SEMI-PARAMETRIC PRINCIPAL COMPONENT ANALYSIS FOR POISSON COUNT DATA WITH APPLICATION TO MICROBIOME DATA ANALYSIS

by

Tianshu Huang

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
August 2017

Table of Contents

List of Tables

List of Figures

Abstract

Principal Component Analysis (PCA) is a widely used tool for dimensional reduction and data visualization. However, it cannot be used directly for microbiome data. In this thesis, we aim to develop PCA for the underlying abundance of OTUs under the assumption that conditional on the latent OTU abundance, the observed counts follow independent Poisson distributions. By correcting this Poisson measurement error, we base our PCA on an unbiased estimator of the covariance matrix of the latent OTU abundances. We further correct the sequencing depth noise by analyzing the data as compositional. In order to deal with the non-normality, we propose a logarithm-transformed Poisson-corrected PCA. We then incorporate sequencing depth correction into this method. Finally, we address the problem of projecting the observed data onto the log-transformed principal component space. We examine the performance of our methods on simulated data and tongue microbiomes data.

List of Abbreviations and Symbols Used

ANCOM	Analysis of composition of microbiomes
OTU	Operational Taxonomic Units
PCA	Principal component analysis
rMLE	restricted Maximum Likelihood Estimation
SVD	singular value decomposition

Acknowledgements

I would like to express the deepest appreciation to all those who gave me the possibility to complete this thesis.

A special thanks to my thesis supervisor, Professor Hong Gu, and co-supervisor, Professor Toby Kenney for their persistent support. They guided me through the thesis research, contributed in countless ways to my development and encouraged me all the time. It was an amazing and challenging learning experience working with them.

I would like to thank my thesis readers Professor Chris Field and Professor Edward Susko for spending time reading my thesis and providing valuable comments on this thesis.

I am grateful to my parents and friends for their support throughout my program. I could not have completed this program without their wise counsel and encouragement.

Chapter 1

Introduction

1.1 Principal Component Analysis as a Dimension Reduction and Data Exploration Tool

Principal component analysis (PCA) is a statistical procedure that reduces the dimensionality of the data while retaining most of the variation in the data set [1]. PCA identifies a number of orthogonal directions for the data to be projected on, called principal components. The data projected on the first principal component accounts for as much as possible of the variability in the data, and each succeeding principle component, subject to the orthogonality constraints, accounts for as much as possible of the remaining variability. By successfully reducing the dimensionality of the data, PCA can thus help to identify new potentially meaningful variables and to understand better the correlation structure among the original variables.

PCA, as an important data exploration method, is widely used to identify patterns in data, based on which a statistical model may be proposed for further analysis of the data.

In computation, PCA is equivalent to applying singular value decomposition (SVD) on the column centralized data matrix. There has been a lot of interest in applying SVD to gene expression data where the dimension of the data is higher than the number of observations [2] [3].

1.2 Measurement Error for PCA

In experiments, the data that we observe differ randomly from the values that we intend to measure and such error is called the measurement error. The measurement error problem has been considered in great many papers and several monographs [4]. Ignoring measurement errors in many cases can lead to biased inference or even erroneous conclusions [5].

PCA is widely used to reduce the dimension of the high dimensional data to the space spanned by a few principal components, under the assumption that the eigenvectors associated with smaller eigenvalues are predominantly decided by the noise or otherwise less important features of the data. However, classical PCA does not distinguish between variance caused by measurement error noise and the true underlying signal variations. Even when an estimate of the measurement error variance is available, this information is not used when calculating the principal component directions, e.g., by deweighting noisy data [6]. It is not difficult to see that when measurement error is additive and independent to the variables being measured, the principal component directions are going to be decided by the sum of the variances due to the variability of the variables being measured and the measurement errors. If the variance due to measurement error is close to a multiple of the identity matrix, then the measurement error will not change the estimated principal components much, so the error introduced by ignoring the measurement error will not usually be serious.

Some work has been done in correcting the independent additive measurement errors in PCA. For example, Maximum likelihood principal component analysis (MLPCA) is an analog to PCA that incorporates information of additive measurement errors to develop PCA models that are optimal in a maximum likelihood sense [7].

An alternative method, termed Weighted EMPCA, gives different weights to different observations with the weights decided by the estimated measurement errors, so that less weight is given to the observations with higher measurement errors [6].

More recently, Hellton and Thoresen [8] investigated the effects of random, additive errors on PCA and illustrated that the measurement error will contribute to a large variability in component loadings, relative to the loading values, such that interpretation based on the loadings can be difficult.

It is not clear how principal component analysis will be influenced when the measurement error is not additive and/or not independent of the underlying variables. We deal with one type of such measurement error in the microbiome data analysis for PCA in this thesis.

1.3 Microbiome Data

The concept of the human microbiome was first suggested by Joshua Lederberg, who proposed the term Microbiome, “to signify the ecological community of commensal, symbiotic, and pathogenic microorganisms that literally share our body space” [9]. Microbes are extremely small living organisms that include bacteria, viruses, and fungi. Large and diverse populations of bacteria, viruses, and fungi occupy almost every surface of the human body and the environment [10]. It is estimated that there are nearly 30 trillion bacterial cells living in or on each human [11], and these along with other microbes are collectively known as the microbiome.

The millions of organisms that make up the human microbiome play an important role in both health and disease. Humans and microbes depend on one another: our bodies provide microbes with resources, and the microbes provide functions necessary for our health [12].

Researchers can use DNA sequencing to identify microbes. One common technique is to sequence a marker: a short, unique DNA sequence that can be used to identify the genome that contains it. Using markers, researchers can identify a microbe without having to sequence its entire genome. This shortcut allows them to identify all the species present in a sample very quickly. The next generation sequencing technology accelerated the data collection on microbiome in and on human bodies under different health conditions and also microbiome from different environments, for example ocean and soil. These data provide potential for us to understand more about how the microbes as a community interact with both humans and the environment. However the development of data analysis techniques is lacking behind the data collection, although there has been a lot of work dedicated to develop new computer tools and technology to make data analysis more manageable [13].

Microbiome sequence data sets are typically high dimensional, with the number of taxa much greater than the number of samples. Also it is sparse as most taxa are only observed in a small number of samples. Typically the data are collected as the counts of different microbes observed in each sample. The sampling error for these count data adds an additional challenge for the downstream analysis [14].

Furthermore, because the total number of reads in a sample is influenced by a number of factors in the sample collection and sequencing, rather than the underlying microbial community, the counts for different microbes are not comparable across different samples. This is the issue referred to as the sequencing depth. Often sequencing depth is measured by the total number of microbes observed in each sample. Obviously the total count of microbes in each sample is related to both the noise integrated through the sequencing process and up stream data sorting and the abundance of all different microbes in the sample. In order to make the data entries comparable for the same OTU across different samples, typically microbiome data have been treated as compositional data and analyzed using the methods developed for the compositional data [15]. The problem with taking proportions in the data and treating them as compositional data is that the different sequencing depths across samples add another level of heterogeneity in addition to the heterogeneity due to the different sampling errors [16].

Some work has been done on analysis of compositional data. Analysis of composition of microbiomes (ANCOM) is a non-parametric statistical framework, which accounts for the underlying structure in the data and can be used for comparing the composition of microbiomes in two or more populations [17]. An additive logistic normal multinomial regression model to associate the covariates to bacterial composition is proposed. The model can naturally account for sampling variabilities and zero observations and also allow for a flexible covariance structure among the bacterial taxa [18].

In this thesis, we will use "sequencing depth" more generally to represent a random multiplier applied to the abundance of all OTUs, which represents the effect of a number of experimental factors in sample collection and sequencing. This multiplier is not identifiable, which is why the tradition in microbiome analysis is to treat it as the total read count. We will follow this tradition when performing PCA on the latent Λ in Chapter 2. However, when dealing with the transformed Λ in Chapter 4, we cannot have that the compositional form of Λ follows a log-normal distribution, or similar, so it does not make sense to let Λ be compositional in this case. In that chapter, we consider sequencing depth to be a random multiplier, and give two methods to identify a unique solution.

1.4 Structure of the Thesis

The major goal of this thesis is to develop PCA for count data assuming the sampling errors are Poisson, with additional effort made to correct for different sequencing depths across the samples.

The structure of the thesis is as follows. PCA analysis for count data with or without sequencing depth complication is developed in Chapter 2. In consideration of the exponential growth of bacteria, we introduce the non-linear logarithm transformation on the latent abundance and build a Poisson error correction PCA without considering the sequencing depth complication in Chapter 3. We then extend the log-transformed PCA to the case which treats sequencing depth as an nuisance random variable in Chapter 4.

The former three chapters focus on the calculation of the principal component directions. For the log-transformed problem, the projection of the latent unobserved data to the calculated principal component space is not a trivial problem. We devote Chapter 5 to the projection of the latent data to the principal component space. Simulation results for different versions of the PCA are included in the corresponding chapters. The real data analysis are included in Chapter 6. Chapter 7 concludes the thesis.

Chapter 2

Principal Component Analysis for Count Data with Poisson Sampling Errors

Suppose we are interested in the analysis of some latent multidimensional variable $\Lambda = (\Lambda_1, \Lambda_2, \cdots, \Lambda_p)$, where Λ_j $(j = 1, \ldots, p)$ is a random variable representing the true abundance of the jth OTU for microbiome data. We collect n samples from this latent variable, which can be arranged in a matrix $\Lambda = (\Lambda_{ij})_{i=1,\ldots,n; j=1,\ldots,p}$. However we only observe the matrix $X = (X_{ij})_{i=1,\ldots,n; j=1,\ldots,p}$, where X_{ij} are conditionally independent Poisson random variables with mean Λ_{ij}. Sometimes we will need to refer to the underlying distribution of the random vector, Λ, of Poisson means, and use $\mathbb{X}$ to denote the corresponding random vector of Poisson counts.

Applying PCA directly on X gives a biased estimate for the principal components of Λ, since $\mathbb{X}_j$ has larger variance in cases where Λ_j is larger. (Or if we use correlation to determine PCA, then $\mathbb{X}_j$ has relatively smaller variance when Λ_j is larger.) Our objective is to devise a method to correct this bias. We develop this method in Section 2.1, followed by the simulation results for this case in Section 2.2.

There is an additional complication when we are studying microbiome data, namely sequencing depth. For microbiome data, the sequencing depth is mostly determined by experimental factors, rather than by any biologically interesting aspects of the samples. Therefore, we are more interested in the compositional form. More formally, we suppose the latent composition of the system is given by some random vector Λ_c, with $\Lambda_c\mathbf{1} = \mathbf{1}$, and that the vector or Poisson mean $\Lambda = s\Lambda_c$ for some (scalar) random variable s, which is to be treated as a noise variable. In Section 2.3, we develop a method to estimate the principal components of this Λ_c vector under this model, followed by the simulation results for this case in Section 2.4.

