[x_{jk}p_j\Delta_j^{(g,z)}s_j^{(g,z)}]-a\sum_{i=1}^{n}\mathbb{E}[x_{ik}p_i^2\Delta_i^{(g,z)}s_i^{(g,z)}] \\
&=a\sum_{i=1}^{n}\mathbb{E}[p_i]\mathbb{E}[x_{ik}p_i\Delta_i^{(g,z)}s_i^{(g,z)}]-a\sum_{i=1}^{n}\mathbb{E}[x_{ik}p_i^2\Delta_i^{(g,z)}s_i^{(g,z)}] \\
&=a\sum_{i=1}^{n}T_i^{\theta}(1)x_{ik}T_i^{\theta}(1)\Delta_i^{(g,z)}s_i^{(g,z)}-a\sum_{i=1}^{n}x_{ik}T_i^{\theta}(1)\Delta_i^{(g,z)}s_i^{(g,z)} \\
&\quad=a\sum_{i=1}^{n}\left(x_{ik}T_i^{\theta}(1)^2\Delta_i^{(g,z)}s_i^{(g,z)}-x_{ik}T_i^{\theta}(1)\Delta_i^{(g,z)}s_i^{(g,z)}\right) \\
&\qquad=a\sum_{i=1}^{n}x_{ik}T_i^{\theta}(1)\Delta_i^{(g,z)}s_i^{(g,z)}\left(T_i^{\theta}(1)-1\right) \\
&\quad=-a\sum_{i=1}^{n}x_{ik}T_i(0)T_i^{\theta}(1)\Delta_i^{(g,z)}H_i^{(g,z)}=-aX[,k]^T w^{(g,z)}. \quad \text{(B.75)}
\end{aligned}
$$

Combining (B.73), (B.74), and (B.75), we conclude that

$$J_{(\beta_{(g,z)},\pi)}(\theta; m, g) = -\left(\frac{1}{\pi} + \frac{1}{1-\pi}\right) X^T w^{(g,z)}. \tag{B.76}$$

B.4 Statistical accelerations and computing

Statistical accelerations

We describe in detail the procedure for obtaining the pilot parameter estimates $(\pi^{\text{pilot}}, \beta_m^{\text{pilot}}, \beta_g^{\text{pilot}})$. This procedure consists of two subroutines, which we label Algorithm 9 and Algorithm 10. The first step (Algorithm 9) is to obtain good parameter estimates for $[\beta_0^m, \gamma_m]^T$ and $[\beta_0^g, \gamma_g]^T$ via regression. Recall that the underlying gene expression parameter vector β_m is $\beta_m = [\beta_0^m, \beta_1^m, \gamma_m]^T \in \mathbb{R}^d$, where β_0^m is the intercept, β_1^m is the effect of the perturbation, and γ_m^T is the effect of the technical factors. To produce estimates $[\beta_0^m]^{\text{pilot}}$ and $[\gamma_m^T]^{\text{pilot}}$, we regress the gene expressions m onto the technical factors X. The intuition for this procedure is as follows: the probability of perturbation π is very small. Therefore, the true log likelihood is approximately equal to the log likelihood that results from omitting p_i from the model:

$$\begin{aligned}\sum_{i=1}^n f_m(m_i; \eta_i^m) = &\underbrace{\sum_{i:p_i=1} f_m(m_i; h_m(\beta_0 + \beta_1 + \gamma^T z_i + o_i^m))}_{\text{few terms}} \\ &+ \underbrace{\sum_{i:p_i=0} f_m(m_i; h_m(\beta_0 + \gamma^T z_i + o_i^m))}_{\text{many terms}} \\ &\approx \sum_{i=1}^n f_m(m_i; h_m(\beta_0 + \gamma^T z_i + o_i^m)).\end{aligned}$$

We similarly can obtain pilot estimates $[\beta_0^g]^{\text{pilot}}$ and $[\gamma_g^T]^{\text{pilot}}$ by regressing the gRNA counts g onto the technical factors X. We extract the fitted values (on the scale of the linear component) for use in a subsequent step: $\hat{f}_i^k = [\beta_0^k]^{\text{pilot}} + \langle [\gamma_k^T]^{\text{pilot}}, z_i \rangle + o_i^k$, for $k \in \{m, g\}$.

Next, we obtain estimates $[\beta_1^m]^{\text{pilot}}$, $[\beta_1^g]^{\text{pilot}}$, and π^{pilot} for β_1^m, β_1^g, and π by fitting a "reduced" GLM-EIV (Algorithm 10). The log likelihood of the no-intercept, univariate GLM with predictor p_i and offset $\hat{f}_i^m$ is approximately equal to the true log likelihood:

$$\sum_{i=1}^n f_m(m_i; \eta_i^m) = \sum_{i=1}^n f_m(m_i; h_m(\beta_0 + \beta_1 p_i + \gamma^T z_i + o_i^m)) \approx \sum_{i=1}^n f_m(m_i; h_m(\beta_1 p_i + \hat{f}_i^m)).$$

Algorithm 9: Computing $[\beta_0^m]^{\text{pilot}}$, $[\gamma_m^T]^{\text{pilot}}$, $[\beta_0^g]^{\text{pilot}}$, and $[\gamma_g^T]^{\text{pilot}}$.

Input Data m, g, o^m, o^g, and X; gene expression distribution f_m and link function r_m; gRNA expression distribution f_g and link function r_g; number of EM starts B.
for $k \in \{m, g\}$ **do**
  Fit a GLM GLM_k with responses k, offsets o^k, design matrix X, distribution f_k, and link function r_k.
  Set $[\beta_0^k]^{\text{pilot}}$ and $[\gamma_k^T]^{\text{pilot}}$ to the fitted coefficients of GLM_k.
  for $i \in \{1, \ldots, n\}$ **do**
    $\hat{f}_i^k \leftarrow [\beta_0^k]^{\text{pilot}} + \langle [\gamma_k^T]^{\text{pilot}}, z_i \rangle + o_i^k$
  end
end
return $([\beta_0^m]^{\text{pilot}} \hat{f}^m, [\gamma_m^T]^{\text{pilot}}, [\beta_0^g]^{\text{pilot}}, \hat{f}^g, [\gamma_g^T]^{\text{pilot}})$

Therefore, to estimate β_1^m, β_1^g, and π, we fit a GLM-EIV model with gene expressions m, gRNA counts g, gene offsets $\hat{f}^m := [\hat{f}_1^m, \ldots, \hat{f}_n^m]^T$, gRNA offsets $\hat{f}^g := [\hat{f}_1^g, \ldots, \hat{f}_n^g]^T$, and *no* intercept or covariate terms. Intuitively, we "encode" all information about technical factors, library sizes, and baseline expression levels into $\hat{f}^m$ and $\hat{f}^g$. We run the algorithm $B \approx 15$ times over randomly-selected starting values for β^m, β^g, and π and select the solution with greatest the log likelihood.

