Title
Capturing hidden covariates with linear factor models and other statistical methods in differential gene expression and expression quantitative trait locus studies

Author
Zhou, Heather J

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Capturing hidden covariates with linear factor models and other statistical methods in differential gene expression and expression quantitative trait locus studies

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Heather J. Zhou

2022

ABSTRACT OF THE THESIS

Capturing hidden covariates with linear factor models and other statistical methods in differential gene expression and expression quantitative trait locus studies

by

Heather J. Zhou
Master of Science in Statistics
University of California, Los Angeles, 2022
Professor Jingyi Li, Chair

This works aims to provide value to three types of readers. First, for students in statistics, psychology, and the social sciences, I provide a summary and review of three classical statistical methods: factor analysis, principal component analysis (PCA), and probabilistic PCA (PPCA), all of which fall under the category of linear factor models. These methods are widely used in many fields, including psychology, education, and computational biology, and are the cornerstones of many new, more complicated methods. However, most available materials about them are either decades old (and very long and use old-style notations) or cursory. This work provides current coverage of them that is in-depth yet concise.

Second, for new computational biologists who are unfamiliar with differential gene expression (DE) analysis and quantitative trait locus (QTL) analysis — in particular, expression quantitative trait locus (eQTL) analysis — I provide an introduction to DE analysis and eQTL analysis from a statistical perspective, with an emphasis on DE and eQTL analysis with hidden covariates. I avoid unnecessary jargon and aim for this material to be accessible to those without much background in biology.

Third, for computational biologists and geneticists who need to work with newly developed computational methods such as surrogate variable analysis (SVA), probabilistic estimation of expression residuals (PEER), and hidden covariates with prior (HCP), I document these methods in a unified framework and explore their connections to classical methods such as factor analysis and PCA. To the best of our knowledge, such precise and in-depth review of SVA, PEER, and HCP is currently not available elsewhere in the literature.

In short, this work aspires to be a useful reference manual for students and researchers working with linear factor models or newly developed methods for capturing hidden covariates in DE or eQTL analysis.

The thesis of Heather J. Zhou is approved.

Chad J. Hazlett

Mark S. Handcock

Jingyi Li, Committee Chair

University of California, Los Angeles

2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This work is funded in part by NSF DGE-1829071 and NIH/NHLBI T32HL139450. My deepest gratitude goes to the professors who taught me at Swarthmore College (Dr. Steve Wang, Dr. Kelly McConville, Dr. Charles Grinstead, to name a few) and UCLA, my advisor Dr. Jingyi Jessica Li, and my husband, family, and peers, without whom I could not have completed this work.

CHAPTER 1

Introduction

High-throughput technologies such as microarrays [1] and next-generation RNA-sequencing (RNA-seq) [2] have enabled biologists to probe the expression of thousands of genes simultaneously at a relatively low cost, making possible a wide variety of genomic research. Two classic examples are differential gene expression (DE) analysis and expression quantitative trait locus (eQTL) analysis. In DE analysis, researchers look for signals where expression levels of a gene differ between conditions. In eQTL analysis, researchers look for associations between genetic variants and gene expression levels. A background in these two types of analysis from a statistical perspective is given in Section 3.1.

In both DE and eQTL analyses, several biological and technical factors, including sex, age, and batches, are known to be potential confounding factors. In addition, many new statistical methods have been developed to infer unknown confounders, the most popular ones (listed in chronological order) being surrogate variable analysis (SVA) [3, 4], probabilistic estimation of expression residuals (PEER) [5, 6], and hidden covariates with prior (HCP) [7]. It is standard practice today to use these methods to capture and correct for hidden confounders in DE and eQTL analyses [e.g., 8, 9].

Despite their popularity, the documentation of the methodology behind SVA, PEER, and HCP is surprisingly inadequate. In Sections 3.2 to 3.4, I fill this gap in the literature by documenting these methods in a precise and in-depth way. Further, I show that these methods are closely related among themselves as well as to classical statistical methods such as factor analysis, principal component analysis (PCA), and probabilistic PCA (PPCA),

shedding light on the theoretical interpretation of SVA, PEER, and HCP.

SVA is purely an algorithm that is not defined based on a statistical model or loss function, though the algorithm itself is heavily based on PCA. To the best of our knowledge, the inner workings of SVA are not precisely documented anywhere — one would need to look at the source code of the R package SVA to know the exact steps of the algorithm. This is an alarming issue especially given SVA's popularity and the fact that SVA is purely an algorithm, so the steps of the algorithm define the method.

PEER is based on a Bayesian probabilistic model and can be considered a Bayesian version of factor analysis. Besides notational issues, the PEER method as described in the original paper, Stegle et al. [5], is not aligned with the current implementation of the R package peer. For example, Stegle et al. [5] claims that PEER can take into account genotype data when estimating the hidden variables, but the R package does not allow the user to input any genotype data.

HCP is defined by minimizing a loss function and is closely related to PCA. The documentation of HCP is far from satisfactory in both the original paper [7] and the R package documentation. For example, as I detail in Section 3.4, the loss function is incorrectly specified in both places in a nontrivial way.

Given that SVA and HCP are closely related to PCA, and PEER is closely related to factor analysis, I further show that PCA and factor analysis are closely related statistical methods (Chapter 2), hence unifying SVA, PEER, and HCP. To do that, I provide a detailed yet concise review of the essentials of factor analysis and PCA and show that although factor analysis is based on a probabilistic model and PCA is traditionally derived by optimizing some objective functions (either maximum variance or minimum reconstruction error), PCA can also be derived as a limit of the PPCA model, which in turn is a special case of the factor analysis model. The reason Chapter 2 is necessary is because during my research, I found that most available materials about factor analysis, PCA, and PPCA are either decades old (and very long and use old-style notations) or cursory. This work provides current coverage

of them that is in-depth yet concise, which can be referred to by students in statistics, psychology, and the social sciences independently of the other chapters.

In sum, this work provides unparalleled documentation of three classical statistical methods: factor analysis, PCA, PPCA, and three new methods developed for capturing hidden covariates in DE and eQTL studies: SVA, PEER, and HCP. All six methods are closely related. Among these six methods, factor analysis, PCA, PPCA, and PEER fall under the category of linear factor models, while HCP is defined by minimizing a loss function and does not have a probabilistic interpretation, and SVA is purely an algorithm that is not defined based on a statistical model or loss function.

CHAPTER 2

Classical methods

2.1 Factor analysis

Factor analysis is based on a generative probabilistic model [10, 11]. We start by introducing two viewpoints of the underlying statistical model (Section 2.1.1 and Section 2.1.2). The two viewpoints describe the exact same model, but the first has the advantage of helping us see the overall idea better, while the second is more convenient for deriving the relevant distributions.

After setting up the model, we will introduce three basic steps of inference: maximum likelihood estimation (Section 2.1.4), rotation (Section 2.1.5), and hidden factor prediction (Section 2.1.6).

2.1.1 Matrix viewpoint

Let X denote the $n \times p$ **observed data** matrix that is observation by variable (i.e., feature). Specifically, in gene expression studies, X would be sample by gene. Let Z denote the $n \times K$ **hidden factor matrix**, also known as the score matrix. That is, we assume that there are K hidden factors, and K is often chosen to be smaller than p. Let $\mathbf{W}^\top$ denote the $K \times p$ **weight matrix**, also known as the effect size matrix or loading matrix; we use transpose here so that the notation in the second viewpoint will be consistent with standard factor analysis notation. Finally, let ε denote the **error matrix**. We use boldface for $\mathbf{W}^\top$ but not for X, Z, and ϵ because $\mathbf{W}^\top$ is a fixed matrix whereas X, Z, and ϵ are random matrices

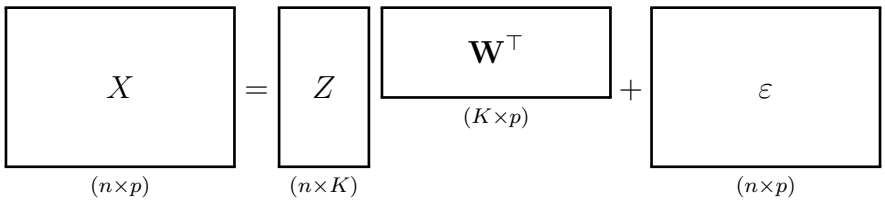

Figure 2.1: Illustration of the factor analysis model.

(see below).

As illustrated in Figure 2.1, the underlying statistical model of factor analysis is

$$X = Z\,\mathbf{W}^\top + \varepsilon, \tag{2.1.1}$$

where we assume that each entry of Z is independently drawn from a standard normal distribution:

$$z_{ik} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)\,,\ i = 1, \cdots, n;\ k = 1, \cdots, K, \tag{2.1.2}$$

and independent from Z, each entry in each column of ε is independently drawn from a normal distribution with gene-specific variance. That is, for $j = 1 \cdots, p$, we have

$$\varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \psi_j)\,,\ i = 1, \cdots, n, \tag{2.1.3}$$

where ψ_j is the gene-specific error variance, often called specific variance, unique variance, or uniqueness. Notice the resemblance of this model to the classic linear regression model, where the response variable is modeled as a linear combination of several predictor variables (plus some noise). The only difference is that in factor analysis, we have p response variables instead of just one, and the predictor variables are unobserved instead of observed. Hence we see that factor analysis is indeed a linear factor model.

The factor analysis model as specified by (2.1.1), (2.1.2), and (2.1.3) is a **frequentist** (rather than Bayesian) model because the parameters, $\mathbf{W}^\top, \psi_1, \cdots, \psi_p$, are considered as fixed (albeit unknown). It can be regarded as a hierarchical model where we first draw z_{ik} based on (2.1.2) and then draw X based on (2.1.1) and (2.1.3). The components of the model are:

- $\mathbf{W}^\top, \psi_1, \cdots, \psi_p$ are the fixed but unknown parameters,
- Z is the missing data or latent variables, and
- X is the observed data.

Without loss of generality, and in accordance with common practice, we have assumed that before factor analysis is applied to any X, each column of X is centered to have mean zero. This assumption allows us to simplify the notation and mathematical derivations by omitting the nuisance mean parameter in (2.1.1).

2.1.2 Per-observation viewpoint

Focusing our attention on the ith observation in (2.1.1), we have

$$x_i^\top = z_i^\top \mathbf{W}^\top + \varepsilon_i^\top, \ i = 1, \cdots, n, \tag{2.1.4}$$

where $x_i^\top$, $z_i^\top$, and $\varepsilon_i^\top$ denote the ith row of X, Z, and ε respectively.