2.1 Poisson Error Correction for PCA

By the law of total variance, we know that for two random variables X and Y, $\mathrm{Var}(X) = \mathbb{E}[\mathrm{Var}[X|Y]] + \mathrm{Var}[\mathbb{E}[X|Y]]$. Now we are interested in two random vectors Λ and $\mathbb{X}$. $\Lambda = \begin{pmatrix}\Lambda_1 & \Lambda_2 \cdots \Lambda_p\end{pmatrix}$, and each Λ_j $(j = 1, 2, \cdots, p)$ is a random variable. $\mathbb{X}|\Lambda$ follows a Poisson distribution with mean Λ and $\mathbb{X} = \begin{pmatrix}\mathbb{X}_1 & \mathbb{X}_2 \cdots \mathbb{X}_p\end{pmatrix}$, $\mathbb{X}_j$ $(j = 1, 2, \cdots, p)$ is a random variable independent with each other conditional on Λ_j.

The law of total variance trivially extends to random vectors. The Variance-Covariance matrix of $\mathbb{X}$ is

$$\begin{aligned}\mathrm{Var}(\mathbb{X}) &= \mathbb{E}[\mathbb{X}\mathbb{X}^T] - \mathbb{E}[\mathbb{X}](\mathbb{E}[\mathbb{X}])^T \\ &= \mathbb{E}[\mathbb{E}[\mathbb{X}\mathbb{X}^T|\Lambda]] - \mathbb{E}[\mathbb{E}[\mathbb{X}|\Lambda]](\mathbb{E}[\mathbb{E}[\mathbb{X}|\Lambda]])^T \\ &= \mathbb{E}\left[\mathrm{Var}[\mathbb{X}|\Lambda] + (\mathbb{E}[\mathbb{X}|\Lambda])(\mathbb{E}[\mathbb{X}|\Lambda])^T\right] - (\mathbb{E}[\mathbb{E}[\mathbb{X}|\Lambda]])(\mathbb{E}[\mathbb{E}[\mathbb{X}|\Lambda]])^T \\ &= \mathbb{E}[\mathrm{Var}[\mathbb{X}|\Lambda]] + \mathrm{Var}[\mathbb{E}[\mathbb{X}|\Lambda]] \end{aligned} \tag{2.1}$$

Since $\mathbb{E}[\mathbb{X}|\Lambda] = \Lambda$,

$$\mathrm{Var}[\mathbb{E}[\mathbb{X}|\Lambda]] = \mathrm{Var}[\Lambda] \tag{2.2}$$

Since $\mathbb{X}_j \sim \mathrm{Po}(\Lambda_j)$, and $\mathbb{X}_j|\Lambda$ is independent of $\mathbb{X}_i|\Lambda$, for all $i \neq j$ and $i, j = 1, 2, \cdots p$, $\mathrm{Var}(\mathbb{X}_i|\Lambda_i) = \Lambda_i$. The off-diagonal elements of $\mathbb{E}[\mathrm{Var}[\mathbb{X}|\Lambda]]$ are 0's, and its jth diagonal entry is Λ_j, so

$$\mathbb{E}[\mathrm{Var}[\mathbb{X}|\Lambda]] = \mathbb{E}[\mathrm{diag}(\Lambda_1, \Lambda_2, \cdots, \Lambda_p)] \tag{2.3}$$

By plugging in equations (2.2) and (2.3), the equation (2.1) is equivalent to

$$\mathrm{Var}[\mathbb{X}] = \mathrm{Var}[\Lambda] + \mathbb{E}[\mathrm{diag}(\Lambda_1, \Lambda_2, \cdots, \Lambda_p)]$$

The latent variance is given by:

$$\begin{aligned}\Sigma_\Lambda &= \mathrm{Var}[\Lambda] \\ &= \mathrm{Var}[\mathbb{X}] - \mathbb{E}[\mathrm{diag}(\Lambda_1, \Lambda_2, \cdots, \Lambda_p)]\end{aligned} \tag{2.4}$$

So an unbiased estimator for Var[Λ] is given by:

$$\begin{aligned}\widehat{\Sigma_\Lambda} &= \frac{1}{n-1}\tilde{X}\tilde{X}^T - \mathrm{diag}\left(\frac{X^T 1}{n}\right) \\ &= \widehat{\Sigma_X} - \mathrm{diag}\left(\overline{X}\right)\end{aligned} \tag{2.5}$$

where matrix X is a realization of random vector $\mathbb{X}$ and $\tilde{X}$ is the column centered matrix of X. We use $\widehat{\Sigma_X}$ to denote the sample Variance-Covariance matrix from data X, and $\overline{X}$ to denote the sample mean of X.

The Variance-Covariance matrix for classical PCA is computed by $\frac{1}{n-1}\tilde{X}\tilde{X}^T$, which is the first term in (2.5). So our method corrects the diagonal elements and the off-diagonal terms keep the same. We then implement eigen-decomposition on this Variance-Covariance matrix to obtain principal components.

2.2 Simulation for Principal Component Analysis of Σ_Λ

We test the performance of the Poisson noise corrected PCA on simulated data. We assume the Poisson mean matrix of dimension n by p, is given by $\Lambda = \mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T$, where $\mathbf{u}$,$\mathbf{v}$ and $\mathbf{w}$ are random vectors generated under normal distributions, say $\mathbf{u} = (u_1, u_2, \cdots, u_p)^T$, $\mathbf{v} = (v_1, v_2, \cdots, v_n)^T$ and $\mathbf{w} = (w_1, w_2, \cdots, w_p)^T$. Each u_i $(i = 1, 2, \cdots, p)$ has the same mean

μ_u and standard deviation σ_u, similarly each element of $\mathbf{v}$ follows a $\mathrm{N}(\mu_v, \sigma_v^2)$ distribution and each element of $\mathbf{w}$ follows a $\mathrm{N}(\mu_w, \sigma_w^2)$ distribution. $\mathbf{1}$ is a vector of length n of all 1's. We choose the means and standard deviations for u, v, and w to ensure all entries of Λ are positive with high probability.

Let $\mathbf{w}_0$ be a unit length vector in the direction of $\mathbf{w}$.

If we could directly observe Λ, its sample variance would be:

$$\begin{aligned}
\widehat{\Sigma_\Lambda} &= \frac{1}{n-1}\left(\Lambda - \frac{\mathbf{1}\mathbf{1}^T}{n}\Lambda\right)^T\left(\Lambda - \frac{\mathbf{1}\mathbf{1}^T}{n}\Lambda\right)\\
&= \frac{1}{n-1}\left((\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T) - \frac{\mathbf{1}\mathbf{1}^T}{n}(\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T)\right)^T\left((\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T) - \frac{\mathbf{1}\mathbf{1}^T}{n}(\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T)\right)\\
&= \frac{1}{n-1}\left((\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T) - (\mathbf{1}\mathbf{u}^T + \frac{\sum_i \mathbf{v}_i}{n}\mathbf{1}\mathbf{w}^T)\right)^T\left((\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T) - (\mathbf{1}\mathbf{u}^T + \frac{\sum_i \mathbf{v}_i}{n}\mathbf{1}\mathbf{w}^T)\right)\\
&= \frac{1}{n-1}\left((\mathbf{v} - \overline{\mathbf{v}}\mathbf{1})\mathbf{w}^T\right)^T\left((\mathbf{v} - \overline{\mathbf{v}}\mathbf{1})\mathbf{w}^T\right)\\
&= \frac{1}{n-1}\mathbf{w}(\mathbf{v} - \overline{\mathbf{v}}\mathbf{1})^T(\mathbf{v} - \overline{\mathbf{v}}\mathbf{1})\mathbf{w}^T\\
&= \widehat{\sigma_v^2}\mathbf{w}\mathbf{w}^T
\end{aligned}$$

So there is only one non-zero principal component which is parallel to $\mathbf{w}$, i.e. $\mathbf{w}_0$.

We generate the data matrix X from its Poisson mean Λ. By applying our method on X to estimate $\widehat{\Sigma_\Lambda}$, we can compare its PCA result with that from uncorrected PCA by using $\widehat{\Sigma_X}$ on X to measure how well our method corrects the Poisson error.

We use the absolute value of cosine of the angle between the estimated and true principal components to assess the accuracy of the estimated principal components. If the cosine of the angle between $\mathbf{w}_0$ and PC1 from Poisson error corrected PCA is significantly greater than the cosine of the angle between $\mathbf{w}_0$ and PC1 from classical PCA, we can conclude that our method corrects the Poisson error.

Data are simulated under 3 scenarios, all with $p = 10$, $\mathbf{u} \sim \mathrm{N}(10, 2^2)$, and $\mathbf{v} \sim \mathrm{N}(5, 0.1^2)$ with $\mathbf{w} \sim \mathrm{N}(30, 5^2)$, $\mathbf{w} \sim \mathrm{N}(30, 15^2)$, and $\mathbf{w} \sim \mathrm{N}(30, 25^2)$. We vary the standard deviation of these normal distributions in the $\mathbf{w}$ direction to cover different ranges. For each scenario, we set sample sizes n as 10, 50, 100, 300, 500, 1000 respectively, and the number of variables p is 10. We then simulate 100 Λ matrices by $\Lambda = \mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T$ for each sample size and for each Λ we generate one X matrix.

The simulation follows the procedure below:

1. First we generate $\Lambda = \mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T$
   (a) $\{u_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_u and standard deviation σ_u
   (b) $\{v_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_v and standard deviation σ_v
   (c) $\{w_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_w and standard deviation σ_w

2. Then we simulate a Poisson sample X from Λ

Figures 2.1, 2.2, and 2.3 show the mean and confidence interval for the absolute value of the cosines of the angles between true and estimated PC1 for the 3 scenarios. The red line shows the results for the Poisson error corrected model while the black line shows the results for the classical PCA. The shaded area is the Confidence Interval of the average cosine of the angle, where CI = MEAN ± 1.96*S.E. and standard error is computed by:

$$\frac{\text{sample standard deviation}}{\sqrt{\text{number of simulations}}}$$

From Figures 2.1, 2.2, 2.3, we find that the Poisson error corrected PCA cosines of the angles are higher than that of the classical PCA, that is particularly true for the large sample sizes, and this pattern is consistent. We can make the conclusion that error corrected model is superior to the classical PCA especially when sample size is large. This is expected, since our method decreases the bias, but increases the variance. As sample size gets large, the increase in variance has less effect, but the reduction in (squared) bias retains the same effect.