The M step of the reduced GLM-EIV algorithm requires fitting two no-intercept, univariate GLMs with offsets. We derive analytic formulas for the MLEs of these GLMs in the three most important cases: Gaussian response with identity link, Poisson response with log link, and negative binomial response with log link (see section B.4.1; the latter formula is asymptotically exact). Consequently, we do not need to run the relatively slow IRLS procedure to carry out the M step of the reduced GLM-EIV algorithm. Overall, the proposed method for obtaining the full set of pilot parameter estimates requires fitting only two GLMs (via IRLS).

B.4.1 Intercept-plus-offset models

A key step in the algorithm for computing the pilot parameter estimates (Algorithm 10) is to fit a weighted, no-intercept, univariate GLM with nonzero offset terms and a binary predictor variable. We derive an analytic formula for the MLE of this GLM for three important pairs of response distributions and link functions: Gaussian response with identity link, Poisson response with log link, and negative binomial response with log link. The GLM that we seek to

Algorithm 10: Computing π^{pilot}, $[\beta_1^m]^{\text{pilot}}$, $[\beta_1^m]^{\text{pilot}}$.

Input Data m, g; fitted offsets $\hat{f}^m$, $\hat{f}^g$.
`bestLik` $\leftarrow -\infty$
for *counter* $\in \{1, \ldots, B\}$ **do**
  Randomly generate starting parameters $\pi^{\text{curr}}, [\beta_1^m]^{\text{curr}}, [\beta_1^g]^{\text{curr}}$.
  while *Not converged* **do**
    for $i \in \{1, \ldots, n\}$ **do**
      // E step
      $T_i(1) \leftarrow \mathbb{P}(P_i = 1 | M_i = m_i, G_i = g_i, \pi^{\text{curr}}, [\beta_1^g]^{\text{curr}}, [\beta_1^m]^{\text{curr}})$
      $T_i(0) \leftarrow 1 - T_i(1)$
    end
    // M step
    $\pi^{\text{curr}} \leftarrow (1/n) \sum_{i=1}^n T_i(1)$
    $w \leftarrow [T_1(0), T_2(0), \ldots, T_n(0), T_1(1), T_2(1), \ldots, T_n(1)]^T$
    for $k \in \{g, m\}$ **do**
      Fit no-intercept, univariate GLM GLM_k with predictors $[\underbrace{0, \ldots, 0}_{\text{n}}, \underbrace{1, \ldots, 1}_{\text{n}}]$, responses $[k, k]^T$, offsets $[\hat{f}^k, \hat{f}^k]^T$, and weights w.
      Set $[\beta_1^k]^{\text{curr}}$ to fitted coefficient of GLM_k.
    end
    Compute log likelihood `currLik` using π^{curr},$[\beta_1^m]^{\text{curr}}$, and $[\beta_1^g]^{\text{curr}}$
  end
  if *currLik* $>$ *bestLik* **then**
    bestLik $\leftarrow$ currLik
    $\pi^{\text{pilot}} \leftarrow \pi^{\text{curr}}; [\beta_1^m]^{\text{pilot}} \leftarrow [\beta_1^m]^{\text{curr}}; [\beta_1^g]^{\text{pilot}} \leftarrow [\beta_1^g]^{\text{curr}}$
  end
end
return $(\pi^{\text{pilot}}, [\beta_1^m]^{\text{pilot}}, [\beta_1^g]^{\text{pilot}})$

estimate has responses $[m, m]^T$, predictors $[\underbrace{0, \ldots, 0}_{\text{n}}, \underbrace{1, \ldots, 1}_{\text{n}}]$, offsets $[\hat{f}^m, \hat{f}^m]$, and weights $w = [T_1(0), \ldots, T_n(0), T_1(1), \ldots, T_n(1)]^T$. Throughout, C denotes a universal constant. The log likelihood of this GLM is

$$\begin{aligned}\mathcal{L}(\beta_1; m) &= \sum_{i=1}^{n} T_i(0) f_m(m_i; h_m(\beta_1 + \hat{f}_i^m)) + \sum_{i=1}^{n} T_i(1) f_m(m_i; h_m(\hat{f}_i^m)) \\ &= \sum_{i=1}^{n} T_i(1) f_m(m_i; h_m(\beta_1 + \hat{f}_i^m)) + C. \quad \text{(B.77)}\end{aligned}$$

Thus, finding the MLE $\hat{\beta}_1$ is equivalent to estimating a GLM with intercept β_1, offsets $\hat{f}^m$, weights $T_i(1)$, and *no* covariate terms. We term such a GLM a *intercept-plus-offset* model. Below, we study intercept-plus-offset models in generality.

General formulation. Let $\beta \in \mathbb{R}$ be an unknown constant. Let $o_1, \ldots, o_n \sim \mathcal{P}_1$, where $\mathcal{P}_1$ is a distribution. Let $Y_i|o_i, \ldots, Y_n|o_i$ be exponential family-distributed random variables with identity sufficient statistic. Suppose the mean μ_i of $Y_i|o_i$ is given by $r(\mu_i) = \beta + o_i$, where $r : \mathbb{R} \to \mathbb{R}$ is a strictly increasing, differentiable link function. We call this model the *intercept-plus-offset* model.

We derive the (weighted) log likelihood of this model. Let $w_1, \ldots, w_n \sim \mathcal{P}_2$ be weights, where $\mathcal{P}_2$ is a distribution bounded above by 1 and below by 0. (A special case, which corresponds to no weights, is $w_i = 1$ for all $i \in \{1, \ldots, n\}$.) Throughout, we assume that $y_i w_i$ and $\exp(o_i) w_i$ have finite first moment. Suppose the cumulant-generating function and carrying density of the exponential family distribution are $\psi : \mathbb{R} \to \mathbb{R}$ and $c : \mathbb{R} \to \mathbb{R}$, respectively. The canonical parameter η_i of the ith observation is

$$\eta_i = ([\psi']^{-1} \circ r^{-1})(\beta + o_i) := h(\beta + o_i), \quad \text{(B.78)}$$

and the density f of $Y_i|\eta_i$ is $f(y_i; \eta_i) = \exp\{y_i \eta_i - \psi(\eta_i) + c(y_i)\}$. The weighted log likelihood is

$$\mathcal{L}(\beta; y_i) = \sum_{i=1}^{n} w_i \log [f(y_i; \eta_i)] = C + \sum_{i=1}^{n} w_i (y_i \eta_i - \psi(\eta_i)). \quad \text{(B.79)}$$

Our goal is to find the weighted MLE $\hat{\beta}$ of β. We consider three important choices for the exponential family distribution and link function. In the first two cases – Gaussian distribution with identity link and Poisson distribution with log link – we find the *finite-sample* maximizer of (B.79); by contrast, in the third case – negative binomial distribution with log link – we find an *asymptotically exact* maximizer.