Transposing (2.1.4), we have

$$\underset{p\times 1}{x_i} = \underset{p\times K}{\mathbf{W}} \ \underset{K\times 1}{z_i} + \underset{p\times 1}{\varepsilon_i}, \ i = 1, \cdots, n. \tag{2.1.5}$$

Further, our assumption in (2.1.2) becomes

$$z_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I}_K), \ i = 1, \cdots, n, \tag{2.1.6}$$

and our assumption in (2.1.3) becomes

$$\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \mathbf{\Psi} \coloneqq \operatorname{diag}(\psi_1, \cdots, \psi_p) = \begin{bmatrix} \psi_1 & & \\ & \ddots & \\ & & \psi_p \end{bmatrix}\right), \ i = 1, \cdots, n, \tag{2.1.7}$$

where ε_i and z_i are independent.

2.1.3 Interpretations of W and $\mathbf{W}^\top$

There are at least three interpretations of $\mathbf{W}$ and $\mathbf{W}^\top$:

- Matrix factorization. From (2.1.1), we see that X can be approximately factorized as $Z\,\mathbf{W}^\top$.
- Weight matrix, also known as effect size matrix or loading matrix. From (2.1.1), we see that each column (variable) of X is a linear combination of the hidden factors (plus some error), where the weights are the corresponding entries of $\mathbf{W}^\top$.
- Basis vectors. From (2.1.5), we see that each x_i is a linear combination of the columns of $\mathbf{W}$ (plus some error), where the coefficients are the entries of z_i.

2.1.4 Maximum likelihood estimation

Historically, many methods have been developed for estimating the parameters ($\mathbf{W}$ and $\mathbf{\Psi}$) in factor analysis. In particular, when normality is not assumed in (2.1.6) and (2.1.7), methods such as the principal component method and the principal factor method may be used [10, 11]. However, it is standard practice nowadays to indeed assume normality and

estimate the parameters by maximizing the likelihood. Such maximum likelihood estimation involves an iterative procedure and a fair amount of algebra, which we will not repeat here [see 12–14] . We will focus on the overall idea instead.

Let's begin by finding the joint distribution of z_i and x_i, which will be useful later for deriving the conditional distribution of z_i given x_i (Section 2.1.6). The marginal distribution of z_i is given by (2.1.6). Thus, by (2.1.5) and (2.1.7), the marginal distribution of x_i is

$$x_i \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \mathbf{W}\,\mathbf{W}^\top + \mathbf{\Psi}\right),\ i = 1, \cdots, n. \tag{2.1.8}$$

Therefore, the joint distribution of z_i and x_i is

$$\begin{bmatrix} z_i \\ x_i \end{bmatrix} \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \begin{bmatrix} \mathbf{I}_K & \mathbf{W}^\top \\ \mathbf{W} & \mathbf{W}\,\mathbf{W}^\top + \mathbf{\Psi} \end{bmatrix}\right),\ i = 1, \cdots, n. \tag{2.1.9}$$

This is because

$$\begin{aligned}
\operatorname{Cov}[z_i, x_i] &= \mathbb{E}\left[(z_i - \mathbb{E}[z_i])\,(x_i - \mathbb{E}[x_i])^\top\right] && \text{by definition} && (2.1.10)\\
&= \mathbb{E}\left[z_i x_i^\top\right] && && (2.1.11)\\
&= \mathbb{E}\left[z_i\,(\mathbf{W}\,z_i + \varepsilon_i)^\top\right] && \text{plugging in (2.1.5)} && (2.1.12)\\
&= \mathbb{E}\left[z_i z_i^\top\,\mathbf{W}^\top\right] + \mathbb{E}\left[z_i \varepsilon_i^\top\right] && && (2.1.13)\\
&= \mathbb{E}\left[z_i z_i^\top\,\mathbf{W}^\top\right] && \text{since } \varepsilon_i \perp z_i && (2.1.14)\\
&= \mathbb{E}\left[z_i z_i^\top\right]\mathbf{W}^\top && && (2.1.15)\\
&= \operatorname{var}[z_i]\,\mathbf{W}^\top && && (2.1.16)\\
&= \mathbf{I}_K\,\mathbf{W}^\top && && (2.1.17)\\
&= \mathbf{W}^\top. && && (2.1.18)
\end{aligned}$$

Using (2.1.8), we can write down the observed data likelihood of the parameters as

$$\mathcal{L}(\mathbf{W}, \mathbf{\Psi}) = \prod_{i=1}^{n} f(x_i; \mathbf{W}, \mathbf{\Psi}) \tag{2.1.19}$$

$$= \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{p}{2}} |\mathbf{W}\mathbf{W}^\top + \mathbf{\Psi}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2} x_i^\top \left(\mathbf{W}\mathbf{W}^\top + \mathbf{\Psi}\right)^{-1} x_i\right\}. \tag{2.1.20}$$

Therefore, the overall idea of maximum likelihood estimation is to maximize (2.1.20) with respect to the parameters. Of note, Rubin and Thayer [14] use the EM algorithm to solve this maximization problem, treating Z as the missing data (Section 2.1.1).

2.1.5 Rotation

Suppose we have found $\widehat{\mathbf{W}}_1$ and $\widehat{\mathbf{\Psi}}$ to be our MLE estimates of $\mathbf{W}$ and $\mathbf{\Psi}$. Looking at (2.1.20), we see that

$$\mathcal{L}\left(\widehat{\mathbf{W}}_1, \widehat{\mathbf{\Psi}}\right) = \mathcal{L}\left(\widehat{\mathbf{W}}_2, \widehat{\mathbf{\Psi}}\right), \tag{2.1.21}$$

where

$$\widehat{\mathbf{W}}_2 \coloneqq \widehat{\mathbf{W}}_1 \mathbf{G} \tag{2.1.22}$$

and $\mathbf{G}$ is any $K \times K$ orthogonal matrix. This is because

$$\widehat{\mathbf{W}}_2 \widehat{\mathbf{W}}_2^\top + \widehat{\mathbf{\Psi}} = (\widehat{\mathbf{W}}_1 \mathbf{G})(\mathbf{G}^\top \widehat{\mathbf{W}}_1^\top) + \widehat{\mathbf{\Psi}} \qquad \text{plugging in (2.1.22)} \tag{2.1.23}$$

$$= \widehat{\mathbf{W}}_1 \widehat{\mathbf{W}}_1^\top + \widehat{\mathbf{\Psi}}. \tag{2.1.24}$$

Transposing (2.1.22), we see that

$$\widehat{\mathbf{W}}_2^\top = \mathbf{G}^\top \widehat{\mathbf{W}}_1^\top, \tag{2.1.25}$$

so that $\widehat{\mathbf{W}}_2^\top$ can be interpreted as a rotated version of $\widehat{\mathbf{W}}_1^\top$ (recall that orthogonal matrices represent rotations. More precisely, they represent rotations, reflections, and compositions thereof).

Therefore, the second step of inference, after obtaining the initial MLE estimates of $\mathbf{W}$ and $\mathbf{\Psi}$, is to choose the "best" $\mathbf{W}$ estimate according to some predetermined criterion. There are several possible criteria [see 15, 16, for two non-exhaustive lists], but all of them are designed to make the structure of the $\mathbf{W}$ estimate as simple as possible, with most entries either close to zero or far from zero and few entries taking intermediate values. The most popular criterion among these is varimax [17], which is implemented in most software packages for factor analysis.

2.1.6 Hidden factor prediction

Suppose that through maximum likelihood estimation and rotation, we have decided on $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{\Psi}}$ as our parameter estimates. Then, the third (optional) step of inference in factor analysis is hidden factor prediction. From (2.1.9), we know that

$$\begin{aligned} z_i \mid x_i, \mathbf{W}, \mathbf{\Psi} \sim \mathcal{N}\Big(& \mathbf{W}^\top \left(\mathbf{W}\,\mathbf{W}^\top + \mathbf{\Psi}\right)^{-1} x_i, \\ & \mathbf{I}_K - \mathbf{W}^\top \left(\mathbf{W}\,\mathbf{W}^\top + \mathbf{\Psi}\right)^{-1} \mathbf{W} \Big). \end{aligned} \tag{2.1.26}$$

Therefore, plugging in our parameter estimates, we may predict z_i to be

$$\widehat{z}_i = \mathbb{E}\left[z_i \mid x_i, \widehat{\mathbf{W}}, \widehat{\mathbf{\Psi}}\right] \tag{2.1.27}$$

$$= \widehat{\mathbf{W}}^\top \left(\widehat{\mathbf{W}}\widehat{\mathbf{W}}^\top + \widehat{\mathbf{\Psi}}\right)^{-1} x_i, \tag{2.1.28}$$

which is known as Thomson's factor scores [18]. Alternatively, Barlett's factor scores [19] may be used, which are derived from weighted least squares and are less congruent with the underlying model of factor analysis. It is for this reason that we omit the details of Barlett's factor scores here.

2.1.7 Discussion (factor analysis)

Factor analysis was originally developed by psychologists as part of an attempt to understand "intelligence" in the early 1900s. The method gained more recognition in the early 1940s when attention was brought to one particular form of factor analysis, namely that based on maximum likelihood estimation [20]. To date, most applications of factor analysis have been in psychology and the social sciences.

Throughout its history, factor analysis has provoked rather turbulent controversy. Besides the large number of assumptions made (Section 2.1.1 and Section 2.1.2), there are several statistical concerns. For example, it is difficult to determine K, the number of hidden factors, and the results may vary significantly depending on the value of K. Further, for a given K, different methods of rotation can produce results that look quite different. This leads to the danger that practitioners may try different values of K and different methods of rotation in order to obtain results that conform to their preconceived ideas [21].

Despite the controversy, the generative model in factor analysis has been extended to form the basis of many other modern methods including generative adversarial network (GAN) and variational autoencoder (VAE). In the next sections, we will discuss other linear factor models and compare them with factor analysis.

2.2 Principal component analysis (PCA)

Principal component analysis (PCA) [11, 22] is a well-established dimension reduction method that has many applications. The method differs from factor analysis in that it does not re-

quire a probabilistic model. In this section, we give a brief summary of the PCA algorithm and its derivation and interpretation. In the next section, we will discuss probabilistic PCA, which offers a probabilistic interpretation of PCA and establishes an explicit connection between PCA and factor analysis.

2.2.1 PCA algorithm

Let X denote the $n \times p$ **observed data** matrix that is observation by variable (i.e., feature). We assume that each column of X has been standardized, i.e., centered to have mean zero and scaled to have variance one. That is, X satisfies

$$\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0,\ j = 1, \cdots, p \tag{2.2.1}$$

and

$$\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1,\ j = 1, \cdots, p, \tag{2.2.2}$$

where x_{ij} denotes the ijth entry of X.