2.3 Poisson Error Corrected PCA with Sequencing Depth correction

Let s be the sequencing depth of the latent Microbiome random vector Λ. For simplicity, we will assume the sequencing depth s is known for all observations. (In practice we will assume the sequencing depth of a sample is the total read count of that sample.) Denote the underlying abundance (or proportion) as Λ_c thus $\Lambda = s\Lambda_c$. The conditional random vector $\mathbb{X}|(s, \Lambda_c)$ follows a Poisson distribution with mean $s\Lambda_c$. The goal of this section is to find an unbiased estimator of $\mathrm{Var}(\Lambda_c)$.

By the law of total variance:

$$\begin{aligned}\mathrm{Var}[s^{-1}\mathbb{X}] &= \mathrm{Var}[\mathbb{E}[s^{-1}\mathbb{X}|(s, \Lambda_c)]] + \mathbb{E}[\mathrm{Var}[s^{-1}\mathbb{X}|(s, \Lambda_c)]] \\ &= \mathrm{Var}[\Lambda_c] + \mathbb{E}[s^{-2}\,\mathrm{diag}(\Lambda)]\end{aligned} \tag{2.6}$$

So

$$\Sigma_{\Lambda_c} = \mathrm{Var}[\Lambda_c] = \mathrm{Var}[s^{-1}\mathbb{X}] - \mathbb{E}[s^{-2}\,\mathrm{diag}(\Lambda)] \tag{2.7}$$

The sample version of $\mathrm{Var}(s^{-1}\mathbb{X})$ in equation (2.7) is

$$\mathrm{Var}(s^{-1}X) = \frac{1}{n-1}(\widetilde{S^{-1}X})^T(\widetilde{S^{-1}X}) \tag{2.8}$$

where matrix X is a realization of random vector $\mathbb{X}$,

$$S = \begin{pmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_n \end{pmatrix},$$

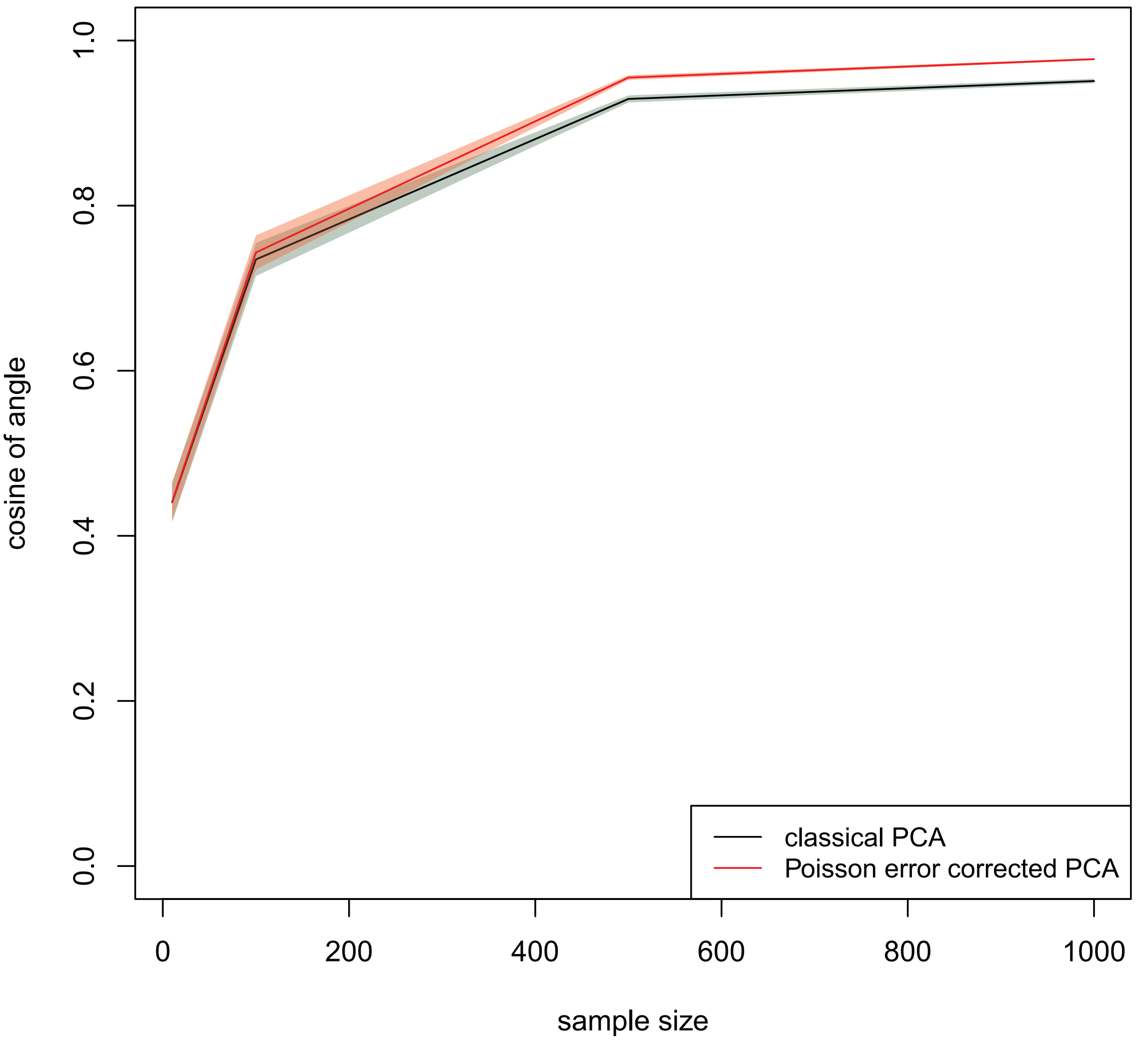


Figure 2.1: Comparison of Linear Corrected Model and Classical PCA for $\mathbf{u} \sim \mathrm{N}(10, 2^2)$, $\mathbf{v} \sim \mathrm{N}(5, 0.1^2)$, $\mathbf{w} \sim \mathrm{N}(30, 5^2)$.

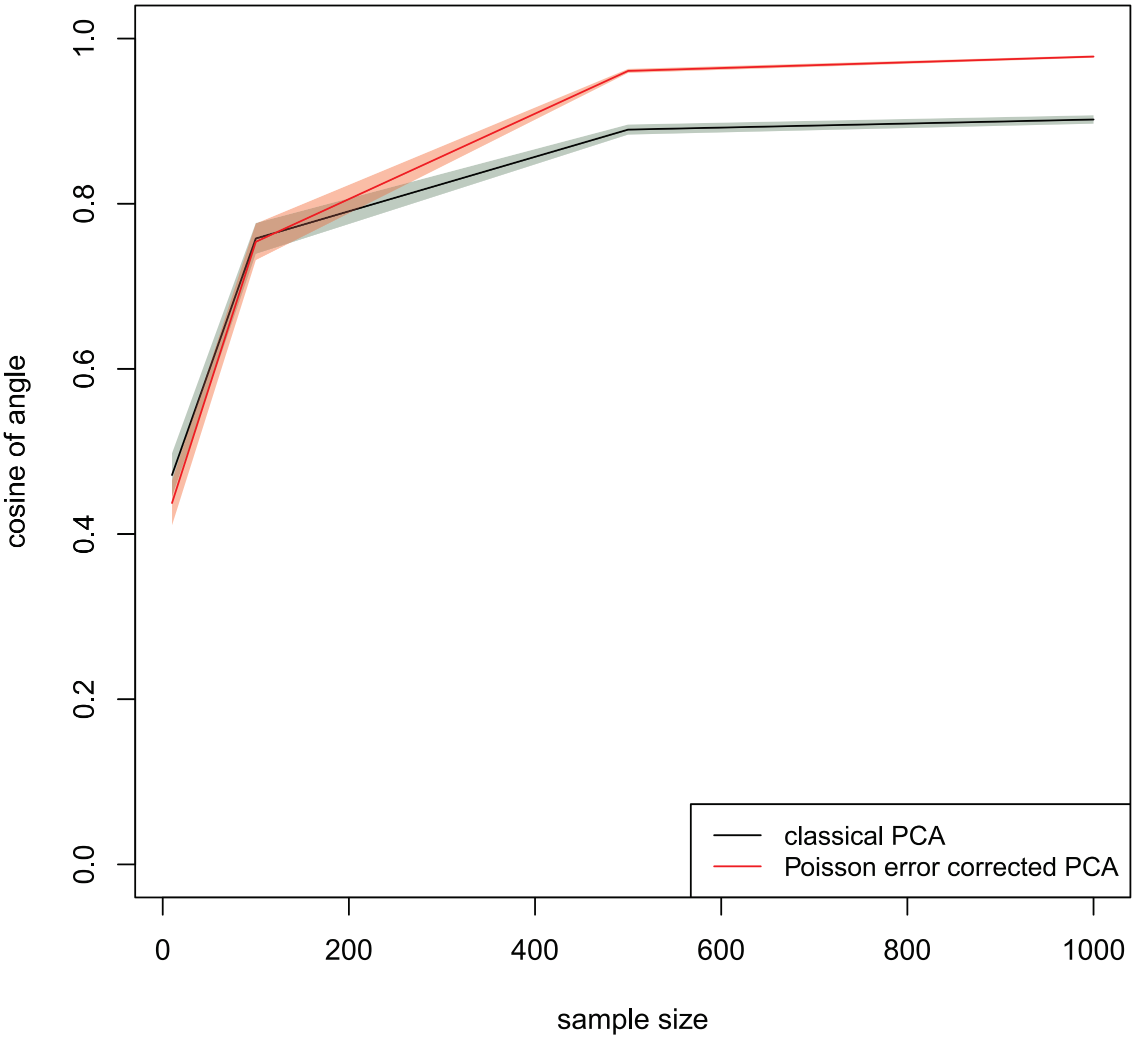


Figure 2.2: Comparison of Linear Corrected Model and Classical PCA for $\mathbf{u} \sim \mathrm{N}(10, 2^2)$, $\mathbf{v} \sim \mathrm{N}(5, 0.1^2)$, $\mathbf{w} \sim \mathrm{N}(30, 15^2)$.