Gaussian. First, consider a Gaussian response distribution and identity link function $r(\mu) = \mu$. The cumulant-generating function ψ is $\psi(\eta) = \eta^2/2$, and so, by (B.78),

$$h(t) = [\psi']^{-1}(r^{-1}(t)) = [\psi']^{-1}(t) = t.$$

Plugging $\eta_i = h(\beta + o_i) = \beta + o_i$ and $\psi(\eta_i) = (1/2)(\beta + o_i)^2$ into (B.79), we obtain

$$\mathcal{L}(\beta; y) = \sum_{i=1}^{n} w_i(y_i(\beta + o_i) - (\beta + o_i)^2/2).$$

The derivative of this expression in β is

$$\frac{\partial \mathcal{L}(\beta; y)}{\partial \beta} = \sum_{i=1}^{n} w_i(y_i - \beta - o_i) = \sum_{i=1}^{n} w_i(y_i - o_i) - \beta \sum_{i=1}^{n} w_i.$$

Setting this quantity to 0 and solving for β, we find that the MLE $\hat{\beta}^{\text{gauss}}$ is

$$\hat{\beta}^{\text{gauss}} = \frac{\sum_{i=1}^{n} w_i(y_i - o_i)}{\sum_{i=1}^{n} w_i}.$$

Poisson. Next, consider a Poisson response distribution and log link function $r(\mu) = \log(\mu)$. The cumulant-generating function ψ is $\psi(\eta) = e^{\eta}$. Therefore, by (B.78),

$$h(t) = [\psi']^{-1}(r^{-1}(t)) = [\psi']^{-1}(\exp(t)) = \log(\exp(t)) = t.$$

Plugging $\eta_i = h(\beta + o_i) = \beta + o_i$ and $\psi(\eta_i) = \exp(\beta + o_i)$ into (B.79), we obtain

$$\mathcal{L}(\beta; y) = \sum_{i=1}^{n} w_i \left(y_i(\beta + o_i) - \exp(\beta + o_i)\right).$$

The derivative of this function in β is

$$\frac{\partial \mathcal{L}(\beta; y)}{\partial \beta} = \sum_{i=1}^{n} w_i y_i - w_i \exp(\beta + o_i) = \sum_{i=1}^{n} w_i y_i - \exp(\beta) \sum_{i=1}^{n} w_i \exp(o_i).$$

Setting to zero and solving for β, we find that the MLE $\hat{\beta}^{\text{pois}}$ is

$$\hat{\beta}^{\text{pois}} = \log\left(\frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i e^{o_i}}\right). \tag{B.80}$$

Negative binomial. Finally, we consider a negative binomial response distribution (with fixed size parameter $s > 0$) and log link function $r(\mu) = \log(\mu)$. The cumulant-generating function ψ is $\psi(\eta) = -s\log(1 - e^{\eta})$. The derivative ψ' of ψ is

$$\psi'(t) = s\left(\frac{e^t}{1-e^t}\right) = \frac{s}{e^{-t}-1}.$$

Define the function $\delta : \mathbb{R} \to \mathbb{R}$ by $\delta(t) = -\log(s/t + 1)$. We see that

$$\psi'(\delta(t)) = \frac{s}{\exp(\log(s/t+1)) - 1} = t,$$

implying $\delta = [\psi']^{-1}$. By (B.78), we have that

$$h(t) = [\psi']^{-1}(r^{-1}(t)) = -\log\left(\frac{s}{\exp(t)} + 1\right) = \log\left(\frac{\exp(t)}{s+\exp(t)}\right).$$

Therefore,

$$\begin{aligned}\eta_i = h(\beta + o_i) = \log\left(\frac{\exp(\beta+o_i)}{s+\exp(\beta+o_i)}\right) &= \beta + o_i - \log\left(s + e^{\beta}e^{o_i}\right) \\ &= \beta - \log\left(s + e^{\beta}e^{o_i}\right) + C, \quad \text{(B.81)}\end{aligned}$$

and

$$\begin{aligned}\psi(\eta_i) &= -s\log\left(1 - \frac{\exp(\beta+o_i)}{s+\exp(\beta+o_i)}\right) = -s\log\left(\frac{s}{s+\exp(\beta+o_i)}\right) \\ &= -s\log(s) + s\log[s + \exp(\beta+o_i)] = s\log(s + e^s e^{o_i}) + C. \quad \text{(B.82)}\end{aligned}$$

Plugging (B.81) and (B.82) into (B.79), the log-likelihood (up to a constant) is

$$\begin{aligned}\mathcal{L}(\beta; y) &= \beta\sum_{i=1}^{n} w_i y_i - \sum_{i=1}^{n} w_i y_i \log(s + e^{\beta}e^{o_i}) - s\sum_{i=1}^{n} w_i \log(s + e^{\beta}e^{o_i}) \\ &= \beta\sum_{i=1}^{n} w_i y_i - \sum_{i=1}^{n}(y_i + s)w_i \log(s + e^{\beta}e^{o_i}).\end{aligned}$$

The derivative of $\mathcal{L}$ in β is

$$\frac{\partial\mathcal{L}(\beta; y)}{\partial\beta} = \sum_{i=1}^{n} w_i y_i - \sum_{i=1}^{n}\frac{w_i(y_i+s)e^{\beta}e^{o_i}}{s + e^{\beta}e^{o_i}}.$$

Setting the derivative to zero, the equation defining the MLE is

$$e^{\beta} \sum_{i=1}^{n} \frac{w_i e^{o_i}(y_i + s)}{e^{\beta} e^{o_i} + s} = \sum_{i=1}^{n} w_i y_i. \tag{B.83}$$

We cannot solve for β in (B.83) analytically. However, we can derive an asymptotically exact solution. By the law of total expectation,

$$\begin{aligned}\mathbb{E}\left[\frac{w_i e^{o_i}(y_i + s)}{e^{\beta+o_i} + s}\right] &= \mathbb{E}\left[\mathbb{E}\left[\frac{w_i e^{o_i}(y_i + s)}{e^{\beta+o_i} + s}\middle|(o_i, w_i)\right]\right] \\ &= \mathbb{E}\left[\frac{w_i e^{o_i}(e^{\beta+o_i} + s)}{e^{\beta+o_i} + s}\right] = \mathbb{E}[w_i e^{o_i}];\end{aligned}$$

the second equality holds because $\mathbb{E}[y_i|o_i] = \mu_i = e^{\beta+o_i}$. Dividing by n on both sides of (B.83) and rearranging,

$$\beta = \log\left(\frac{(1/n)\sum_{i=1}^{n} w_i e^{o_i}(y_i + s)/(e^{\beta} e^{o_i} + s)}{(1/n)\sum_{i=1}^{n} w_i y_i.}\right). \tag{B.84}$$

By weak LLN, the limit (in probability) of the MLE $\hat{\beta}^{\mathrm{NB}}$ is

$$\hat{\beta}^{\mathrm{NB}} \xrightarrow{P} \log\left(\frac{\mathbb{E}[w_i y_i]}{\mathbb{E}[w_i e^{o_i}]}\right). \tag{B.85}$$

But the Poisson MLE $\hat{\beta}^{\mathrm{Pois}}$ (B.80) converges in probability to the same limit:

$$\hat{\beta}^{\mathrm{pois}} = \log\left(\frac{(1/n)\sum_{i=1}^{n} w_i y_i}{(1/n)\sum_{i=1}^{n} w_i e^{o_i}}\right) \xrightarrow{P} \log\left(\frac{\mathbb{E}[w_i y_i]}{\mathbb{E}[w_i e^{o_i}]}\right).$$

Therefore, for large n, we can approximate $\hat{\beta}^{\mathrm{NB}}$ by $\hat{\beta}^{\mathrm{pois}}$.