The PCA algorithm consists of two steps. In the first step, we calculate the sample covariance matrix $\widehat{\Sigma}$ and perform eigendecomposition on it:

$$\widehat{\Sigma} = \frac{1}{n}X^\top X \qquad \text{definition of sample covariance matrix} \tag{2.2.3}$$

$$\coloneqq Q\Lambda Q^\top, \qquad \text{eigendecomposition} \tag{2.2.4}$$

where

$$\underset{p\times p}{Q} = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_p \\ | & & | \end{bmatrix} \tag{2.2.5}$$

is an orthogonal matrix whose columns are eigenvectors of $\widehat{\Sigma}$, and

$$\underset{p\times p}{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix}, \ \lambda_1 \geq \cdots \geq \lambda_p \geq 0, \tag{2.2.6}$$

is a diagonal matrix whose diagonal entries are the corresponding eigenvalues of $\widehat{\Sigma}$. We know that $\widehat{\Sigma}$ is orthogonally diagonalizable because it is a symmetric matrix (recall the spectral theorem [23]: a matrix is orthogonally diagonalizable if and only if it is symmetric). The eigenvalues are all non-negative because $\widehat{\Sigma}$ is positive semidefinite.

In the second step, we calculate Z as the following:

$$Z = XQ, \tag{2.2.7}$$

where the columns of Z are called the **principal components** (PCs) or scores, and Q is called the **loading matrix** or rotation matrix. It is worth noting that some authors may refer to $q_1, \cdots, q_p$ as the PCs. This usage is confusing and should be avoided [24].

A technical complication is that although Λ is unique, Q is not unique. In general, we may multiply an arbitrary subset of the eigenvectors by -1 and the eigendecomposition would still hold. In addition, if $\lambda_j = \cdots = \lambda_{j'}$ for any $j,\ j' \in \{1, \cdots, p\},\ j < j'$, then we may replace $q_j, \cdots, q_{j'}$ with any other set of orthonormal vectors that span the same subspace. Fortunately, the non-uniqueness of Q has minimal real consequences on Z beyond the signs of the PCs. The reason for this is twofold. First, if $\lambda_j = \cdots = \lambda_{j'} = 0$, then the corresponding PCs are all zero and thus all dropped (Section 2.2.3). Second, in real data

sets, non-zero eigenvalues are rarely, if ever, repeated.

The above two steps conclude the PCA algorithm. In practice, however, singular value decomposition (SVD) of the data matrix (as opposed to eigendecomposition of the sample covariance matrix) provides a more computationally efficient way of finding the loading matrix and the PCs. Suppose a singular value decomposition of X is

$$\underset{n\times p}{X} = \underset{n\times n}{U}\ \underset{n\times p}{\Sigma}\ \underset{p\times p}{V^\top}, \tag{2.2.8}$$

where U is an orthogonal matrix whose columns are left singular vectors of X, V is an orthogonal matrix whose columns are right singular vectors of X, and Σ is a rectangular diagonal matrix whose diagonal entries are singular values of X (all chosen to be non-negative) arranged in descending order. Then, we have

$$\widehat{\Sigma} = \frac{1}{n}X^\top X = \frac{1}{n}\left(U\Sigma V^\top\right)^\top\left(U\Sigma V^\top\right) \qquad \text{plugging in (2.2.8)} \tag{2.2.9}$$

$$= \frac{1}{n}V\Sigma^\top U^\top U\Sigma V^\top \tag{2.2.10}$$

$$= V\left(\frac{1}{n}\Sigma^\top\Sigma\right)V^\top, \tag{2.2.11}$$

which constitutes an eigendecomposition of $\widehat{\Sigma}$ where the eigenvalues are arranged in descending order.

Therefore, the loading matrix is given by V, and the PCs are given by

$$Z = XV \qquad \text{plugging in (2.2.7)} \tag{2.2.12}$$

$$= \left(U\Sigma V^\top\right)V \qquad \text{plugging in (2.2.8)} \tag{2.2.13}$$

$$= U\Sigma, \tag{2.2.14}$$

from which it is evident that the non-zero PCs are unnormalized versions of left singular vectors of X, or equivalently, right singular vectors of $X^\top$.

2.2.2 Derivation

The most common derivation of PCA is based on maximum variance [25]. Define $\alpha_1^*, \cdots, \alpha_p^* \in \mathbb{R}^p$ sequentially as

$$\alpha_1^* = \underset{\alpha_1 \in \mathbb{R}^p}{\arg\max} \operatorname{Var}(X\alpha_1) \qquad \text{subject to } \|\alpha_1\|_2 = 1, \tag{2.2.15}$$

$$\alpha_2^* = \underset{\alpha_2 \in \mathbb{R}^p}{\arg\max} \operatorname{Var}(X\alpha_2) \qquad \text{subject to } \|\alpha_2\|_2 = 1,\ \alpha_2^\top \alpha_1^* = 0, \tag{2.2.16}$$

$$\vdots$$

$$\alpha_p^* = \underset{\alpha_p \in \mathbb{R}^p}{\arg\max} \operatorname{Var}(X\alpha_p) \qquad \text{subject to } \|\alpha_p\|_2 = 1,\ \alpha_p^\top \alpha_j^* = 0\ \forall j < p. \tag{2.2.17}$$

The principal components of X are defined as $X\alpha_1^*, \cdots, X\alpha_p^*$. That is, they are defined sequentially as the linear combinations of the columns of X with maximum variance, subject to certain constraints. It can then be shown that $\alpha_1^*, \cdots, \alpha_p^*$ are given by $q_1, \cdots, q_p$ respectively, where $q_1, \cdots, q_p$ are eigenvectors of $\widehat{\Sigma}$ as defined in (2.2.5). The key to the proof lies in the fact that for a constant vector $\alpha \in \mathbb{R}^p$, $X\alpha$ is an n-dimensional vector with zero mean, which means that

$$\operatorname{Var}(X\alpha) = \frac{1}{n}(X\alpha)^\top X\alpha \qquad \text{definition of sample variance} \tag{2.2.18}$$

$$= \alpha^\top \left(\frac{1}{n} X^\top X\right) \alpha. \tag{2.2.19}$$

A complementary property of PCA, which is closely related to the original discussion of Pearson [26], is the minimum reconstruction error property. Given $K < p$, define Q_K as the matrix that contains the first K columns of Q. That is,

$$\underset{p \times K}{Q_K} \coloneqq \begin{bmatrix} | & & | \\ q_1 & \cdots & q_K \\ | & & | \end{bmatrix}. \tag{2.2.20}$$

The minimum reconstruction error property of PCA states that Q_K is a global minimizer of the loss function

$$\begin{aligned}\mathcal{J}\left(\widetilde{Q}_K\right) &\coloneqq \left|\left|\left|X - X\widetilde{Q}_K\widetilde{Q}_K^\top\right|\right|\right|_F^2 \\ &= \sum_{i=1}^n \left\|x_i^\top - x_i^\top\widetilde{Q}_K\widetilde{Q}_K^\top\right\|_2^2 = \sum_{i=1}^n \left\|x_i - \widetilde{Q}_K\widetilde{Q}_K^\top x_i\right\|_2^2,\end{aligned} \tag{2.2.21}$$

where $\widetilde{Q}_K$ denotes an arbitrary $p \times K$ matrix whose columns are orthonormal, $|||\cdot|||_F$ denotes the Frobenius norm of a matrix, and $x_i^\top$ denotes the ith row of X. Since $\widetilde{Q}_K\widetilde{Q}_K^\top x_i$ represents the (orthogonal) projection of x_i onto the subspace spanned by the columns of $\widetilde{Q}_K$, (2.2.21) measures the total error when approximating each x_i with its projection onto the subspace spanned by the columns of $\widetilde{Q}_K$.

2.2.3 Interpretation

Multiplying both sides of (2.2.7) by $Q^\top$, we have

$$X = ZQ^\top, \tag{2.2.22}$$

which is reminiscent of (2.1.1) in factor analysis.

Focusing our attention on the ith observation, we have

$$x_i^\top = z_i^\top Q^\top,\ i = 1, \cdots, n, \tag{2.2.23}$$

where $x_i^\top$ and $z_i^\top$ denote the ith row of X and Z respectively.

Transposing (2.2.23), we have

$$x_i = Qz_i,\ i = 1, \cdots, n, \tag{2.2.24}$$

which is reminiscent of (2.1.5) in factor analysis. (2.2.24) means that z_i is x_i in the coordinate system of Q.

A central idea of PCA is that when the data is viewed in the new coordinate system, the total variance in the original data is preserved, but variance is now concentrated on the first PCs. Further, the PCs are uncorrelated with each other. Therefore, we can capture a large portion of the variance in the original data by keeping only the first K PCs, $K < p$, thus achieving dimension reduction. Specifically, we claim that

$$\sum_{j=1}^{p} \mathrm{Var}\left(X_j\right) = \sum_{j=1}^{p} \mathrm{Var}\left(Z_j\right), \tag{2.2.25}$$

$$\mathrm{Var}\left(Z_j\right) = \lambda_j, \ j = 1, \cdots, p, \tag{2.2.26}$$

and

$$\mathrm{Cov}\left(Z_j, Z_{j'}\right) = 0, \ j, \ j' = 1, \cdots, p, \ j \neq j', \tag{2.2.27}$$

where X_j denotes the jth column of X (the jth original variable) and Z_j denotes the jth column of Z (the jth PC).