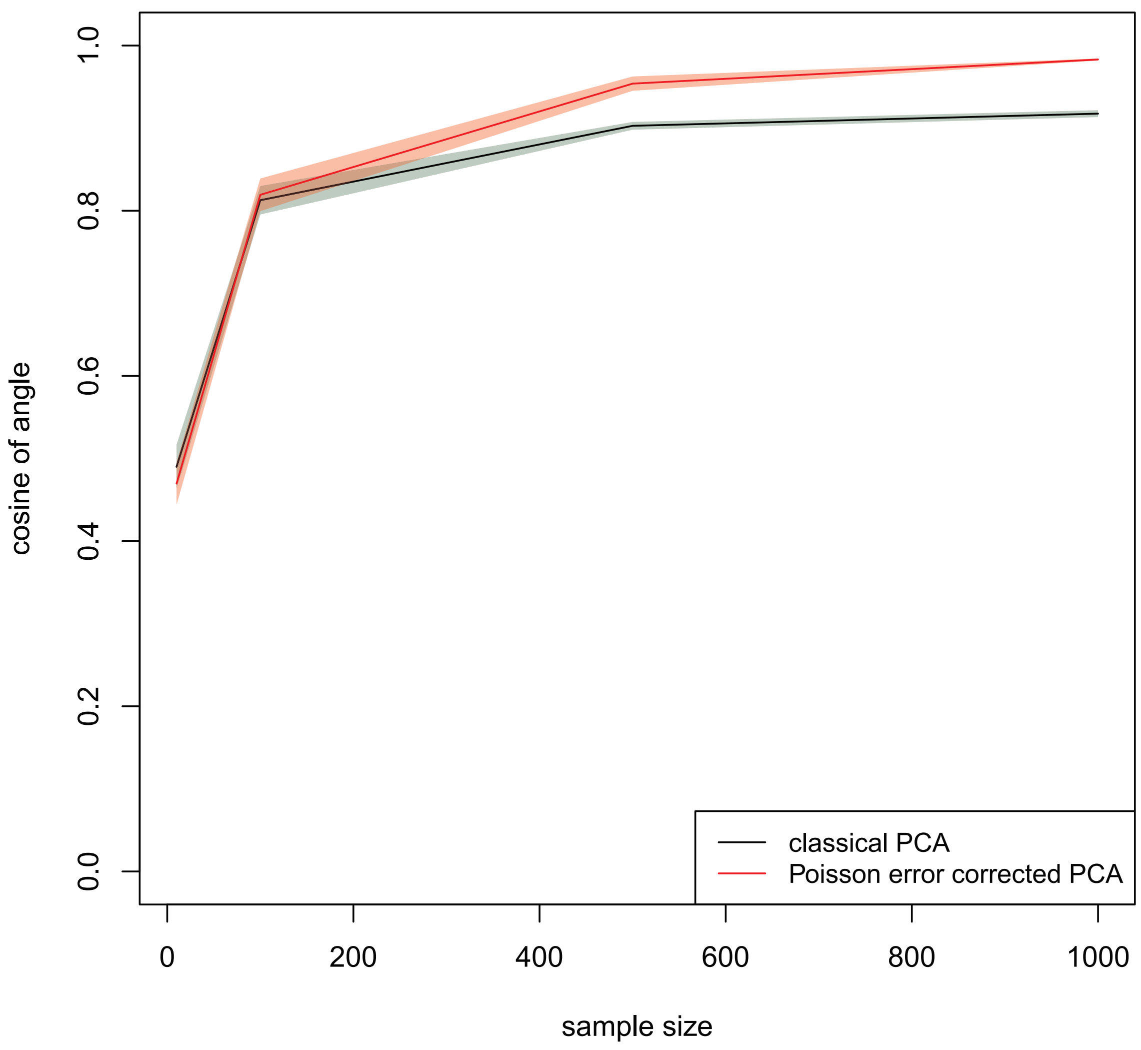


Figure 2.3: Comparison of Linear Corrected Model and Classical PCA for $\mathbf{u} \sim \mathrm{N}(10, 2^2)$, $\mathbf{v} \sim \mathrm{N}(5, 0.1^2)$, $\mathbf{w} \sim \mathrm{N}(30, 25^2)$.

s_i is the sequencing depth of X_i ($i = 1, 2, \cdots, n$), and $\widetilde{S^{-1}X} = S^{-1}X - \mathbf{1}\mathbf{1}^T\frac{S^{-1}X}{n}$.

So an unbiased estimator for Var(Λ_c) is $\frac{1}{n-1}(\widetilde{S^{-1}X})^T(\widetilde{S^{-1}X}) - \text{diag}\left(\frac{(S^{-2}X)^T 1}{n}\right)$.

2.4 Simulation for Principal Component Analysis of Σ_{Λ_c}

In order to test the performance of the Poisson error corrected PCA model with sequencing depth correction, we simulate the true microbiome abundance matrix Λ from two components: sequencing depth $\mathbf{s}$ and compositional form Λ_c, $\mathbf{s} = (s_1, s_2, \cdots, s_n)^T$ is a random vector generated from a normal distribution and $\Lambda_c = \mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T$, where $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are random vectors generated from some different distributions.

We assign appropriate μ_s and σ_s for $\mathbf{s}$ to make it a positive vector with high probability, where $\{s_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_s and standard deviation σ_s. However since rare elements can be non-positive when sample size n is very large, we change all non-positive elements in the latent S into 5. 5 was chosen so that the total abundance of each row of Poisson data X is at least 1. For the simulated distribution of S, there are very few samples with $0 < S < 5$, and no problems arise from leaving these values of S unchanged. This will enable us to compute the inverse matrix of estimated $S = \text{diag}(\sum_{j=1}^p X_{ij})$.

In compositional data Λ_c, $\mathbf{u}$ and $\mathbf{w}$ are random vectors generated from some different normal distributions separately, say $\mathbf{u} = (u_1, u_2, \cdots, u_p)^T$, and $\mathbf{w} = (w_1, w_2, \cdots, w_p)^T$. Each u_j ($j = 1, 2, \cdots, p$) has the same mean $\mu_u = \frac{1}{p}$ (p is the number of variables of Λ) and standard deviation $\sigma_u = \frac{1}{3p}$. In this way, the vast majority of elements in $\mathbf{u}$ are positive and the sum of $\mathbf{u}$ is approximately 1. We can change the non-positive elements into 0 and then center the vector $\mathbf{u}$ so that the sum is exactly 1. Each w_j ($i = 1, 2, \cdots, p$) has the same mean 0 and standard deviation σ_w. We centralize $\mathbf{w}$ to make the requirement $\mathbf{w}^T\mathbf{1} = 0$ hold. Let $\mathbf{w}_0$ be a unit length vector in the direction of $\mathbf{w}$.
We simulate each v_i ($i = 1, 2, \cdots, n$) as a_i plus $b_i - a_i$ times a Beta distribution, where (a_i,b_i) is the interval for v_i which makes every element in row i positive. Namely, we can write the elements in Λ_c as $[u_j + v_i w_j]_{i=1,2,...,n; j=1,2,..,p}$, and the restriction is $u_j + v_i w_j \geq 0$, $j = 1, 2, ..., p$ for the ith row. Since each u_j is positive, we can then get $v_i \geq -\frac{u_j}{w_j}$ if w_j is positive, and $v_i \leq -\frac{u_j}{w_j}$ if w_j is negative. By solving these j inequalities, we get an interval (a_i,b_i) which contains all the possible v_i's. Then we simulate $B \sim Beta(\alpha,\beta)$, and let $v_i = (b_i - a_i)B + a_i$. The parameters α and β are chosen to give $\mathbb{E}(v_i) = 0$ so that $\mathbf{u}$ is the mean of the data.
Now $\Lambda = S(\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T)$, the Variance-Covariance matrix of Λ_c is $\Sigma_{\Lambda_c} = \text{Cov}(\Lambda_c) = \text{Var}(\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T)$.
Suppose u, v, and w are realizations of random vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$, the sample version of the Variance-Covariance matrix Var(Λ_c) is:

$$\begin{aligned}
\widehat{\mathrm{Var}(\Lambda_c)} &= \mathrm{Var}(\mathbf{1}u^T + vw^T) = \mathrm{Var}(vw^T) \\
&= \frac{1}{n-1}(vw^T - \overline{vw^T})^T(vw^T - \overline{vw^T}) \\
&= \frac{1}{n-1}((v-\overline{v})w^T)^T((v-\overline{v})w^T) \\
&= \frac{1}{n-1}w(v-\overline{v})^T(v-\overline{v})w^T \\
&= \hat{\sigma}_v^2 ww^T
\end{aligned}$$

According to the eigen-decomposition rule, the first eigenvector of the above matrix is parallel to **w**, which means that the first principal component of the constructed synthetic data Λ_c is $\mathbf{w}_0$.

If the direction of the PC1 of $\widehat{\Sigma_{\Lambda_c}}$ is closer to that of $\mathrm{Var}(\Lambda_c)$ when using our corrected covariance algorithm than when using uncorrected PCA, we can make the conclusion that our method corrects some of the bias from Poisson error.

We simulate under 2 scenarios, all with $p = 10$, $\mathbf{s} \sim \mathrm{N}(500, 250^2)$, $\mathbf{u} \sim \mathrm{N}(\frac{1}{p}, (\frac{1}{3p})^2)$ where p is the number of variables, $\mathbf{w} \sim \mathrm{N}(0, 1^2)$, and $\mathbf{w} \sim \mathrm{N}(0, 5^2)$. For each scenario, different number of observations, i.e. 10, 50, 100, 300, 500, 1000, are simulated. We simulate 100 Λ matrices by $\Lambda = \mathbf{S}(\mathbf{1u}^T + \mathbf{vw}^T)$ where **S** is the diagonal matrix of **s** and apply the corrected model on the data and compare the results with that from classical PCA on X and on the compositional form $S^{-1}X$ separately.

Figures 2.4 and 2.5 show the mean and confidence interval of the absolute value of the cosine of the angle between the PC1's from three different methods and the true PC1 for the 2 scenarios. The red line shows the results for the Poisson error corrected method with sequencing depth corrected, while the black line shows the results for classical PCA on X and the blue line shows the results for classical PCA on the compositional form of X. The shaded area is the Confidence Interval of the mean cosine of the angle.

From Figures 2.4 and 2.5, we find that the Poisson error corrected PCA with sequencing depth corrected results are better than that of the two classical PCA methods and this pattern is consistent. It is also noticed that for large sample sizes, although the results of the correction is significant, the practical advantages of our method over the PCA on the proportional data is limited. This is mainly because when the sample size is large, the proportions, as estimators for the true composition, are unbiased and the variance estimates need very little correction.