Application to GLM-EIV. The GLM that we seek to estimate (B.77) is an approximate intercept-plus-offset model: $T_1(1), \ldots, T_n(1)$ are the weights $w_1, \ldots, w_n$, and $\hat{f}_1^m, \ldots, \hat{f}_n^m$ are the offsets $o_1, \ldots, o_m$. Of course, $T_1(1), \ldots, T_1(n)$ are in general dependent random variables, as are $\hat{f}_1^m, \ldots, \hat{f}_n^m$. $T_i(1)$ depends on m_i and g_i, as well as the final parameter estimate $(\hat{\pi}, \hat{\beta}_m, \hat{\beta}_g)$, which itself is a function of m and g; the situation is similar for the $\hat{f}_i^m$s. In practice, we find that the intercept-plus-offset model is very good approximation to the GLM (B.77), especially when the number of cells n is large. Additionally, we note that the GLM (B.77) is fitted as a subroutine of the algorithm for producing pilot parameter estimates (Algorithm 10). The quality of the pilot parameter estimates does not affect the validity of the estimation and inference procedures (Algorithm 1), barring issues related to convergence to local optima.

B.4.2 Computing

We describe in detail the at-scale GLM-EIV pipeline. First, we run a round of "precomputations" on all d_g genes and d_p perturbations. The precomputations involve regressing the gene expressions (or gRNA counts) onto the technical factors, thereby "factoring out" Algorithm 9. Next, we run differential expression analyses on the full set of gene-perturbation pairs; for a given pair, this amounts to obtaining the complete set of pilot parameters (by running a reduced GLM-EIV), fitting the GLM-EIV model (Algorithm 1), and performing inference. The three loops in Algorithm 11 are embarassingly parallel and therefore can be massively parallelized.

Algorithm 11: Applying GLM-EIV at scale.

$G \leftarrow \{\text{gene}_1, \ldots, \text{gene}_{d_g}\}; P \leftarrow \{\text{perturbation}_1, \ldots, \text{perturbation}_{d_p}\}$
for *gene* $\in G$ **do**
  Run precomputation (Algorithm 9) on gene; save $\hat{f}^m$, $[\beta_0^m]^{\text{pilot}}$ and $[\gamma_m^T]^{\text{pilot}}$.
end
for *perturbation* $\in P$ **do**
  Run precomputation (Algorithm 9) on perturbation; save $\hat{f}^g$, $[\beta_0^g]^{\text{pilot}}$ and $[\gamma_g^T]^{\text{pilot}}$.
end
for *(gene, perturbation)* $\in G \times P$ **do**
  Load $\hat{f}^m$, $\hat{f}^g$, $[\beta_0^m]^{\text{pilot}}$ $[\gamma_m^T]^{\text{pilot}}$, $[\beta_0^g]^{\text{pilot}}$ and $[\gamma_g^T]^{\text{pilot}}$.
  Compute $[\beta_1^m]^{\text{pilot}}, [\beta_1^g]^{\text{pilot}}, \pi^{\text{pilot}}$ by fitting a reduced GLM-EIV (Algorithm 10).
  Run GLM-EIV using the pilot parameters (Algorithm 1).
end

B.5 Additional simulation study

We ran an additional simulation study in which we modeled the gene and gRNA expressions using a Gaussian distribution with identity link. We generated data on $n = 150,000$ cells, fixing the target of inference β_1^m to -4 and the probability of perturbation π to 0.05. We included "sequencing batch" (modeled as a Bernoulli-distributed variable) and "sequencing depth" (modeled as a Poisson-distributed variable) as covariates in the model. We did not include sequencing depth as an offset because use of the identity link renders offsets meaningless. We varied β_1^g over a grid on the interval $[0, 7]$. We generated

$n_{\text{sim}} = 1,000$ synthetic datasets for each value of β_1^g. We applied accelerated GLM-EIV and thresholded regression to the simulated data. We assessed these methods on the metrics of bias, mean squared error, confidence interval coverage rate, and confidence interval width. We found that accelerated GLM-EIV outperformed the thresholding method: the former method exhibited smaller bias, smaller mean squared error, higher confidence interval coverage rate, and smaller confidence interval width than the latter method (Figure B.5).

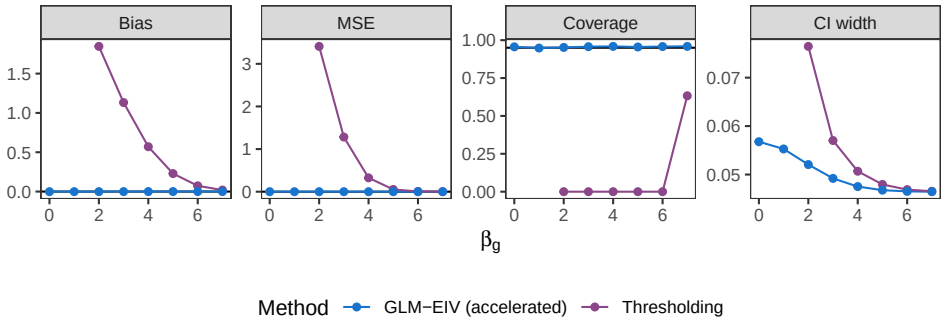


Figure B.5: Additional simulation results on Gaussian data. GLM-EIV (accelerated) outperformed the thresholding method on bias, mean squared error, confidence interval coverage rate, and confidence interval width metrics.

B.6 Data analysis details

First, we performed quality control and basic pre-processing on both datasets. As is standard in single-cell analysis, we removed cells with a high fraction ($> 8\%$) of mitochondrial reads (Choudhary and Satija, 2022). We additionally excluded genes that were expressed in fewer than 10% of cells or that had a mean expression level of less than 1. We excluded cells in the Gasperini dataset with gene transcript UMI or gRNA counts below the 5th percentile or above the 95th percentile to reduce the effect of outliers. We did not repeat this latter quality control step on the Xie data because the Xie data appeared to be less noisy. The quality-controlled Gasperini and Xie datasets contained $n = 170,645$ (resp. $n = 101,508$) cells, $2,079$ (resp. $1,030$) genes, and $6,598$ (resp. 516) distinct perturbations.

The Gasperini dataset came with $17,028$ candidate *cis* pairs, $97,818$ negative control pairs, and 322 positive control pairs. The *cis* pairs consisted of genes

paired to nearby enhancers with unknown regulatory effects. The negative control pairs consisted of non-targeting gRNAs paired to genes. The positive control pairs are described in the main text. The Xie data did not come with either *cis*, negative control, or positive control pairs. Therefore, we constructed a set of 681 candidate *cis* pairs by pairing perturbations to nearby genes, and we constructed a set of $50,000$ *in silico* negative control by pairing perturbations to genes on different chromosomes. See the *Methods* section of Barry et al. (2021a) for details on the construction of *cis* and *in silico* negative control pairs on the Xie data. Because the negative control pairs are not expected to exhibit a regulatory relationship, the ground truth fold change in gene expression for these pairs is taken to be unity.