We prove (2.2.26) and (2.2.27) by calculating $\widehat{\Sigma}_Z$, the sample covariance matrix of Z:

$$\widehat{\Sigma}_Z = \frac{1}{n} Z^\top Z \qquad \text{definition of sample covariance matrix} \tag{2.2.28}$$

$$= \frac{1}{n} \left(XQ\right)^\top XQ \qquad \text{plugging in (2.2.7)} \tag{2.2.29}$$

$$= Q^\top \left(\frac{1}{n} X^\top X\right) Q \tag{2.2.30}$$

$$= Q^\top \left(Q \Lambda Q^\top\right) Q \qquad \text{plugging in (2.2.4)} \tag{2.2.31}$$

$$= \Lambda. \tag{2.2.32}$$

(2.2.25) can be proven by the following:

$$\sum_{j=1}^{p} \mathrm{Var}\left(X_j\right) = \mathrm{Tr}\left(\widehat{\Sigma}\right) \qquad \text{by definition} \tag{2.2.33}$$

$$= \mathrm{Tr}\left(Q\Lambda Q^{\top}\right) \qquad \text{plugging in (2.2.4)} \tag{2.2.34}$$

$$= \mathrm{Tr}\left(\Lambda Q^{\top} Q\right) \qquad \text{cyclic property of trace} \tag{2.2.35}$$

$$= \mathrm{Tr}\left(\Lambda\right) \tag{2.2.36}$$

$$= \sum_{j=1}^{p} \mathrm{Var}\left(Z_j\right). \qquad \text{by (2.2.26)} \tag{2.2.37}$$

Because of (2.2.25) and (2.2.26), we may define and calculate the proportion of variance in the original data explained by the jth PC as

$$\frac{\lambda_j}{\sum_{j'=1}^{p} \mathrm{Var}\left(X_{j'}\right)} = \frac{\lambda_j}{\sum_{j'=1}^{p} \mathrm{Var}\left(Z_{j'}\right)} = \frac{\lambda_j}{\sum_{j'=1}^{p} \lambda_{j'}} \tag{2.2.38}$$

and the cumulative proportion of variance explained by the first K PCs as

$$\frac{\sum_{j=1}^{K} \lambda_j}{\sum_{j'=1}^{p} \lambda_{j'}}. \tag{2.2.39}$$

(2.2.38) and (2.2.39) provide a basis for deciding the number of PCs to keep.

2.2.4 Similarity to factor analysis

When we dimension reduce X to the first K columns of Z, we are approximating $X = ZQ^\top$ (Equation 2.2.22) with

$$X \approx \underline{Z}_K Q_K^\top \tag{2.2.40}$$

$$\coloneqq \begin{bmatrix} | & & | \\ Z_1 & \cdots & Z_K \\ | & & | \end{bmatrix} \begin{bmatrix} — & q_1^\top & — \\ & \vdots & \\ — & q_K^\top & — \end{bmatrix} \tag{2.2.41}$$

$$= \begin{bmatrix} | & & | \\ \frac{1}{\sqrt{\lambda_1}} Z_1 & \cdots & \frac{1}{\sqrt{\lambda_K}} Z_K \\ | & & | \end{bmatrix} \begin{bmatrix} — & \sqrt{\lambda_1} q_1^\top & — \\ & \vdots & \\ — & \sqrt{\lambda_K} q_K^\top & — \end{bmatrix}, \tag{2.2.42}$$

where we have defined $\underline{Z}_K$ and Q_K to be the matrices that contain the first K columns of Z and Q respectively (see also (2.2.20)). We use an underscore in $\underline{Z}_K$ to avoid ambiguity with the notation for the PCs. The approximation in (2.2.40) is "best" in the sense of minimum reconstruction error (Section 2.2.2). The only difference between (2.2.41) and (2.2.42) is that in (2.2.42), we have normalized the PCs to have unit variance and scaled $q_1^\top, \cdots, q_K^\top$ accordingly.

Comparing (2.2.40) to (2.1.1), we see that $\underline{Z}_K$ in PCA is analogous to the hidden factor matrix in factor analysis, and Q_K in PCA is analogous to $\mathbf{W}$ in factor analysis. Thus, Q_K and $Q_K^\top$ in PCA can be interpreted in the same ways as $\mathbf{W}$ and $\mathbf{W}^\top$ in factor analysis: matrix factorization, weight matrix, and basis vectors (Section 2.1.3). An explicit connection between PCA and factor analysis will be explored in Section 2.3.

2.3 Probabilistic PCA (PPCA)

Probabilistic PCA (PPCA) [27] is based on a generative statistical model that is very similar to the factor analysis model. It offers a probabilistic interpretation of PCA and establishes an explicit connection between PCA and factor analysis.

In this section, we will use mostly the same notations as we did in Section 2.1, but for clarity, we will redefine the notations here. Let X denote the $n \times p$ **observed data** matrix that is observation by variable (i.e., feature); let Z denote the $n \times K$ **hidden factor matrix**; let $\mathbf{W}^\top$ denote the $K \times p$ **weight matrix**; finally, let ε denote the **error matrix**.

2.3.1 PPCA model

Recall the factor analysis model as specified by (2.1.5), (2.1.6), and (2.1.7). PPCA simplifies the model by assuming $\psi_1 = \cdots = \psi_p$. Hence the PPCA model assumes

$$\underset{p\times 1}{x_i} = \underset{p\times K}{\mathbf{W}}\ \underset{K\times 1}{z_i} + \underset{p\times 1}{\varepsilon_i}\,,\ i = 1, \cdots, n, \tag{2.3.1}$$

where $x_i^\top$, $z_i^\top$, and $\varepsilon_i^\top$ denote the ith row of X, Z, and ε respectively,

$$z_i \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \mathbf{I}_K\right),\ i = 1, \cdots, n, \tag{2.3.2}$$

and

$$\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \sigma^2 \mathbf{I}_p\right),\ i = 1, \cdots, n, \tag{2.3.3}$$

where ε_i and z_i are independent. Therefore, the parameters in the PPCA model are $\mathbf{W}$ and σ^2.

Applying (2.1.9), we know that the joint distribution of z_i and x_i is

$$\begin{bmatrix} z_i \\ x_i \end{bmatrix} \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \begin{bmatrix} \mathbf{I}_K & \mathbf{W}^\top \\ \mathbf{W} & \mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p \end{bmatrix}\right), \; i = 1, \cdots, n. \tag{2.3.4}$$

Applying (2.1.26), we have

$$\begin{aligned} z_i \mid x_i, \mathbf{W}, \sigma^2 \sim \mathcal{N}\Big(& \mathbf{W}^\top \left(\mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p\right)^{-1} x_i, \\ & \mathbf{I}_K - \mathbf{W}^\top \left(\mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p\right)^{-1} \mathbf{W} \Big), \end{aligned} \tag{2.3.5}$$

which can be simplified as

$$\begin{aligned} z_i \mid x_i, \mathbf{W}, \sigma^2 \sim \mathcal{N}\Big(& \left(\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K\right)^{-1} \mathbf{W}^\top x_i, \\ & \sigma^2 \left(\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K\right)^{-1} \Big). \end{aligned} \tag{2.3.6}$$

We can show that the two expressions for the conditional mean are equal, i.e.,

$$\mathbf{W}^\top \left(\mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p\right)^{-1} = \left(\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K\right)^{-1} \mathbf{W}^\top, \tag{2.3.7}$$

by moving the matrix inverses to the opposite sides of the equation, and we can show that the two expressions for the conditional variance are equal by using the Woodbury matrix identity. Alternatively, (2.3.6) may be derived directly by using Bayes' rule.

2.3.2 Maximum likelihood estimation

From (2.3.4), we know that the marginal distribution of x_i is

$$x_i \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p\right), \; i = 1, \cdots, n. \tag{2.3.8}$$

Therefore, we can write down the observed data likelihood of the parameters as

$$\mathcal{L}\left(\mathbf{W}, \sigma^2\right) = \prod_{i=1}^{n} f\left(x_i; \mathbf{W}, \sigma^2\right) \tag{2.3.9}$$

$$= \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{p}{2}} \left|\, \mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p \right|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} x_i^\top \left(\mathbf{W}\,\mathbf{W}^\top + \sigma^2 \mathbf{I}_p\right)^{-1} x_i \right\}. \tag{2.3.10}$$

In contrast to factor analysis (Section 2.1.4), in PPCA, the maximum likelihood estimates of $\mathbf{W}$ and σ^2 can be obtained analytically. Before giving the MLE solutions, define

$$\widehat{\Sigma} \coloneqq \frac{1}{n} X^\top X \qquad \text{sample covariance matrix} \tag{2.3.11}$$

$$\coloneqq Q \Lambda Q^\top, \qquad \text{eigendecomposition} \tag{2.3.12}$$

where Q is an orthogonal matrix whose columns are eigenvectors of $\widehat{\Sigma}$, and

$$\underset{p \times p}{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix}, \ \lambda_1 \geq \cdots \geq \lambda_p \geq 0, \tag{2.3.13}$$

is a diagonal matrix whose diagonal entries are the corresponding eigenvalues of $\widehat{\Sigma}$. These are the same steps that we took in PCA (Section 2.2.1). The MLE estimates of $\mathbf{W}$ and σ^2 are then given by

$$\widehat{\sigma}^2_{\text{MLE}} = \frac{1}{p-K} \sum_{j=K+1}^{p} \lambda_j \tag{2.3.14}$$

and

$$\underset{p \times K}{\widehat{\mathbf{W}}_{\text{MLE}}} = \underset{p \times K}{Q_K} \left(\underset{K \times K}{\Lambda_K} - \widehat{\sigma}^2_{\text{MLE}} \mathbf{I}_K \right)^{\frac{1}{2}} \underset{K \times K}{\mathbf{G}}, \tag{2.3.15}$$

where Q_K denotes the matrix that contains the first K columns of Q, Λ_K denotes the $K \times K$ diagonal matrix that is on the top-left corner of Λ, and $\mathbf{G}$ is any $K \times K$ orthogonal matrix [27].