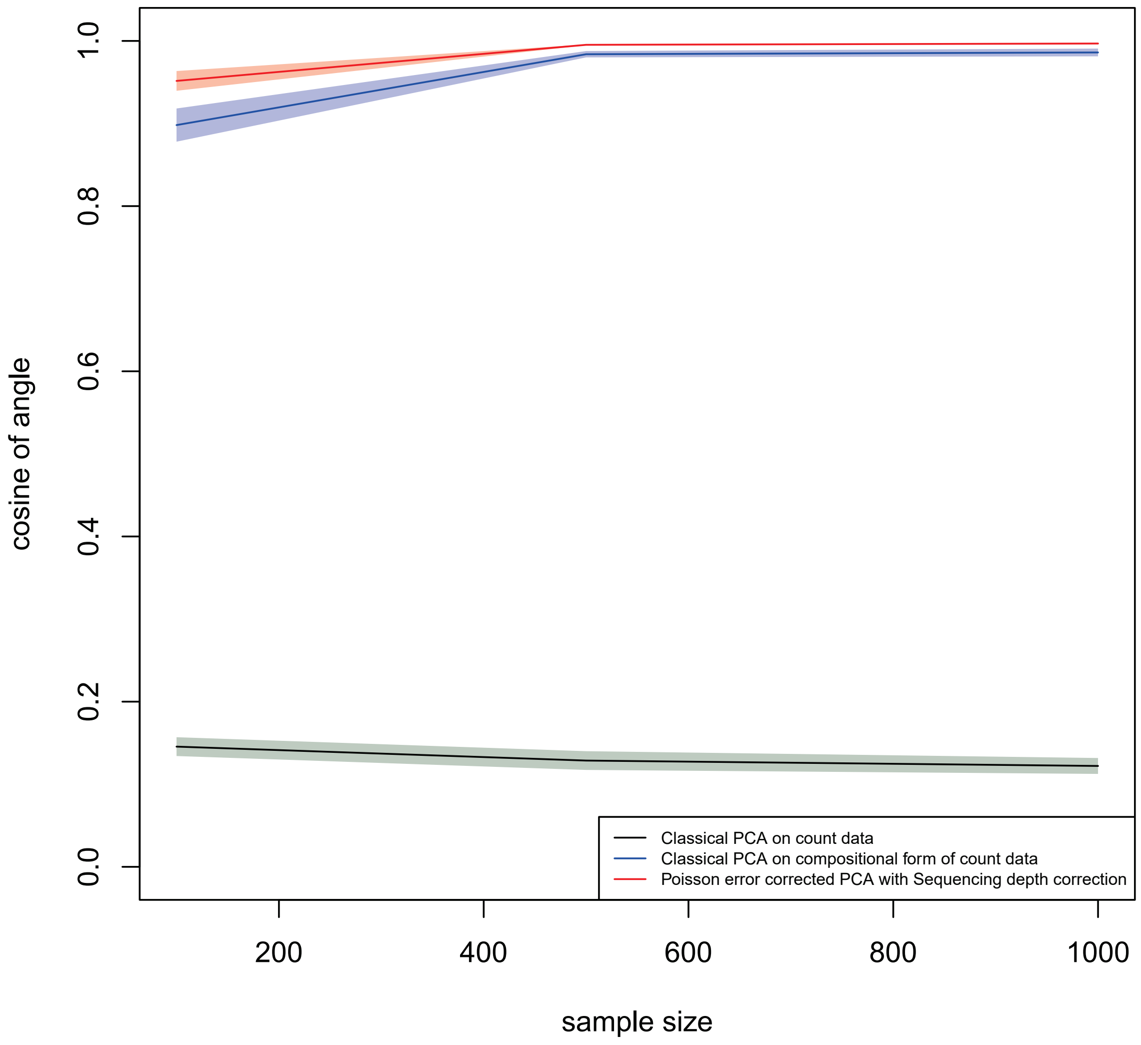


Figure 2.4: Comparison of Poisson error corrected Model with sequencing depth corrected and Classical PCA for X and the compositional form of X, $\mathbf{s} \sim \mathrm{N}(500, 250^2)$, $\mathbf{u} \sim \mathrm{N}(\frac{1}{p}, (\frac{1}{3p})^2)$, $\mathbf{w} \sim \mathrm{N}(0, 1^2)$.

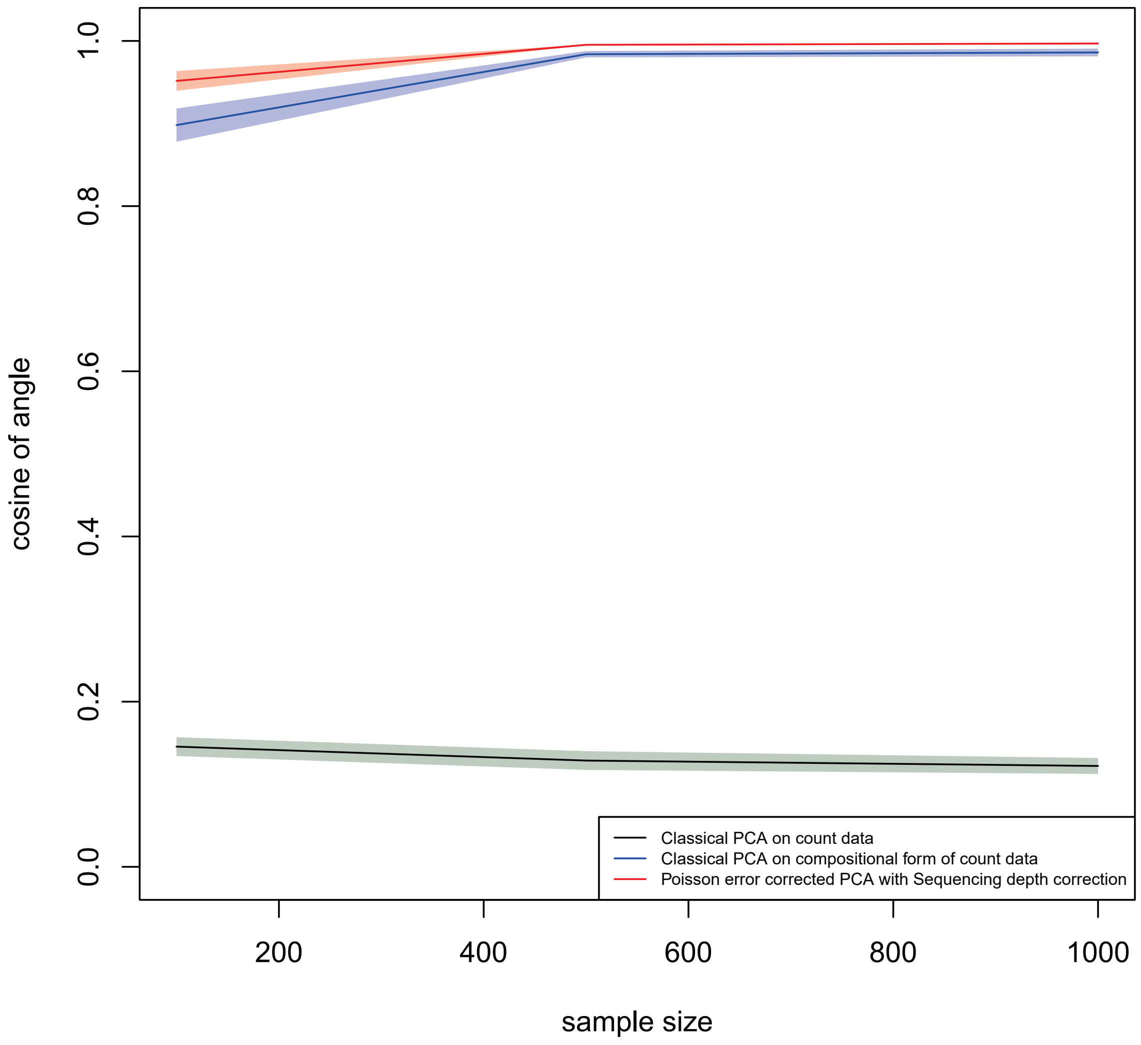


Figure 2.5: Comparison of Poisson error corrected Model with sequencing depth corrected and Classical PCA for X and the compositional form of X, $\mathbf{s} \sim \mathrm{N}(500, 250^2)$, $\mathbf{u} \sim \mathrm{N}(\frac{1}{p}, (\frac{1}{3p})^2)$, $\mathbf{w} \sim \mathrm{N}(0, 5^2)$.

Chapter 3

Principal Component Analysis on Non-linearly Transformed Latent Λ

In Chapter 2, we derived the Poisson error correction for PCA. In Section 2.2, we showed that this method does improve PC estimation compared with applying PCA directly to the X matrix. Sometimes, however we may be more interested in a transformation of the latent variable Λ. In this chapter, we show how the method can be modified to estimate the principal components of the transformed latent variables. Several transformations can be used here, e.g.: logistic, log and square root transformation. Microbiome data can be fitted using a log-normal distribution, so the logarithm transformation is the most appropriate to represent the data features. (A number of papers related to the Microbiome use log scale to measure the data [19] [20]).

3.1 Poisson Noise Corrected log transformed PCA

Suppose that we are now interested in the principal components of $f(\Lambda)$ for some function f. Since $\mathbb{X} \sim \text{Po}(\Lambda)$, we can use the law of total variance as we did for the linear case for any function $g(\mathbb{X})$.

$$\text{Var}(g(\mathbb{X})) = \mathbb{E}(\text{Var}(g(\mathbb{X})|\Lambda)) + \text{Var}(\mathbb{E}(g(\mathbb{X})|\Lambda))$$

We want to choose $g(\mathbb{X})$ to be an unbiased estimator for $f(\Lambda)$ so that $\mathbb{E}(g(\mathbb{X})|\Lambda) = f(\Lambda)$. Then we will have

$$\text{Var}(f(\Lambda)) = \text{Var}(\mathbb{E}(g(\mathbb{X})|\Lambda)) = \text{Var}(g(\mathbb{X})) - \mathbb{E}(\text{Var}(g(\mathbb{X})|\Lambda)) \tag{3.1}$$

For a given $f(X)$ we can solve the following to get the function $g(X)$. Since $X|\Lambda$ follows a Poisson distribution, let $g_n = g(n)$ and we have

$$\mathbb{E}(g(X)|\Lambda = \lambda) = e^{-\lambda} \sum_{n=0}^{\infty} \frac{g_n \lambda^n}{n!}$$
$$e^{\lambda} f(\lambda) = \sum_{n=0}^{\infty} \frac{g_n \lambda^n}{n!}$$

so g_n are the coefficients of the Maclaurin Series of $e^{\lambda} f(\lambda)$.