We modeled the gene expression counts using a negative binomial distributions with unknown size parameter s; we estimated s using the glm.nb package. Choudhary and Satija (2022) report that Poisson models accurately capture highly sparse single-cell data. Although Choudhary and Satija did not investigate the application of Poisson models gRNA data specifically, we modeled the gRNA counts using Poisson distributions, as the gRNA modality exhibited greater sparsity than the gene modality.

We applied GLM-EIV and the thresholding method to analyze the entire set of pairs in both datasets. We did not report results on the candidate *cis* pairs in the text because we do not know the ground truth for these pairs, making them less useful for method assessment. We focused our attention instead on the negative control pairs in both datasets and the positive control pairs in the Gasperini dataset.

We describe in more detail how we conducted the "excess background contamination" analysis. For each positive control pair, we varied excess background contamination over the grid $[0.0, 0.05, 0.1, \ldots, 0.4]$. For a given level of excess background contamination, we generated $B = 50$ synthetic gRNA datasets, holding fixed the raw gene expressions, covariates, library sizes, and fitted perturbation probabilities. We fitted GLM-EIV and the thresholding method to the data, yielding estimates $[\hat{\beta}_1^m]^{(1)}, \ldots, [\hat{\beta}_1^m]^{(B)}$. Next, we averaged over the $[\hat{\beta}_1^m]^{(i)}$s to obtain the mean estimate for a given pair and level of background contamination, and we calculated the REC using these mean estimates.

B.7 Additional related work

Several authors working on statistical methods for single-cell data recently have extended models that (implicitly or explicitly) assume Gaussianity and homoscedasticity to a broader class of exponential family distributions. For

example, Lin et al. (2021) and Townes et al. (2019) (separately) developed eSVD and GLM-PCA, generalizations of SVD and PCA, respectively, to exponential family response distributions. Unlike their vanilla counterparts, eSVD and GLM-PCA can model gene expression counts directly, improving performance on dimension reduction tasks. We see our work (in part) as a continuation of this broad effort to "port" common statistical methods and models to single-cell count data. Our focus, however, is on regression rather than dimension reduction: we extend the classical errors-in-variables model in several key directions (see above), enabling its direct and natural application to multimodal single-cell data.

C

Supplementary tables and figures for Chapter 4

Paper	Datasets	CRISPR modality	Platform	Target	Modality measured	Cell type
Frangieh 2021	co-culture, control, IFN-γ (3)	CRISPRko	Perturb-CITE seq	Gene TSSs	Gene expressions*	TIL
Papalexi 2021	ECCITE screen (1)	CRISPRko	ECCITE-seq	Gene TSSs	Gene and protein expressions	K562
Schraivogel 2020	Enhancer screen (1)	CRISPRi	Targeted perturb-seq	Enhancers	Gene expressions	THP1
-	Simulated dataset (1)	-	-	Gene TSSs	Gene expressions	-

Table C.1: **Datasets analyzed in this work**. The first column indicates the name of a low-MOI single-cell CRISPR screen paper; the second column indicates the datasets that we obtained from that paper; and the subsequent columns indicate the (paper-specific) biological attributes of the data, including CRISPR modality, technology platform, target type, cellular modality measured, and cell type. *The Frangieh data also contain protein measurements, but we focus exclusively on the gene modality in this work.

Dataset	**N genes (or proteins)**	**N cells**	**N targeting gRNAs**	**N NT gRNAs**	**N neg. control pairs**	**N pos. control pairs**
Frangieh Co-culture	14,438	46,427	744	74	596,344	181
Frangieh control	15,449	30,486	744	74	528,239	170
Frangieh IFN-γ	14,654	50,053	744	74	565,502	181
Papalexi (gene)	14,559	20,729	101	9	100,458	25
Papalexi (protein)	4	20,729	101	9	36	2
Schraivogel*	82 (Chr11), 71 (Chr8)	99,884 (Chr11), 88,715 (Chr8)	3,073 (Chr11), 4,089 (Chr8)	30 (Chr11), 30 (Chr8)	4,693 (pooled)	25 (pooled)
Simulated	4439	10,000	-	25	108,510	-

Table C.2: **Statistical attributes of the datasets**. The number of genes, cells, targeting gRNAs, NT gRNAs, negative control pairs, and positive control pairs for each dataset. Neg., negative; pos., positive.
*Schraivogel separately assayed two chromosomes: Chr11 and Chr8. Given the similarity of these assays, we pool together the negative and positive control pairs across assays.

Method	**Sparsity**	**Confounding (beyond library size)**	**Model misspecification**
Seurat-Wilcox	No	No	Yes
Seurat-NB	No	No	No
t-test	Yes	No	No
MAST	No	No	No
KS test	No	No	Yes
NB regression (with covariates)	No	Yes	No
Standard permutation test	Yes	No	Yes
SCEPTRE	**Yes**	**Yes**	**Yes**

Table C.3: **Analysis challenges addressed by each method**. Each cell indicates whether the method in the row addresses the analysis challenge in the column. SCEPTRE (bottom row) is the only method that addresses all three analysis challenges. Note: MIMOSCA is excluded from this table, as we could not determine which analysis challenge(s) MIMOSCA addresses.