Thus, using (2.3.6), we may predict z_i to be

$$\widehat{z}_i = \mathbb{E}\left[z_i \mid x_i, \widehat{\mathbf{W}}_{\text{MLE}}, \widehat{\sigma}^2_{\text{MLE}}\right] \tag{2.3.16}$$

$$= \left(\widehat{\mathbf{W}}_{\text{MLE}}^\top \widehat{\mathbf{W}}_{\text{MLE}} + \widehat{\sigma}^2_{\text{MLE}} \mathbf{I}_K\right)^{-1} \widehat{\mathbf{W}}_{\text{MLE}}^\top x_i. \tag{2.3.17}$$

2.3.3 Connection to PCA

To recover PCA from PPCA, assume σ^2 is known and let $\sigma^2 \to 0$. When σ^2 is known, the only unknown parameter in the model is $\mathbf{W}$, so (2.3.15) becomes

$$\underset{p\times K}{\widehat{\mathbf{W}}_{\text{MLE}}} = \underset{p\times K}{Q_K} \underset{K\times K}{\left(\Lambda_K - \sigma^2 \mathbf{I}_K\right)^{\frac{1}{2}}} \underset{K\times K}{\mathbf{G}}, \tag{2.3.18}$$

and (2.3.17) becomes

$$\widehat{z}_i = \left(\widehat{\mathbf{W}}_{\text{MLE}}^\top \widehat{\mathbf{W}}_{\text{MLE}} + \sigma^2 \mathbf{I}_K\right)^{-1} \widehat{\mathbf{W}}_{\text{MLE}}^\top x_i. \tag{2.3.19}$$

Therefore, as $\sigma^2 \to 0$, we have

$$\widehat{\mathbf{W}}_{\text{MLE}} \to Q_K \Lambda_K^{\frac{1}{2}} \mathbf{G}, \tag{2.3.20}$$

and thus

$$\widehat{z}_i \to \left(\left(Q_K\Lambda_K^{\frac{1}{2}}\mathbf{G}\right)^\top Q_K\Lambda_K^{\frac{1}{2}}\mathbf{G}\right)^{-1}\left(Q_K\Lambda_K^{\frac{1}{2}}\mathbf{G}\right)^\top x_i \tag{2.3.21}$$

$$= \left(\mathbf{G}^\top\Lambda_K^{\frac{1}{2}}Q_K^\top Q_K\Lambda_K^{\frac{1}{2}}\mathbf{G}\right)^{-1}\mathbf{G}^\top\Lambda_K^{\frac{1}{2}}Q_K^\top x_i \tag{2.3.22}$$

$$= \mathbf{G}^\top\Lambda_K^{-1}\mathbf{G}\mathbf{G}^\top\Lambda_K^{\frac{1}{2}}Q_K^\top x_i \tag{2.3.23}$$

$$= \mathbf{G}^\top\Lambda_K^{-\frac{1}{2}}Q_K^\top x_i. \tag{2.3.24}$$

In other words, we have

$$\widehat{z}_i^\top \to x_i^\top Q_K\Lambda_K^{-\frac{1}{2}}\mathbf{G}, \tag{2.3.25}$$

$$\widehat{Z} \to XQ_K\Lambda_K^{-\frac{1}{2}}\mathbf{G}, \tag{2.3.26}$$

and

$$\widehat{\mathbf{W}}_{\text{MLE}}^\top \to \mathbf{G}^\top\Lambda_K^{\frac{1}{2}}Q_K^\top. \tag{2.3.27}$$

(2.3.26) and (2.3.27) recover the two matrices in (2.2.42) in PCA when we take $\mathbf{G} = \mathbf{I}_K$.

In summary, if we assume that the error variances are equal and known in factor analysis, we can recover PCA from factor analysis as the error variance approaches zero.

CHAPTER 3

New methods

3.1 Background

In this section, I will introduce the relevant biological background regarding differential gene expression (DE) analysis and expression quantitative trait locus (eQTL) analysis with hidden covariates. Afterwards, I will introduce surrogate variable analysis (SVA) [3, 4], probabilistic estimation of expression residuals (PEER) [5, 6], and hidden covariates with prior (HCP) [7], which are widely-used modern methods designed for capturing hidden covariates in DE analysis (SVA) or eQTL analysis (PEER and HCP).

As much as I try to keep the notations consistent, the notations in Chapter 3 will be slightly different from those in Chapter 2 because the conventions in the literature are different. A summary of the main notations used in Chapter 3 is given in Table 3.1.

	DE	**eQTL**
Gene expression matrix (responses)	Y, $n \times p$, sample by gene	
Variable(s) of interest	X_0, $n \times K_0$, K_0 small	S, $n \times q$, q large
Known covariates	X_1, $n \times K_1$	
Hidden covariates	X_2, $n \times K$	
Hidden covariate inference method(s)	SVA	(SVA), PEER, HCP

Table 3.1: Summary of main notations used in Chapter 3. SVA was designed for capturing hidden covariates in DE analysis but is sometimes used for capturing hidden covariates in eQTL analysis as well.

3.1.1 Differential gene expression (DE) analysis with hidden covariates

Let Y denote the $n \times p$ **gene expression** matrix that is sample by gene (observation by feature); the ijth entry represents the expression level of gene j in the ith biological sample. Let X_0 denote the $n \times K_0$ matrix of **variable(s) of interest**. Let X_1 denote the $n \times K_1$ **known covariate** matrix. Lastly, let X_2 denote the $n \times K$ **hidden covariate** matrix. In both DE studies and eQTL studies, X_2 represents unknown batch effects, technical confounders, and/or biological confounders that affect the measured gene expression levels. It is broadly recognized in the field that failure to account for hidden covariates can lead to loss of power and/or precision in detecting biological signals — assuming that the hidden covariates indeed exist and confound the relationship between the gene expression levels and the variables of interest, that is (a similar idea is explored in the field of causal inference [see 28, for example]).

In DE analysis, the research question is whether the K_0 variables of interest are *collectively* associated with the expression level of each gene, controlling for the effect of the known and hidden covariates. For example, suppose $K_0 = 1$ and X_0 is a binary variable representing disease status. For each gene, we would like to conduct a multiple linear regression with the gene expression vector as the response variable and X_0, X_1, and X_2 as the predictors — and test the null hypothesis that the coefficient corresponding to X_0 is zero (given X_1 and X_2). Since X_2 is unknown, a two-step approach is usually used: first, infer X_2 using SVA or other methods; second, use the inferred X_2 in the subsequent linear regression analysis.

In general, K_0 may be greater than 1 (though it is usually small). Suppose $K_0 = 2$ and X_0 consists of two binary variables that together represent a categorical variable with three levels, e.g., a disease status variable with three levels. In this case, for each gene, we would like to conduct a multiple linear regression with X_0, X_1, and X_2 as the predictors and test the null hypothesis that the coefficients corresponding to the columns of X_0 are *all* zero (given X_1 and X_2).

3.1.2 Expression quantitative trait locus (eQTL) analysis with hidden covariates

Define Y, X_1, and X_2 in the same way as in Section 3.1.1 (see Table 3.1 for a summary). In addition, let S denote the $n \times q$ single nucleotide polymorphism (SNP) **genotype matrix** that is sample by SNP. SNPs are the most common type of genetic variation among people. The $i\ell$th entry of S is 0, 1, or 2 and encodes the number of minor allele copies that the ith individual has at the ℓth SNP in the genome.

To simplify the matter, we may think of S as a large matrix that stores measurements of each person's genetic profile, each column (SNP) being a measurement. The goal of an eQTL study is to test the significance of the association between each gene's expression level and each SNP, controlling for the effect of the known and hidden covariates — to see whether the SNP may have a role in regulating the gene's expression level. As in DE analysis, since X_2 is unknown, a two-step approach is usually used: first, infer X_2 using PEER or other methods; second, use the inferred X_2 in the subsequent linear regression analysis.

What I have described in the above paragraph, if done, would encompass both cis-eQTL analysis and trans-eQTL analysis and would typically be very computationally expensive in real data. This is because in real data, the total number of SNPs in the genome, q, can be as large as tens of millions. In practice, most studies focus on cis-eQTL analysis and only test the significance of the association between the expression level of each gene and each of its *local* SNPs, i.e., SNPs that are on the same chromosome as the gene and located close to the gene (e.g., within one megabase from the transcription start site of the gene).

3.2 Surrogate variable analysis (SVA)

Surrogate variable analysis (SVA) [3, 4] is a popular method for estimating hidden covariates in DE analysis and to a lesser extent, in eQTL analysis. Historically, there have been two versions of the SVA method: two-step SVA [3] and iteratively reweighted SVA (IRW-SVA)

[4]. Currently, the main SVA method available in the R package SVA [29] is IRW-SVA; the package documentation states that two-step SVA is included in the package primarily for backward-compatibility purposes. Therefore, in this work, I will focus on IRW-SVA exclusively.

Although more than a dozen papers have been published on variations and extensions of SVA [see 29, for an incomplete list], to the best of our knowledge, the inner workings of IRW-SVA are not precisely documented anywhere — one would need to look at the source code of the package to know the exact steps of the algorithm. Given that IRW-SVA (the same is true for two-step SVA) is purely an algorithm that is not defined based on a statistical model or loss function, the steps of the algorithm are the heart of the algorithm. Therefore, in Section 3.2.2, I will fill this gap in the literature by documenting the IRW-SVA algorithm. In Section 4.1, I will attempt to reproduce and build upon the simulation study in Leek and Storey [4], which, among other things, will give readers a better idea of how SVA is used in practice.

3.2.1 Prerequisite: local false discovery rate (lfdr)

Before discussing the SVA algorithm, I will give a brief review of local false discovery rate (lfdr) [30], an empirical Bayes idea that is an integral part of the SVA algorithm.

Consider a simple example of a two-group model. Let the random variable $X \in \{0, 1\}$ denote the gender of a person and let the random variable Y denote the height of the person. The first layer of the model is

$$\begin{cases} P(X = 0) = \rho, \\ P(X = 1) = 1 - \rho. \end{cases} \tag{3.2.1}$$

The second layer of the model is

$$\begin{cases} \Big[Y \mid X = 0\Big] \sim f_0\left(y\right), \\ \Big[Y \mid X = 1\Big] \sim f_1\left(y\right). \end{cases} \tag{3.2.2}$$

Then, the marginal distribution of Y is

$$f\left(y\right) = \rho f_0\left(y\right) + \left(1 - \rho\right) f_1\left(y\right), \tag{3.2.3}$$

a mixture distribution.

Therefore, by Bayes' rule, we have

$$P\left(X = 1 \mid y\right) = \frac{P\left(X = 1\right) f\left(y \mid X = 1\right)}{f\left(y\right)} \tag{3.2.4}$$

$$= \frac{P\left(X = 1\right) f\left(y \mid X = 1\right)}{\rho f_0\left(y\right) + \left(1 - \rho\right) f_1\left(y\right)}. \qquad \text{plugging in (3.2.3)} \tag{3.2.5}$$

Now, in large-scale hypothesis testing, where we wish to conduct hundreds, thousands, or even more hypothesis tests simultaneously, assume that each hypothesis test satisfies the following independently:

$$\begin{cases} P\left(H_0 \text{ is true}\right) = \pi_0, \\ P\left(H_0 \text{ is false}\right) = 1 - \pi_0, \end{cases} \tag{3.2.6}$$

and

$$\begin{cases} \Big[Z \mid H_0 \text{ is true}\Big] \sim f_0\left(z\right), \\ \Big[Z \mid H_0 \text{ is false}\Big] \sim f_1\left(z\right), \end{cases} \tag{3.2.7}$$

where Z denotes the test statistic of the hypothesis test. $f_1\left(z\right)$ can be thought of as a mixture distribution. For example, given that $H_0 : \mu = 0$ is false, suppose $\mu = 1$ with

probability 0.3, in which case the test statistic follows density g_1, and $\mu = 2$ with probability 0.7, in which case the test statistic follows density g_2. Then f_1 would be $0.3 \times g_1 + 0.7 \times g_2$.