Now we consider the case of interest $f(\lambda) = \log(\lambda)$. Unfortunately, there is no Maclaurin Series for $f(\lambda) = \log(\lambda)$, because of the singularity at 0. This means that there is no unbiased estimator for $\log(\lambda)$. We can however find a function $f(\lambda)$ which approximates

$\log(\lambda)$ on a range of interest. We can take a Taylor Series about a point a.

$$\begin{aligned}\log(\lambda) &= \log(a) + \log\left(\frac{\lambda}{a}\right)\\ &= \log(a) + \log\left(1 - \left(1 - \frac{\lambda}{a}\right)\right)\\ &= \log(a) - \sum_{m=1}^{\infty} \frac{1}{m}\left(1 - \frac{\lambda}{a}\right)^{m} \qquad (3.2)\end{aligned}$$

When we attempt to sum up the series to calculate the coefficients of λ^m, the series diverges. However, we can truncate the series at some value n_T to obtain some polynomial $p_{n_T,a}(\lambda)$. This will approximate $\log(\lambda)$ on some interval. Suppose we have chosen $p_{n_T,a}(\lambda)$ to approximate $\log(\lambda)$ on an interval that contains all true values of λ. Let $p_{n_T,a}(\lambda) = \sum_{m=0}^{n_T} p_m\lambda^m$.

Selecting the values of a and n is crucial in the expression (3.2). When $\lambda < 2a$, expression (3.2) converges. The accuracy of the approximation increases as the value of n_T increases, but the variance also increases, so we need to find the trade-off between the accuracy and consistency.

Microbiome sequence data has the characteristics of being over-dispersed and highly-skewed, so we can not use the same a and n_T pair in the approximation (3.2) for all the reads. Allowing a and n_T to vary between OTUs may help to solve this problem. However, due to the sequencing depth problem which comes from collecting the data during the experiment, there can still be too many observations that are larger than 2 times the sample mean, which makes the approximation (3.2) not valid.

In this thesis, we propose the following function $h(\lambda)$ to approximate $\log(\lambda)$ in consideration of the above concerns. For small values of λ, we use expression (3.2) with fixed a and n_T. For large λ, we simply use $\log(\lambda)$. Now we seek to find a weight function $\mathrm{m}(\lambda)$ that assigns weight to each piece and ensure the smoothness of the function. We choose $m(\lambda) = \frac{1}{1+e^{-\frac{\lambda-\mu}{s}}}$, where μ and s are two parameters in the cumulative logistic distribution, and let $h(\lambda) = \mathrm{m}(\lambda)\log(\lambda) + (1 - \mathrm{m}(\lambda))\left(\log(a) - \sum_{m=1}^{n_T} \frac{1}{m}\left(1 - \frac{\lambda}{a}\right)^m\right)$ be the function that approximates $\log(\lambda)$.

Figure 3.1 compares these approximations for $\log(\lambda)$, the red line shows the results for $\log(\lambda)$ while the blue line shows the results for the Taylor expansion $p_{n_T,a}(\lambda)$. The two functions both have limitations so that we introduce $h(\lambda)$, which is shown by the black dotted line. Here we set $a = 10$, $n_T = 10$, $\mu = 2$, $s = 1$. Since log(0) is not defined, we plot $\log(\lambda)$ and $h(\lambda)$ from infinitesimal, while the Taylor expansion starts from 0. The (smooth) function $h(\lambda)$ approximates $\log(\lambda)$ very well.

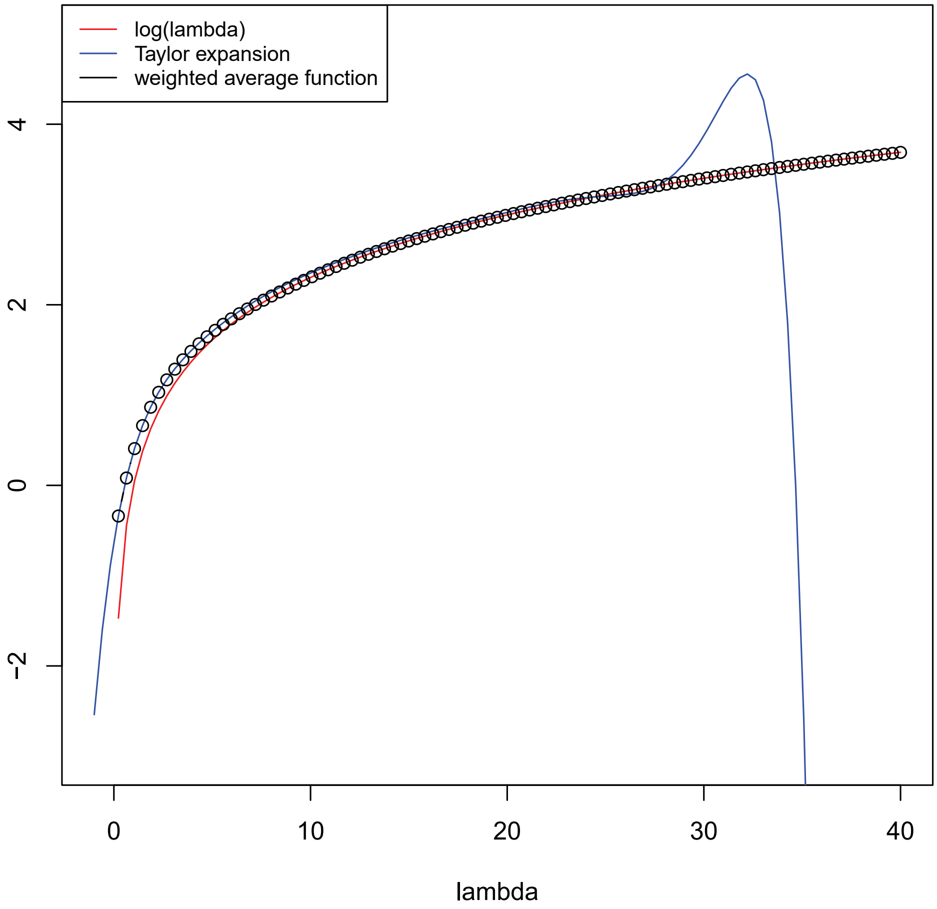


Figure 3.1: The weighted average function to approximate $\log(\lambda)$.

Now we seek to solve

$$\begin{aligned}
e^{\lambda} f(\lambda) &= \sum_{l=0}^{\infty} \frac{g_l \lambda^l}{l!} \\
\left(\sum_{k=0}^{\infty} \frac{\lambda^k}{k!}\right)\left(\sum_{m=0}^{n_T} p_m \lambda^m\right) &= \sum_{l=0}^{\infty} \frac{g_l \lambda^l}{l!} \\
\sum_{k=0}^{\infty} \sum_{m=0}^{n_T} p_m \frac{\lambda^{m+k}}{k!} &= \sum_{l=0}^{\infty} \frac{g_l \lambda^l}{l!} \\
g_l &= l! \sum_{m=0}^{n_T} \frac{p_m}{(l-m)!} \\
&= \sum_{m=0}^{n_T} m! \binom{l}{m} p_m \qquad (3.3)
\end{aligned}$$

where the fourth line follows from the substitution $l = k + m$ on the left hand side. Now we can test the performance of g_k in estimating $\log(\lambda)$.

In Figure 3.2, the red line shows the results for the MSE of $\log(X)$ as an estimator for $\log(\lambda)$ while the blue line shows the results for the MSE of estimating $\log(\lambda)$ by taking the Taylor expansion using g_x from Equation (3.3). For $\lambda < 10$, the estimate using the $\log(X)$ method varies a lot. Meanwhile, the Taylor expansion method is inconsistent when $\lambda > 18$. The dotted black line shows the results for the MSE of estimating $\log(\lambda)$ by the weighted average function $\mathrm{m}(x)\log(x) + (1 - \mathrm{m}(x))\, g_x$ where $\mathrm{m}(x) = \frac{1}{1+e^{-\frac{x-\mu}{s}}}$. Here we set $a = 10$, $n_T = 10$, $\mu = 2$, $s = 1$. In computation, we replace $X = 0$ with $X = 0.1^{-5}$ to make $\log(X)$ valid. When $\lambda = 0$, $\log(\lambda)$ is computed by $\log(0.1^{-5})$, we can then calculate MSE of three estimated $\widehat{\log(\lambda)}$. There is a lot of variance for $g(X)$, because a few large values of X can have very large influence, so the apparent spike at $\lambda = 18$ is caused by a few large influence points appearing in the 1000 simulations. We will use g_k with fixed parameters in the following calculations.

Having calculated the g_k, we now look to calculate the variance of $f(\lambda)$. We calculate

$$\mathbb{E}(g(X)|\Lambda = \lambda) = f(\lambda) \simeq \sum_{k=0}^{\infty} \left(\sum_{m=0}^{n_T} m! \binom{k}{m} p_m\right) \frac{\lambda^k}{k!} e^{-\lambda}$$

and

$$\mathbb{E}(g(X)^2|\Lambda = \lambda) \simeq \sum_{k=0}^{\infty} \left(\sum_{m=0}^{n_T} m! \binom{k}{m} p_m\right)^2 \frac{\lambda^k}{k!} e^{-\lambda}$$

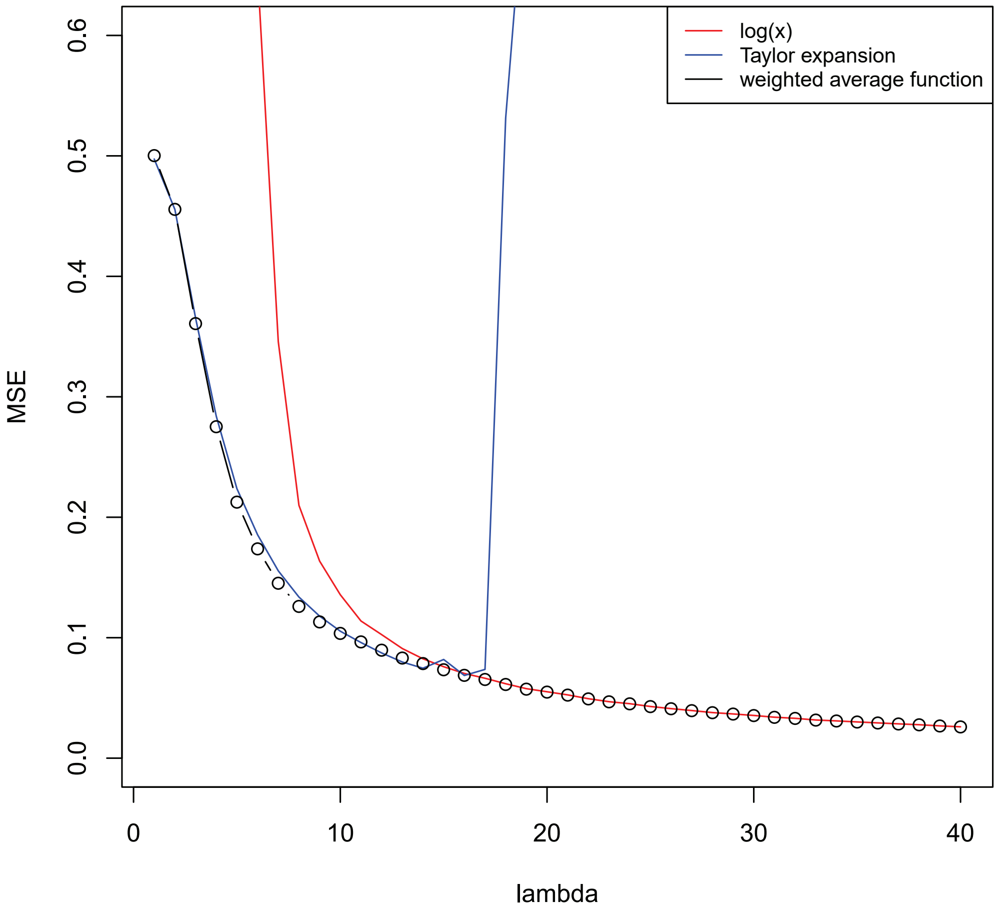


Figure 3.2: The weighted average function as an estimation of log(λ).