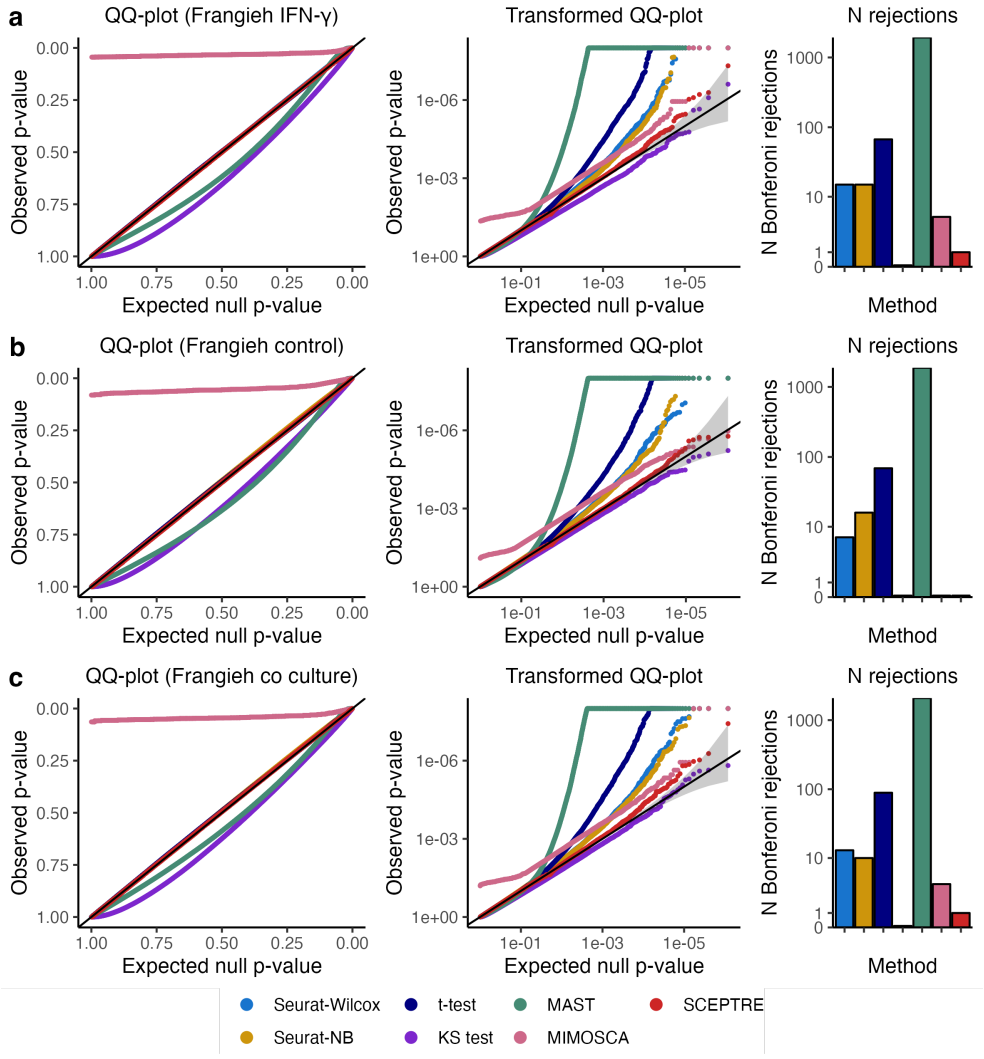


Figure C.1: **Calibration results for all methods on Frangieh IFN-γ, Frangieh control, and Frangieh co-culture negative control data.** Left, untransformed QQ plots; middle, negative log-10 transformed QQ plots; right; number of false rejections after a Bonferroni correction at level 0.1.

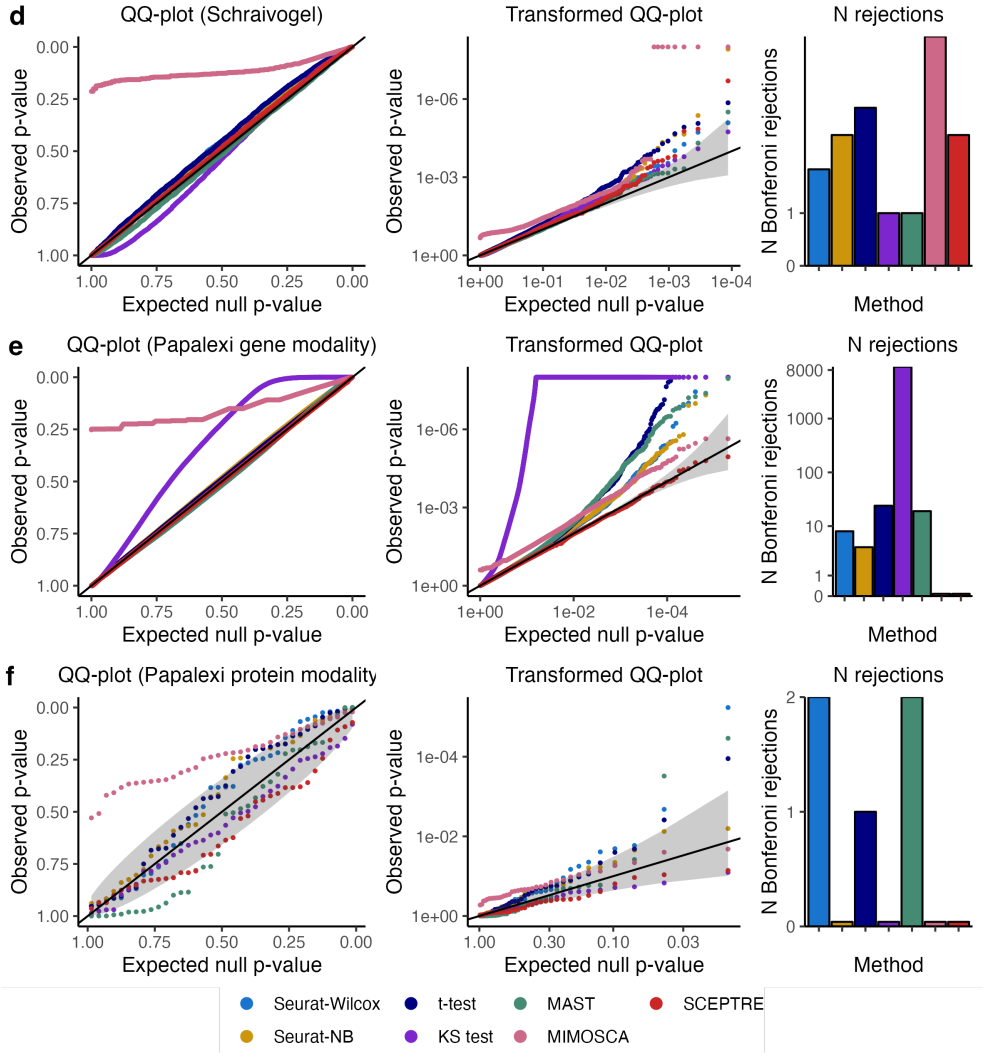


Figure C.2: **Calibration results for all methods on Schraivogel, Papalexi (gene modality), and Papalexi (protein modality) negative control data.** Interpretation is the same as in Figure C.1.

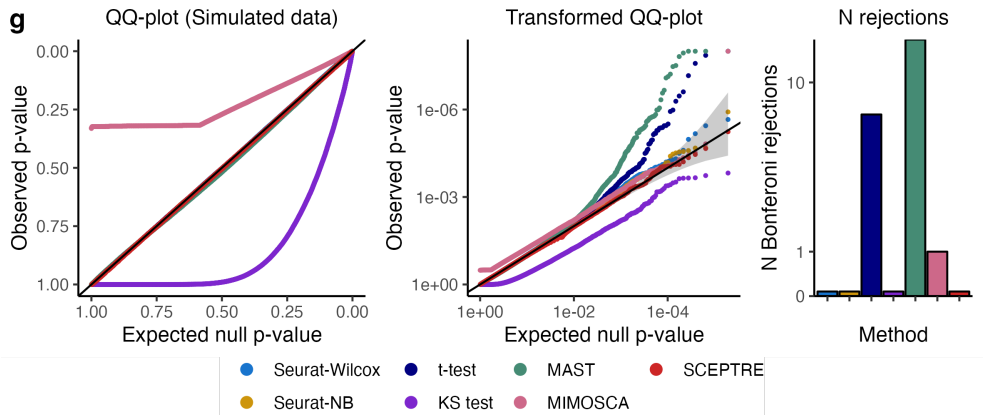


Figure C.3: **Calibration results for all methods on simulated data.** Interpretation is the same as in Figure C.1.