From (3.2.6) and (3.2.7), we know that the marginal distribution of Z is

$$f(z) = \pi_0 f_0(z) + (1 - \pi_0) f_1(z), \tag{3.2.8}$$

a mixture distribution.

The local false discovery rate (lfdr) is defined as

$$\begin{aligned} lfdr(z) &\coloneqq P(H_0 \text{ is true} \mid z) && && (3.2.9)\\ &= \frac{P(H_0 \text{ is true}) f(z \mid H_0 \text{ is true})}{f(z)} && \text{Bayes' rule} && (3.2.10)\\ &= \frac{\pi_0 f_0(z)}{f(z)}, && \text{definition of } \pi_0 \text{ and } f_0(z) && (3.2.11) \end{aligned}$$

which we estimate as

$$\widehat{lfdr}(z) = \frac{\widehat{\pi}_0 f_0(z)}{\widehat{f}(z)}, \tag{3.2.12}$$

where $\widehat{\pi}_0$ may be obtained as described in Storey and Tibshirani [31] or simply set equal to 1 for a conservative lfdr estimate, and $\widehat{f}(z)$ may be obtained by fitting a smooth curve to the histogram of the z values from all the hypothesis tests.

3.2.2 SVA algorithm

Recall the notations from Section 3.1.1 (see Table 3.1 for a summary). The goal of SVA is to infer X_2 so that the inferred X_2 ("surrogate variables") can be used in the downstream DE analysis (Section 3.1.1). In a nutshell, SVA iteratively reweights the columns of Y and then performs PCA on the reweighted Y, and the weights are based on the estimated probabilities

that the genes' expression levels are associated with the variables of interest and the hidden covariates, respectively: the more likely that a gene's expression level is associated with *any* of the *variables of interest*, the *less* weight the gene gets; on the other hand, the more likely that a gene's expression level is associated with *any* of the *hidden covariates*, the *more* weight the gene gets. I describe the details of the SVA algorithm in Algorithm 1 and Algorithm 2, where I refer to basic R functions such as `cbind()`, `dnorm()`, and `qnorm()`.

3.2.3 Choice of K

The SVA package provides two ways of automatically choosing K (the number of hidden covariates to infer): BE, a permutation-based approach that can be traced back to Buja and Eyuboglu [32], and Leek, an approach based on Leek [33]. The default approach in the SVA package is BE, which I summarize in Algorithm 3.

3.2.4 Thoughts on the SVA algorithm

I think there are three aspects of the SVA algorithm (Algorithm 1 and Algorithm 2) that are currently not justified in the SVA papers or documentation and need further justification.

First, by definition (Section 3.2.1), the local false discovery rates (Lines 11 and 15 of Algorithm 1) should be calculated based on the F-statistics from the partial F-tests. Alternatively, the p-values from the partial F-tests may be treated as test statistics whose null distributions are $\text{Unif}\,(0, 1)$. The probit transformation followed by a density calculation of the standard normal distribution (Line 5 of Algorithm 2) lacks theoretical justification.

Second, in Line 9 of Algorithm 1, the reduced model should be set as an intercept term plus the first K PCs so that in Line 10, the null hypothesis becomes that the variables of interest *and* the known covariates all have coefficient $= 0$ given the first K PCs — not only should we give a gene less weight in PCA if its expression level is likely to be associated with a variable of interest, we should also give it less weight if its expression level is likely

Algorithm 1: iteratively reweighted surrogate variable analysis (IRW-SVA)

Input:

- Y, $n \times p$ gene expression matrix that is sample by gene.
- $mod = \texttt{cbind}\,(1, X_0, X_1)$, design matrix that contains both the variables of interest and the known covariates. X_1 may be `NULL`, but X_0 can not be `NULL`.
- $mod0 = \texttt{cbind}\,(1, X_1)$, design matrix that contains only the known covariates. If X_1 is `NULL`, then $mod0$ is just a column of ones.
- K, number of hidden covariates to infer, i.e., number of surrogate variables to obtain.
- B, number of iterations; default is 5.

Output: inferred X_2 ($n \times K$ matrix of surrogate variables).

$R \leftarrow Y$; // Initialize the residual matrix as Y.
for $j \leftarrow 1$ **to** p **do**
  Regress the jth column of Y against mod;
  Replace the jth column of R with the residuals from the linear regression above;
  // Note that residuals from a linear regression always have zero mean.
end
Get the initial PCs by performing PCA on R without scaling the columns;
for $b \leftarrow 1$ **to** B **do**
  Set the full model to be mod plus the first K PCs (normalized); // Whether the PCs are normalized or not does not affect the result.
  Set the reduced model to be $mod0$ plus the first K PCs (normalized); // The difference between the full and reduced models is the variables of interest.
  For each gene (i.e., for each column of Y), get the p-value for H_0: the variables of interest all have coefficient $= 0$ given the reduced model, via a partial F-test for linear models;
  Convert the p-values to lfdr's (Alg. 2). Denote these lfdr's as $lfdr1$; // Each lfdr represents $P(\text{the variables of interest all have coefficient} = 0 \mid \text{F-statistic})$.
  Set the full model to be $mod0$ plus the first K PCs (normalized);
  Set the reduced model to be $mod0$; // The difference is the PCs.
  For each gene (i.e., for each column of Y), get the p-value for H_0: the PCs all have coefficient $= 0$ given the reduced model, via a partial F-test for linear models;
  Convert the p-values to lfdr's (Alg. 2). Denote these lfdr's as $lfdr2$; // Each lfdr represents $P(\text{the PCs all have coefficient} = 0 \mid \text{F-statistic})$.
  Weight the columns of Y by $lfdr1\,(1 - lfdr2)$;
  Perform PCA on the weighted Y after centering each column (without scaling the columns);
end
return the first K PCs (normalized) from the last PCA performed;

Algorithm 2: lfdr calculation in IRW-SVA (main steps only)

Input: $pVals$, p-values, vector of length p.

Output: $lfdrs$, local false discovery rates, vector of length p.

$\lambda \leftarrow 0.8$;

$\widehat{\pi}_0 \leftarrow \texttt{sum}\left(pVals > \lambda\right)/p\left(1-\lambda\right)$; `// Estimate the prior proportion of true nulls following Storey and Tibshirani [31].`

$\epsilon \leftarrow 10^{-8}$;

Floor each element of $pVals$ at ϵ and cap each element of $pVals$ at $1-\epsilon$;

$\widehat{f}_0 \leftarrow \texttt{dnorm}\left(\texttt{qnorm}\left(pVals\right)\right)$; `// qnorm() returns the quantile of the standard normal distribution (equivalent to probit). dnorm() returns the density.`

$\widehat{f}$ is obtained by fitting a smooth curve to the histogram of $pVals$ via kernel density estimation (`density()`) followed by cubic smoothing spline (`smooth.spline()`);

$lfdrs \leftarrow \widehat{\pi}_0 \widehat{f}_0 / \widehat{f}$; `// Compare to Equation (3.2.12).`

return $lfdrs$;

Algorithm 3: the BE algorithm for choosing K in SVA

Input:

- Y, $n \times p$ gene expression matrix that is sample by gene.
- $mod = \texttt{cbind}\left(1, X_0, X_1\right)$, design matrix that contains both the variables of interest and the known covariates.

Output: K, number of hidden covariates to infer.

Residualize Y against mod to obtain R; `// See Lines 1 to 5 of Algorithm 1.`

Perform PCA on R without scaling the columns. Denote the proportion of variance explained (PVE) by the kth PC by PVE_k, $k = 1, \cdots, \min\left(n, p\right)$;

$B \leftarrow 20$;

for $b \leftarrow 1$ **to** B **do**

  Permute each column of R to obtain R_b;

  Residualize R_b against mod to obtain R_b'; `// See Lines 1 to 5 of Algorithm 1.`

  Perform PCA on R_b' without scaling the columns;

end

The p-value for the kth PC is calculated as the proportion of permutations where the PVE of the kth PC is greater than or equal to PVE_k;

Enforce that the p-values increase (i.e., are non-decreasing) as k increases;

$\alpha \leftarrow 0.1$;

return the number of PCs with a p-value smaller than or equal to α;

to be associated with a known covariate. Similarly, in Lines 12 and 13 of Algorithm 1, the full model should be set as *mod* plus the first K PCs, and the reduced model should be set as *mod*, so that we control for the effect of both the variables of interest and the known covariates when determining whether a gene's expression level is associated with at least one of the PCs — instead of only controlling for the effect of the known covariates.

Third, in general, it is good practice to scale each variable (i.e., feature) to have unit variance before performing PCA (Section 2.2). Therefore, it is probably a good idea to scale the variables in Line 6 of Algorithm 1 (for the initial run of PCA) and immediately before Line 16 of Algorithm 1 (before weighting the variables based on the estimated probabilities), as well as in Lines 2 and 7 of Algorithm 3 (when choosing K).

3.3 Probabilistic estimation of expression residuals (PEER)

Probabilistic estimation of expression residuals (PEER) [5, 6] is arguably the most commonly used method for inferring hidden covariates in eQTL analysis. It is based on a Bayesian probabilistic model, which I summarize in Section 3.3.1.

3.3.1 PEER model

The PEER model is a Bayesian version of the factor analysis model (Equations 2.1.1 to 2.1.3) that explicitly models the known covariates. Recall the notations from Section 3.1.2 (see Table 3.1 for a summary). The overall equation of the PEER model is

$$\underset{n\times p}{Y} = \underset{n\times(K_1+K)}{X}\ \underset{(K_1+K)\times p}{W} + \underset{n\times p}{E}, \tag{3.3.1}$$

where Y is the $n \times p$ gene expression matrix that is sample by gene, X is the column concatenation of X_1 and X_2 — the known covariates and the hidden covariates, W is the weight matrix, and E is the error matrix.

The model assumptions are

$$x_{ik} \overset{\text{iid}}{\sim} \mathcal{N}(0,1),\ i = 1, \cdots, n;\ k = K_1 + 1, \cdots, K_1 + K, \tag{3.3.2}$$

$$w_{kj} \overset{\text{ind.}}{\sim} \mathcal{N}\left(0, \frac{1}{\beta_k}\right),\ k = 1, \cdots, K_1 + K;\ j = 1, \cdots, p, \tag{3.3.3}$$

where the covariate-specific weight precision β_k satisfies

$$\beta_k \overset{\text{iid}}{\sim} \Gamma(a_1, b_1),\ k = 1, \cdots, K_1 + K, \tag{3.3.4}$$

and

$$e_{ij} \overset{\text{ind.}}{\sim} \mathcal{N}\left(0, \frac{1}{\tau_j}\right),\ i = 1, \cdots, n;\ j = 1, \cdots, p, \tag{3.3.5}$$

where the gene-specific error precision τ_j satisfies

$$\tau_j \overset{\text{iid}}{\sim} \Gamma(a_2, b_2),\ j = 1, \cdots, p. \tag{3.3.6}$$

In the PEER model, a_1, b_1, a_2, b_2 are the hyperparameters. Once they are specified (the default is $a_1 = 0.001$, $b_1 = 0.1$, $a_2 = 0.1$, and $b_2 = 10$), inference is performed using variational Bayes and the posterior means are reported. In particular, the posterior means of X_2 are reported as the PEER factors and the posterior means of E are reported as the residuals [5].