so

$$\begin{aligned}\mathrm{Var}(g(X)|\Lambda=\lambda) &= \sum_{k=0}^{\infty}\left(\sum_{m=0}^{n_T} m!\binom{k}{m}p_m\right)^2\frac{\lambda^k}{k!}e^{-\lambda}-\left(\sum_{k=0}^{\infty}\left(\sum_{m=0}^{n_T} m!\binom{k}{m}p_m\right)\frac{\lambda^k}{k!}e^{-\lambda}\right)^2\\ &= \sum_{k=0}^{\infty}\left(\sum_{m=0}^{n_T} m!\binom{k}{m}p_m\right)^2\frac{\lambda^k}{k!}e^{-\lambda}-\sum_{k=0}^{\infty}\left(\sum_{l=0}^{k}\binom{k}{l}\left(\sum_{m=0}^{n_T} m!\binom{k-l}{m}p_m\right)\left(\sum_{j=0}^{n_T} j!\binom{l}{j}p_j\right)\right)\frac{\lambda^k}{k!}e^{-2\lambda}\end{aligned} \tag{3.4}$$

Lemma 3.1.1. *If $X \sim \mathrm{Po}(\lambda)$ then*

$$h_k(X)=\begin{cases}0, & \text{if } X\neq k\\ 1, & \text{if } X=k\end{cases} \tag{3.5}$$

is an unbiased estimator for $\frac{\lambda^k}{k!}e^{-\lambda}$.
Meanwhile,

$$s_k(X)=(-1)^{X-k}\binom{X}{k} \tag{3.6}$$

is an unbiased estimator for $\frac{\lambda^k}{k!}e^{-2\lambda}$. Note that $\binom{X}{k}=0$ when $X<k$.

Proof. By Taylor's expansion,

$$\begin{aligned}\frac{\lambda^k}{k!}e^{-2\lambda} &= \frac{\lambda^k}{k!}e^{-\lambda}\left(\frac{(-\lambda)^0}{0!}+\frac{(-\lambda)^1}{1!}+\frac{(-\lambda)^2}{2!}+\cdots+\frac{(-\lambda)^n}{n!}+\dots\right)\\ &= \frac{\lambda^k}{k!}e^{-\lambda}(-1)^0+\frac{\lambda^{k+1}}{(k+1)!}e^{-\lambda}\frac{(k+1)!}{k!1!}(-1)^1+\frac{\lambda^{k+2}}{(k+2)!}e^{-\lambda}\frac{(k+2)!}{k!2!}(-1)^2+\dots\\ &+\frac{\lambda^{k+n}}{(k+n)!}e^{-\lambda}\frac{(k+n)!}{k!n!}(-1)^n+\dots\\ &= \frac{\lambda^k}{k!}e^{-\lambda}\binom{k}{0}(-1)^0+\frac{\lambda^{k+1}}{(k+1)!}e^{-\lambda}\binom{k+1}{1}(-1)^1+\frac{\lambda^{k+2}}{(k+2)!}e^{-\lambda}\binom{k+2}{2}(-1)^2+\dots\\ &+\frac{\lambda^{k+n}}{(k+n)!}e^{-\lambda}\binom{k+n}{n}(-1)^n+\dots \qquad (3.7)\\ &= e^{-\lambda}\sum_{n=k}^{\infty}\frac{\lambda^n}{n!}(-1)^{n-k}\binom{n}{k} \qquad (3.8)\end{aligned}$$

On the other hand we can expand $\mathbb{E}\left((-1)^{X-k}\binom{X}{k}\right)=e^{-\lambda}\sum_{n=k}^{\infty}\frac{\lambda^n}{n!}(-1)^{n-k}\binom{n}{k}$.

□

Although (3.6) is an unbiased estimator for $\frac{\lambda^k}{k!}e^{-2\lambda}$, it has large variance which will cause error in the subsequent calculation. Instead, we use a Bayesian approach to derive an estimator where the data is given by a single observation x and we assume an improper uniform prior c.

Lemma 3.1.2. *If $X \sim \mathrm{Po}(\lambda)$ where λ follows an improper uniform prior,*
(i) the posterior mean of $\frac{\lambda^k}{k!}e^{-\lambda}$ is

$$2^{-(x+k+1)}\frac{(x+k)!}{x!k!} \tag{3.9}$$

(ii) the posterior mean of $\frac{\lambda^k}{k!}e^{-2\lambda}$ is

$$3^{-(x+k+1)}\frac{(x+k)!}{x!k!} \tag{3.10}$$

Proof. Assume an improper uniform prior c:

$$\begin{aligned} p(\lambda|x) &\propto L(\lambda;x)c \\ &\propto \frac{\lambda^x}{x!}e^{-\lambda}c \\ &\propto \Gamma(\lambda; x+1, 1) \end{aligned} \tag{3.11}$$

The posterior distribution of $\lambda|x$ is $\Gamma(\lambda; x+1, 1)$.

$$\begin{aligned} \mathbb{E}_{\Lambda|X}\left[e^{-\lambda}\frac{\lambda^k}{k!}\right] &= \int_0^\infty \frac{e^{-\lambda}\lambda^k}{k!}\frac{\lambda^x}{x!}e^{-\lambda}\,d\lambda \\ &= \int_0^\infty \frac{e^{-2\lambda}\lambda^{x+k}}{x!k!}\,d\lambda \\ &= \frac{2^{-(x+k)}}{x!k!}\int_0^\infty e^{-2\lambda}(2\lambda)^{x+k}\,d\lambda \\ &= \frac{2^{-(x+k+1)}}{x!k!}(x+k)!\int_0^\infty e^{-2\lambda}(2\lambda)^{x+k}\frac{1}{(x+k)!}\,d(2\lambda) \\ &= 2^{-(x+k+1)}\frac{(x+k)!}{x!k!} \end{aligned} \tag{3.12}$$

So $2^{-(x+k+1)}\frac{(x+k)!}{x!k!}$ is an estimator for $e^{-\lambda}\frac{\lambda^k}{k!}$.

$$\begin{aligned} \mathbb{E}_{\Lambda|X}\left[e^{-2\lambda}\frac{\lambda^k}{k!}\right] &= \int_0^\infty \frac{e^{-2\lambda}\lambda^k}{k!}\frac{\lambda^x}{x!}e^{-\lambda}\,d\lambda \\ &= \int_0^\infty \frac{e^{-3\lambda}\lambda^{x+k}}{x!k!}\,d\lambda \\ &= \frac{3^{-(x+k)}}{x!k!}\int_0^\infty e^{-3\lambda}(3\lambda)^{x+k}\,d\lambda \\ &= \frac{3^{-(x+k+1)}}{x!k!}(x+k)!\int_0^\infty e^{-3\lambda}(3\lambda)^{x+k}\frac{1}{(x+k)!}\,d(3\lambda) \\ &= 3^{-(x+k+1)}\frac{(x+k)!}{x!k!} \end{aligned} \tag{3.13}$$

So $3^{-(x+k+1)}\frac{(x+k)!}{x!k!}$ is an estimator for $e^{-2\lambda}\frac{\lambda^k}{k!}$. □

Plugging the estimators in (3.5) and (3.13) into (3.4) gives that

$$u(X) = \left(\sum_{m=0}^{n_T} m!\binom{X}{m}p_m\right)^2 - \sum_{k=0}^{\infty}\left(\sum_{l=0}^{k}\binom{k}{l}\left(\sum_{m=0}^{n_T} m!\binom{k-l}{m}p_m\right)\left(\sum_{j=0}^{n_T} j!\binom{l}{j}p_j\right)\right)3^{-(x+k+1)}\frac{(x+k)!}{x!k!} \tag{3.14}$$

is an estimator for $\mathrm{Var}[g(X)|\Lambda]$. If we are using the weighted average function $h(X) = \mathrm{m}(X)\log(X) + (1 - m(X))g(X)$, then we have that

$$\mathrm{Var}(h(X)|\Lambda = \lambda) = \sum_{k=0}^{\infty}(h(k))^2\frac{\lambda^k}{k!}e^{-k} - \sum_{k=0}^{\infty}\left(\sum_{l=0}^{k}h(l)h(k-l)\right)\frac{\lambda^k}{k!}e^{-2\lambda}$$

so

$$v(x) = (h(X))^2 - \sum_{k=0}^{\infty}\left(\sum_{l=0}^{k}h(l)h(k-l)\right)3^{-(x+k+1)}\frac{(x+k)!}{x!k!} \tag{3.15}$$

is an estimator for $\mathrm{Var}(h(X)|\Lambda)$.

Subtracting $\sum\frac{u(X)}{n}$ from the sample variance $\frac{\sum\left(g(X)-\sum\frac{g(X)}{n}\right)^2}{n-1}$ gives an estimator of $\mathrm{Var}(\mathbb{E}(g(X)|\lambda)) = \mathrm{Var}(f(\Lambda))$. Similarly, we can subtract $\sum\frac{v(X)}{n}$ from the sample variance $\frac{\sum\left(h(X)-\sum\frac{h(X)}{n}\right)^2}{n-1}$. For the off-diagonal terms, we have $\mathrm{Cov}(g(X_1), g(X_2)|\Lambda_1 = \lambda_1, \Lambda_2 = \lambda_2) = 0$, so $\mathrm{Cov}(f(\lambda_1), f(\lambda_2)) = \mathrm{Cov}(g(X_1), g(X_2))$, so only the diagonal elements of the variance matrix need to be modified.