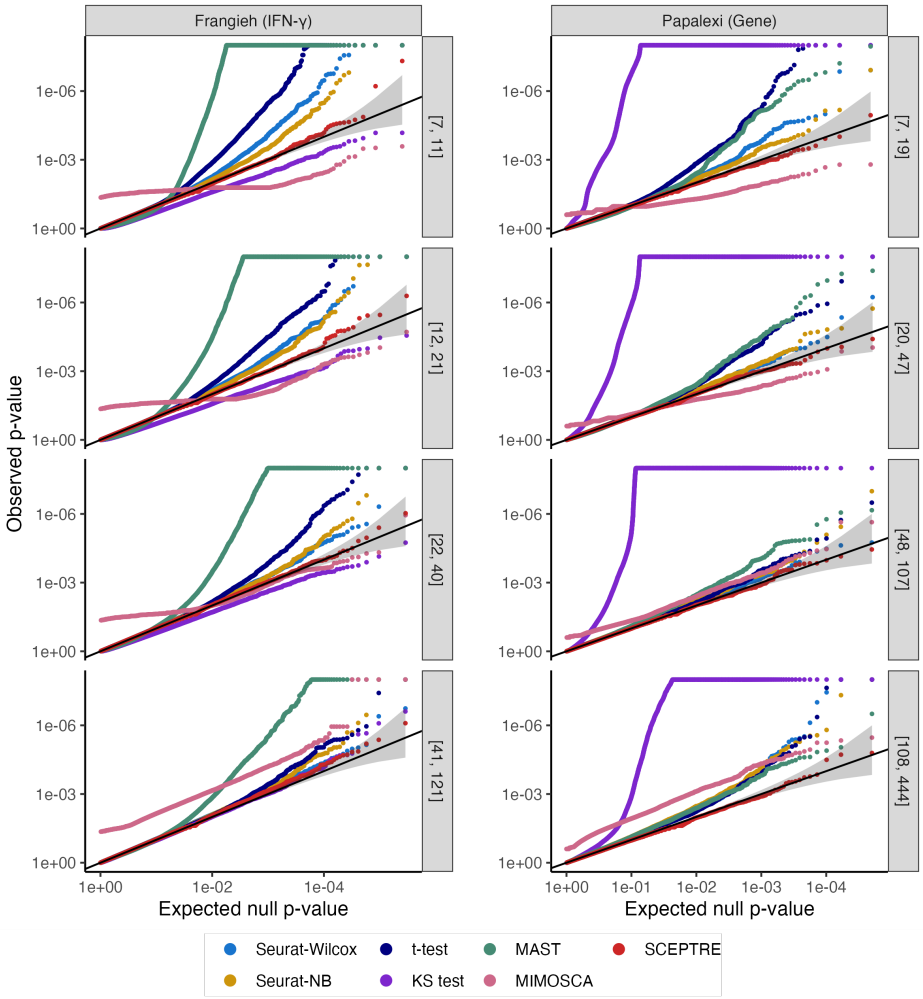


Figure C.4: **Calibration results for all methods on Frangieh IFN-γ and Papalexi (gene modality) datasets, stratified by effective sample size**. Negative control gene-gRNA pairs are partitioned into four bins of approximately equal size based on the number of treatment cells with nonzero expression in a given pair. The interval on the right-hand side of each panel indicates the minimum and maximum number of treatment cells with nonzero gene expression for pairs in that bin. Some methods (e.g., Seurat-Wilcox on the Frangieh IFN-γ data) exhibit better calibration as the number of treatment cells with nonzero expression increases (i.e., as sparsity decreases).

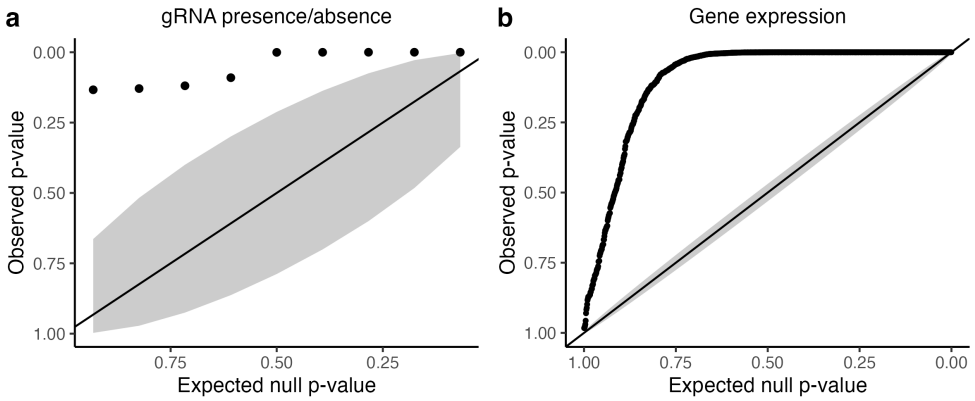


Figure C.5: **Confounding due to biological replicate on the Papalexi (gene modality) data.** Left, QQ plot of p-values for tests of association between the gRNA indicator and biological replicate for each NT gRNA (tests carried out using Fisher's exact test). Right, p-values for tests of association between (relative) gene expression and biological replicate for each gene (tests carried out using NB GLM likelihood ratio test). The inflation of the p-values indicates that the bulk of NT gRNAs and genes is impacted by biological replicate, creating a confounding effect.