After the PEER factors and the residuals are obtained, Stegle et al. [6] recommends either including the PEER factors as covariates or using the residuals as the response variables in the downstream analysis. The performance of these two approaches will be compared in a future study.

3.4 Hidden covariates with prior (HCP)

Hidden covariates with prior (HCP) [7] is another popular method for inferring hidden covariates in eQTL analysis. It is defined by minimizing a loss function, which I summarize in Section 3.4.1. Neither Mostafavi et al. [7] nor the HCP R package documents the HCP method well. For example, the squares in the loss function (3.4.1) are missing in both Mostafavi et al. [7] and the R package documentation, but one can deduce that the squares are there by inspecting the coordinate descent steps in the source code of the R package. This work aims to provide a better, more accurate documentation of the HCP method.

3.4.1 HCP loss function

Recall the notations from Section 3.1.2 (see Table 3.1 for a summary). Given Y, the $n \times p$ gene expression matrix that is sample by gene, and X_1, the $n \times K_1$ known covariate matrix, the HCP method looks for

$$\underset{X_2,\, W_1,\, W_2}{\arg\min} \left|\left|\left| \underset{n\times p}{Y} - \underset{n\times K}{X_2}\ \underset{K\times p}{W_2} \right|\right|\right|_2^2 + \lambda_1 \left|\left|\left| \underset{n\times K}{X_2} - \underset{n\times K_1}{X_1}\ \underset{K_1\times K}{W_1} \right|\right|\right|_2^2 + \lambda_2 |||W_1|||_2^2 + \lambda_3 |||W_2|||_2^2, \qquad (3.4.1)$$

where X_2 is the hidden covariate matrix, W_1 and W_2 are weight matrices of the appropriate dimensions, and $\lambda_1, \lambda_2, \lambda_3 > 0$ are the tuning parameters. The name of the method, "hidden covariates with prior", comes from the second term in (3.4.1), where we inform the hidden covariates with the known covariates. The optimization is done through coordinate descent with one deterministic initialization. The obtained X_2 is reported as the inferred hidden covariates ("HCPs").

We have seen that both SVA and PEER are closely related to PCA. The HCP method is closely related to PCA as well — the first term in (3.4.1) is very similar to (2.2.21), the only difference being the rows of W_2 in (3.4.1) are not required to be orthonormal and X_2 is

not required to be the projection of Y onto the subspace spanned by the rows of W_2.

CHAPTER 4

Results

4.1 Simulation study shows that SVA can capture non-global hidden covariates in DE analysis

In this section, I attempt to reproduce and build upon the simulation study in Leek and Storey [4]. My analysis illustrates how SVA is used in practice and demonstrates that SVA can be useful in capturing non-global hidden covariates in certain simulated DE data sets.

4.1.1 Data simulation

Leek and Storey [4] designed 16 simulation scenarios [4, Table S1 and Table S2]. For simplicity, I focus on the first eight simulation scenarios (i.e., experiments, with 10 replicates each) and omit the last eight, which are extensions of the first eight simulation scenarios. A summary of my experiments is given in Table 4.1.

Using notations consistent with those in Table 3.1, for each experiment and each replicate, I simulate a data set under

$$\underset{n\times p}{Y} = \underset{n\times K_0}{X_0}\ \underset{K_0\times p}{W_0} + \underset{n\times K}{X_2}\ \underset{K\times p}{W_2} + \underset{n\times p}{E}\,, \tag{4.1.1}$$

where Y is the gene expression matrix that is sample by gene, X_0 is the variables of interest, W_0 is the corresponding effect size matrix, X_2 is the hidden covariates, W_2 is the corresponding effect size matrix, and E is the noise matrix. The data dimensions are $n = 20$,

	Type of X_2	**Correlation**	**Overlap**	$(X_2)_{i1},\ i = 1, \cdots, 10$	$(X_2)_{i1},\ i = 11, \cdots, 20$	**Nonzero entries of W_2**
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	discrete	low	low	$Ber\,(0.5)$	$Ber\,(0.5)$	201 - 700
2			high			101 - 600
3		high	low	$Ber\,(0.7)$	$Ber\,(0.2)$	201 - 700
4			high			101 - 600
5	continuous	low	low	$\mathcal{N}\,(0,1)$	$\mathcal{N}\,(0,1)$	201 - 700
6			high			101 - 600
7		high	low	$\mathcal{N}\,(0,1)$	$\mathcal{N}\,(1,1)$	201 - 700
8			high			101 - 600

Table 4.1: Summary of eight experiments. The first column is the index of the experiment. The third column refers to the correlation between the variable of interest and the hidden covariate. The fourth column refers to the association overlap. Some cells are empty because they repeat the cells directly above them. Each experiment is characterized by Columns (2) to (4). Columns (2) and (3) determine Columns (5) and (6). Column (4) determines Column (7).

$p = 1000$, $K_0 = 1$, $K = 1$. That is, there are 20 individuals, 1000 genes, one variable of interest, and one hidden covariate.

Specifically, for each data set, X_0 is a column vector where the first 10 entries are one and the last 10 entries are zero. W_0 has its first 300 entries drawn independently from $\mathcal{N}\,(0, 2.5)$ and its other entries equal to zero. X_2 is simulated based on Columns (5) and (6) of Table 4.1, which in turn are determined by Columns (2) and (3). The nonzero entries of W_2 are drawn independently from $\mathcal{N}\,(0, 2.5)$; which entries of W_2 are nonzero is determined by Column (7) of Table 4.1, which in turn is determined by Column (4). Lastly, the noise matrix is simulated using

$$e_{ij} \overset{\text{ind.}}{\sim} \mathcal{N}\left(0, \sigma_j^2\right),\ i = 1, \cdots, n;\ j = 1, \cdots, p, \tag{4.1.2}$$

where the gene-specific noise standard deviation σ_j is drawn from

$$\sigma_j \overset{\text{iid}}{\sim} InvGamma\left(10, 9\right), \ j = 1, \cdots, p. \tag{4.1.3}$$

In total, $8 \times 10 = 80$ data sets are simulated.

4.1.2 Methods

Recall differential gene expression (DE) analysis (Section 3.1.1). For each gene, we would like to decide whether its expression level is associated with the variable of interest, controlling for the effect of the hidden covariate. I compare the performance of four approaches on the 80 simulated data sets: Ideal, Unadjusted, PCA, and SVA.

For Ideal, we assume that X_2 is known. Therefore, for each gene, I conduct a multiple linear regression with the gene expression vector as the response variable and X_0 and X_2 as the predictors, and I store the p-value corresponding to the null hypothesis that the coefficient corresponding to X_0 is zero given X_2.

For Unadjusted, we assume that the true model does not contain X_2. Therefore, for each gene, I conduct a simple linear regression with the gene expression vector as the response variable and X_0 as the predictor.

For PCA, first, I residualize Y against X_0 (see Lines 1 to 5 of Algorithm 1). Second, I perform PCA on the residualized gene expression matrix with centering and scaling. Third, I choose the number of PCs using my own implementation of the BE algorithm, which is analogous to Algorithm 3, with $\alpha = 0.05$. Here, BE always chooses the true K ($K = 1$). Lastly, for each gene, I conduct a multiple linear regression with the gene expression vector as the response variable and X_0 and the first PC as the predictors.

For SVA, first, I apply the SVA package to choose the number of SVs via BE (Algorithm 3). Here, BE almost always chooses the true K ($K = 1$), except it chooses $K = 2$ in one replicate of the fourth experiment. Then, I run the main SVA algorithm with the chosen K (Algorithm 1). Lastly, for each gene, I conduct a multiple linear regression with the gene

expression vector as the response variable and X_0 and the obtained SVs as predictors.

4.1.3 Performance comparison

In this section, I evaluate the performance of SVA against Ideal, Unadjusted, and PCA (when applicable) in three ways: speed, adjusted R^2 in capturing the true hidden covariate, and area under the precision-recall curve (AUPRC). I believe that these measures are more practical and direct than those used in Leek and Storey [4].

My analysis shows that SVA is fast and can capture non-global hidden covariates reasonably well in the simulated data sets. In terms of computational efficiency, SVA is comparable to PCA; both can run within a few seconds (Figure 4.1). In terms of adjusted R^2 and AUPRC, SVA improves upon Unadjusted and PCA; this is true in some of the experiments more than others (Figure 4.2 and Figure 4.3).

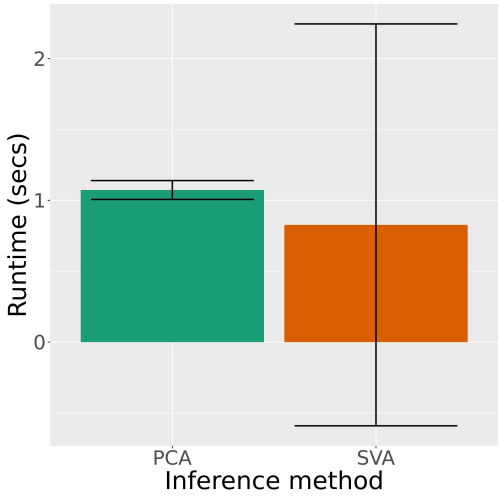

Figure 4.1: Comparison in terms of computational efficiency. The height of the green bar is the average runtime of PCA (including residualization; Section 4.1.2) across the 80 simulated data sets. The height of the orange bar is the average runtime of SVA (including BE). The error bars represent standard deviations.

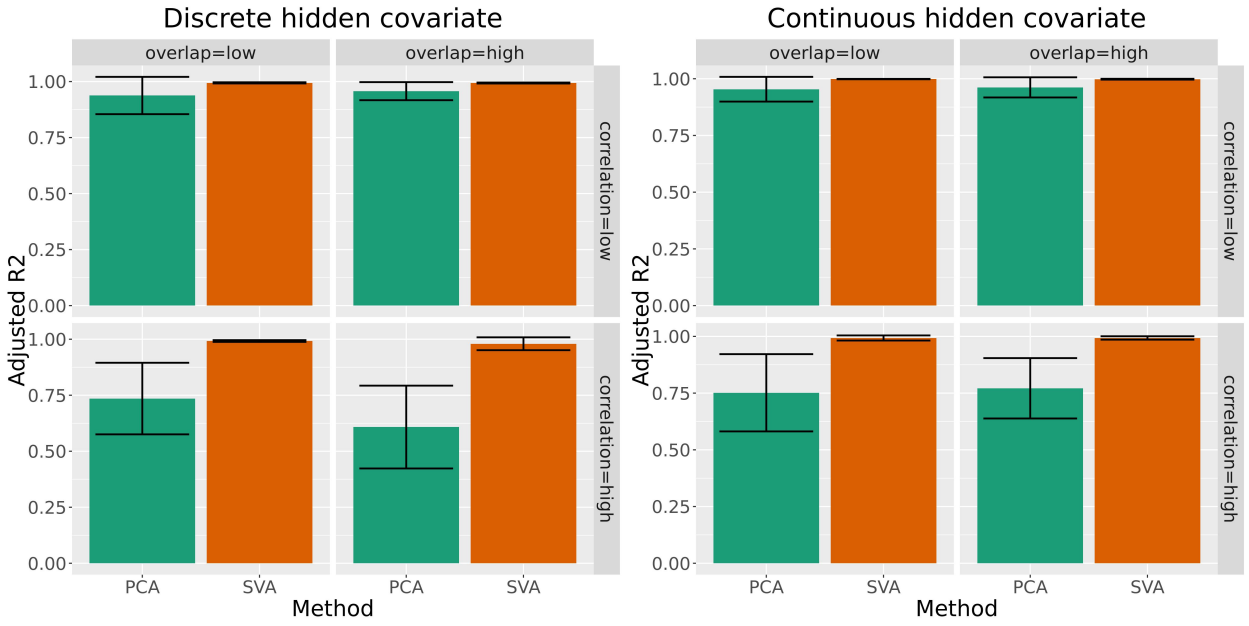

Figure 4.2: Comparison in terms of adjusted R^2 in capturing the true hidden covariate. The eight subplots correpsond to the eight experiments. The height of each orange bar is the average adjusted R^2 of the SV(s) in capturing the true hidden covariate across 10 replicates. The error bars represent standard deviations.

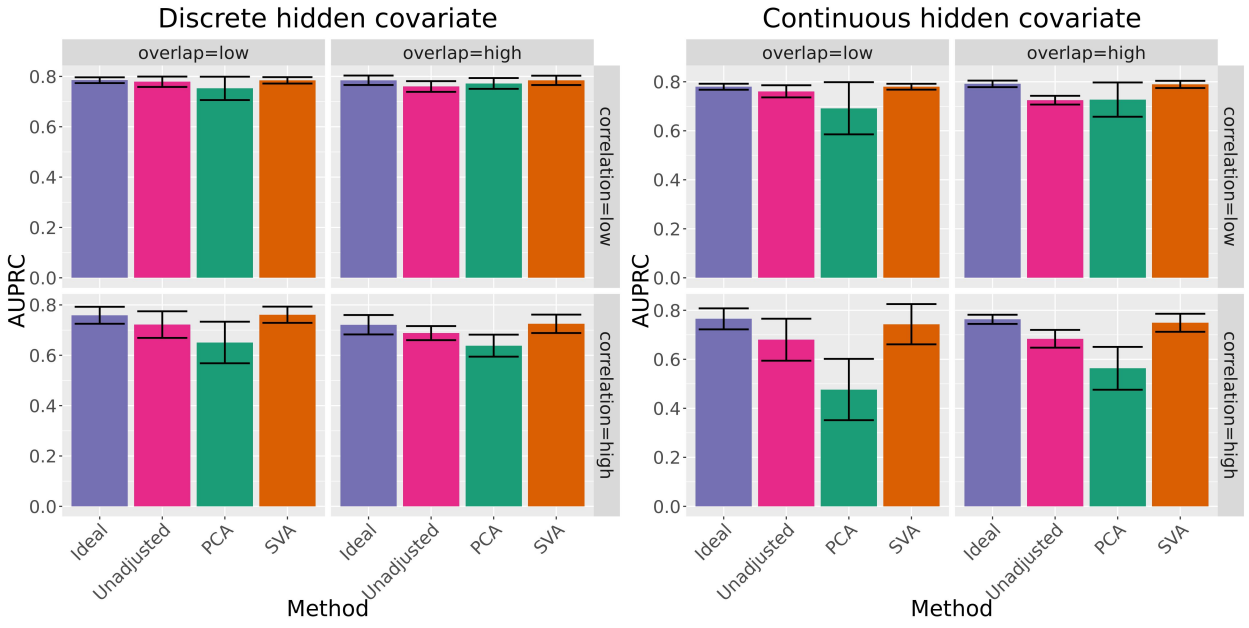

Figure 4.3: Comparison in terms of AUPRC. The eight subplots correpsond to the eight experiments. The height of each orange bar is the average AUPRC using the SVA approach (Section 4.1.2) across 10 replicates. The error bars represent standard deviations.

CHAPTER 5

Conclusion

In Chapter 2, I provide a summary and review of three classical statistical methods: factor analysis, principal component analysis (PCA), and probabilistic PCA (PPCA), all of which fall under the category of linear factor models. I show that although factor analysis is based on a probabilistic model and PCA is traditionally derived by optimizing some objective functions (either maximum variance or minimum reconstruction error), PCA can also be derived as a limit of the PPCA model, which in turn is a special case of the factor analysis model. This chapter can be a valuable resource for students in statistics and other disciplines who need to learn about factor analysis, PCA, and PPCA.

In Chapter 3, I provide an overview of differential gene expression (DE) analysis and expression quantitative trait locus (eQTL) analysis from a statistical perspective, with an emphasis on DE and eQTL analysis with hidden covariates. For the first time in the scientific literature, I accurately document surrogate variable analysis (SVA), probabilistic estimation of expression residuals (PEER), and hidden covariates with prior (HCP) — the most popular methods for inferring hidden covariates in DE and eQTL analysis today — and delineate their connections to factor analysis and PCA. This chapter can be a valuable resource for computational biologists who need a better understanding of the methodology behind SVA, PEER, and HCP, as well as those who need an introduction to DE and eQTL analysis from a statistical perspective.

Though not the emphasis of this work, in Chapter 4, I perform a simulation study following the design in Leek and Storey [4] and show that SVA can capture non-global

hidden covariates reasonably well in certain simulated DE data sets. This chapter may be referenced as an example of clear and precise documentation of simulation studies.

Altogether, this work can be a useful reference manual for students and researchers working with linear factor models or newly developed methods for capturing hidden covariates in DE or eQTL analysis.

Bibliography

[1] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, 1995.

[2] Stephan C. Schuster. Next-generation sequencing transforms today's biology. *Nature methods*, 5(1):16–18, 2008.

[3] Jeffrey T. Leek and John D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):e161, 2007.

[4] Jefferey T. Leek and John D. Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, 105(48):18718–18723, 2008.

[5] Oliver Stegle, Leopold Parts, Richard Durbin, and John Winn. A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies. *PLoS Computational Biology*, 6(5):e1000770, 2010.

[6] Oliver Stegle, Leopold Parts, Matias Piipari, John Winn, and Richard Durbin. Using probabilistic estimation of expression residuals (PEER) to obtain increased power and interpretability of gene expression analyses. *Nature Protocols*, 7(3):500–507, 2012.

[7] Sara Mostafavi, Alexis Battle, Xiaowei Zhu, Alexander E. Urban, Douglas Levinson, Stephen B. Montgomery, and Daphne Koller. Normalizing RNA-sequencing data by modeling hidden covariates with prior knowledge. *PLoS ONE*, 8(7):e68141, 2013.

[8] GTEx Consortium. The GTEx Consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509):1318–1330, 2020.

[9] Rebecca L. Walker, Gokul Ramaswami, Christopher Hartl, Nicholas Mancuso, Michael J. Gandal, Luis de la Torre-Ubieta, Bogdan Pasaniuc, Jason L. Stein, and

Daniel H. Geschwind. Genetic control of expression and splicing in developing human brain informs disease mechanisms. *Cell*, 179(3):750–771, 2019.

[10] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, London, 1979.

[11] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Pearson Prentice Hall, Upper Saddle River, NJ, sixth edition, 2007.

[12] K. G. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32(4):443–482, 1967.

[13] D. N. Lawley and A. E. Maxwell. *Factor Analysis as a Statistical Method*. Butterworths, London, second edition, 1971.

[14] Donald B. Rubin and Dorothy T. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.

[15] Raymond B. Cattell. *The Scientific Use of Factor Analysis in Behavioral and Life Sciences*. Plenum Press, New York, 1978.

[16] Michael Richman. Rotation of principal components. *Journal of Climatology*, 6:293–355, 1986.

[17] Henry F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

[18] Godfrey Hilton Thomson. *The Factorial Analysis of Human Ability*. Houghton Mifflin, Boston, fifth edition, 1951.

[19] M. S. Bartlett. The statistical conception of mental factors. *British Journal of Psychology*, 28(1):97–104, 1937.

[20] D. N. Lawley. VI.—The estimation of factor loadings by the method of maximum likelihood. *Proceedings of the Royal Society of Edinburgh*, 60(1):64–82, 1940.

[21] Christopher Chatfield and Alexander J. Collins. *Introduction to Multivariate Analysis*. Springer US, Boston, MA, 1980.

[22] Ian T. Jolliffe. *Principal Component Analysis*. Springer, New York, second edition, 2002.

[23] Otto Bretscher. *Linear Algebra With Applications*. Pearson Prentice Hall, Upper Saddle River, NJ, fourth edition, 2009.

[24] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065), 2016.

[25] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[26] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572, 1901.

[27] Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61(3):611–622, 1999.

[28] Yixin Wang and David M. Blei. The Blessings of Multiple Causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019.

[29] Jeffrey T. Leek, W. Evan Johnson, Hilary S. Parker, Andrew E. Jaffe, and John D. Storey. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics*, 28(6):882–883, 2012.

[30] Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference*, volume 5. Cambridge University Press, 2016.

[31] J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.

[32] Andreas Buja and Nermin Eyuboglu. Remarks on parallel analysis. *Multivariate Behavioral Research*, 27(4):509–540, 1992.

[33] Jeffrey T. Leek. Asymptotic conditional singular value decomposition for high-dimensional genomic data. *Biometrics*, 67(2):344–352, 2011.