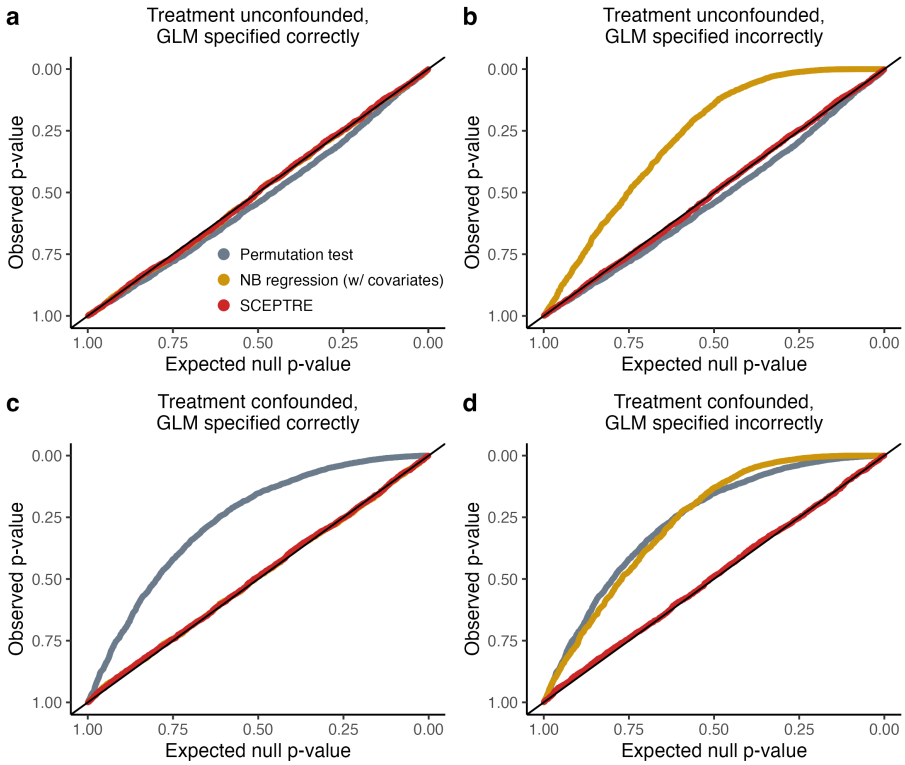


Figure C.6: **Demonstration of the CAMP ("confounder adjustment via marginal permutations") phenomenon on realistic semi-synthetic data**. Application of a standard permutation test, NB regression, and SCEPTRE to realistic semi-synthetic data generated under two conditions: confounded and unconfounded. Panels **a** and **b** (resp., **c** and **d**) show the results on the unconfounded (resp., confounded) data; meanwhile, panels **a** and **c** (resp., **b** and **d**) show the results under correct (resp. incorrect) specification of the negative binomial size parameter. The permutation test (gray) works well when the data are unconfounded (panels **a** and **b**) but breaks down in the presence of confounding (panels **c** and **d**). On the other hand, NB regression is well-calibrated when the size parameter is correctly specified (panels **a** and **c**) but fails when the size parameter is misspecified (panels **b** and **d**). SCEPTRE is well-calibrated in all settings. We note that SCEPTRE is expected to break down when the (i) problem is confounded and the NB regression model is arbitrarily misspecified or (ii) the problem is confounded and the sparsity is high. Details of the simulation study are given in Section Simulation study details.

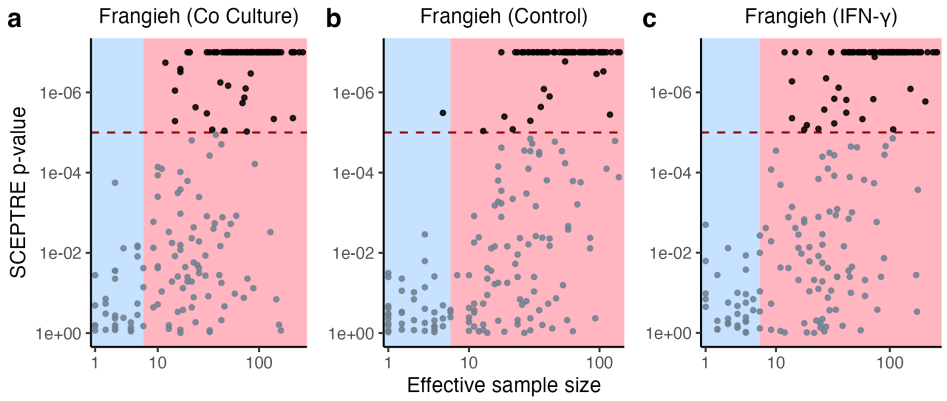


Figure C.7: **SCEPTRE's power to detect associations increases as effective sample size increases**. **a-c**, SCEPTRE p-value (truncated at 10^{-6}) versus effective sample size for each pair on the Frangieh co-culture (**a**), control (**b**), and IFN-γ (**c**) positive control data. The horizontal dashed line is drawn at 10^{-5}, demarcating a highly significant discovery. SCEPTRE makes only one rejection at a highly significant level on pairs for which the effective sample size less than seven (blue region).

D

Code and data availability

Code and data for Chapter 2

Analysis results are available online at `https://upenn.box.com/v/sceptre-files-v8`. All analysis was performed on publicly available data. The Gasperini et al. CRISPR screen data are available at `www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120861`. The Xie et al. single-cell and bulk CRISPR screen data are available at `www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE129837`. The ChIP-seq data are taken from the ENCODE project (Dunham et al., 2012) and are available at `www.encodeproject.org/`. The HI-C enrichment analysis is based on the data from Rao et al. (2014), available at `www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525`. The eQTL and eRNA co-expression p-values are taken from the GeneHancer database (Fishilevich et al., 2017), available as part of GeneCards (`www.genecards.org/`). The `sceptre` R package is available at `github.com/Katsevich-Lab/sceptre`. The scripts used to run the analyses reported in this paper are available at `github.com/Katsevich-Lab/sceptre-manuscript` (permanent version deposited at Zenodo; DOI 10.5281/zenodo.5643541; Barry et al. 2021b).

Code and data for Chapter 3

Results are deposited at `upenn.box.com/v/glmeiv-files-v1`. Github repositories containing manuscript replication code, the `glmeiv` R package, and the cloud/HPC-scale GLM-EIV pipeline are available at `github.com/timothy-barry/glmeiv-manuscript`, `github.com/timothy-barry/glmeiv`, and `github.com/timothy-barry/glmeiv-pipeline`, respectively. Detailed replication instructions are available in the first repository.

Code and data for Chapter 4

The code for Chapter 4 is spread across nine Github repositories. One can navigate to `github.com/Katsevich-Lab/sceptre2-manuscript` (i.e., the second Github repository of those listed) for instructions on reproducing the analyses reported in this Chapter.

1. The `sceptre` package implements the SCEPTRE method. The repository contains detailed tutorials and examples.

   `katsevich-lab.github.io/sceptre/`

2. The `sceptre2-manuscript` repository contains code to reproduce all analyses reported in this paper.

   `github.com/Katsevich-Lab/sceptre2-manuscript`

3. The `lowmoi` package implements the existing single-cell CRISPR screen analysis methods. (Methods originally written in Python are implemented via `reticulate`).

   `github.com/Katsevich-Lab/lowmoi`

4. The `undercover-grna-pipeline` repository contains the Nextflow pipeline to carry out negative control benchmarking analysis.

   `github.com/Katsevich-Lab/undercover-grna-pipeline`

5. The `pc-grna-pipeline` repository contains the Nextflow pipeline to carry out the positive control benchmarking analysis.

   `github.com/Katsevich-Lab/pc-grna-pipeline`

6. The `ondisc` package implements data structures that we use to store the single-cell expression data.

   `github.com/timothy-barry/ondisc`

7. The `import-frangieh-2021` repository imports and processes the Frangieh data.

   `github.com/Katsevich-Lab/import-frangieh-2021`

8. The `import-papalexi-2021` repository imports and processes the Papalexi data.

   `github.com/Katsevich-Lab/import-papalexi-2021`

9. The `import-schraivogel-2020` repository imports and processes the Schraivogel data.

   `github.com/Katsevich-Lab/import-schraivogel-2020`

Next, the uniformly processed single-cell CRISPR screen data (stored in `ondisc` format) are available in the following directory: `www.dropbox.com/sh/jekmk1v4mr4kj3b/AAAhznGqk-TIZKhW40xiU6ORa?dl=0`. The ChIP-seq data are available at `10.1016/j.immuni.2013.08.009`. Finally, all results are stored in the following directory: `https://www.dropbox.com/sh/76sviudz0cg0mgy/AACoigzD0CWGa9S5HVgG2gHua?dl=0`.