3.2 Simulation for Principal Component Analysis of $\Sigma_{\log(\Lambda)}$

In order to examine the performance of the Poisson noise corrected log transformed PCA, we simulate $\Lambda = e^{\mathbf{1u}^T+\mathbf{vw}^T}$, where $\mathbf{u}$,$\mathbf{v}$ and $\mathbf{w}$ are random vectors. $\mathbf{u} = (u_1, u_2, \cdots, u_p)^T$, $\mathbf{v} = (v_1, v_2, \cdots, v_n)^T$ and $\mathbf{w} = (w_1, w_2, \cdots, w_p)^T$. Each u_i $(i = 1, 2, \cdots, p)$ has the same mean μ_u and standard deviation σ_u, similarly $\mathbf{v}$ follows a $\mathrm{N}(\mu_v, \sigma_v)$ distribution and $\mathbf{w}$ follows a $\mathrm{N}(\mu_w, \sigma_w)$ distribution.

Let $\mathbf{w}_0$ be a unit length vector in the direction of $\mathbf{w}$. Now, $\mathbf{w}_0$ is the first principal component (PC1) of $\log(\Lambda)$.

We generate the observed X from its Poisson mean Λ. By applying our method on X, we can compare its result with the true principal component $\mathbf{w}_0$, to determine whether our method corrects the Poisson error.

The simulation procedure is the same as in Section 2.1.

We simulate under 2 scenarios, all with $p = 10$, $\mathbf{u} \sim \mathrm{N}(0.5, 0.5^2)$ and $\mathbf{v} \sim \mathrm{N}(0.5, 0.5^2)$, one scenario with $\mathbf{w} \sim \mathrm{N}(0, 1^2)$, and the other with $\mathbf{w} \sim \mathrm{N}(0, 2^2)$. We simulate 10, 50, 100, 500, 1000 observations for each scenarios. We simulate 100 Λ matrices by $\Lambda = e^{\mathbf{1u}^T+\mathbf{vw}^T}$ and simulate one X matrix for each Λ.

We record the mean and variance of the cosine of the angle between the estimated PC1 and the true direction $\mathbf{w}_0$ by our method and by applying PCA on X and $\log(X)$ to make comparisons.

Figures 3.3 and 3.4 show the average absolute value of the cosine of the angle between the estimated PC1 and the true direction for the 2 scenarios. The red line is for the results for the Poisson error corrected log transformed PCA while the black line shows the results for classical PCA on X and the blue line shows the results for classical PCA on $\log(X)$. The shaded area is the 95% Confidence Interval of the mean absolute value of the cosine of the angle.

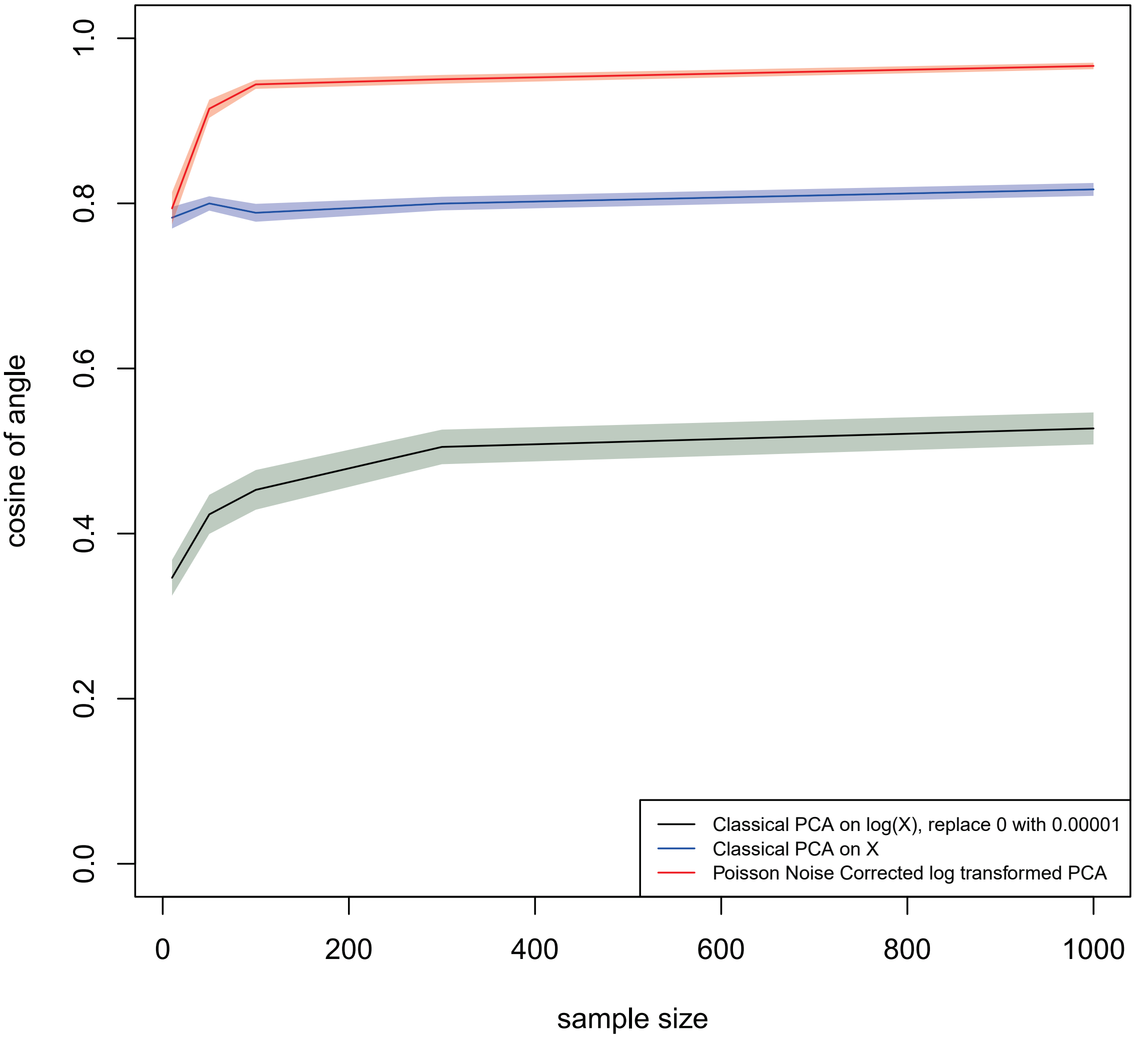


Figure 3.3: Comparison of Poisson noise corrected log transformed Model and Classical PCA on X and $\log(X)$, where $\mathbf{u} \sim N(0.5, 0.5^2)$, $\mathbf{v} \sim N(0.5, 0.5^2)$, $\mathbf{w} \sim N(0, 1^2)$.

From the figures we find that the Poisson noise corrected log transformed PCA performs much better than classical PCA, both on X and on $\log(X)$, and the result shows good consistency.

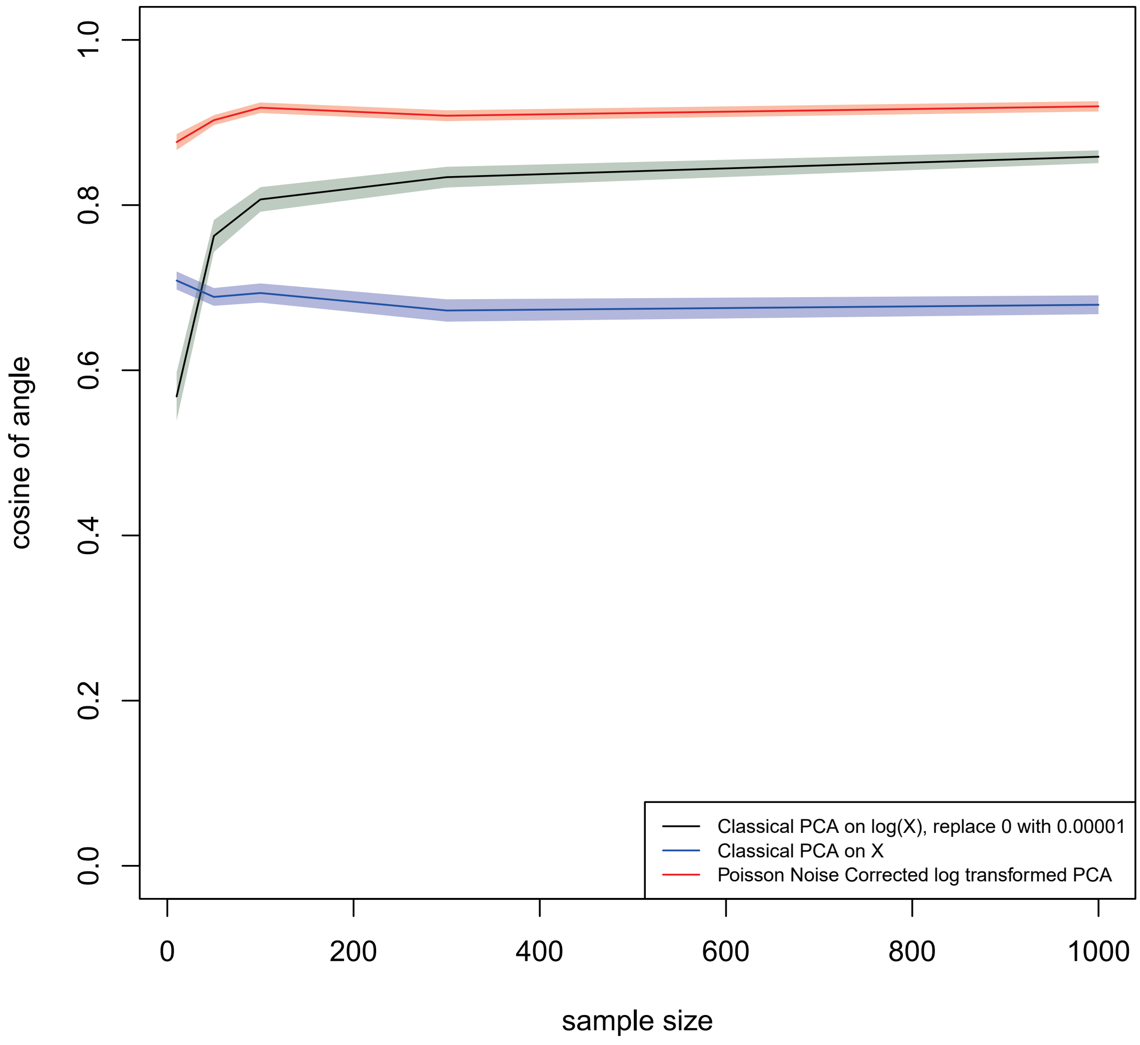


Figure 3.4: Comparison of Poisson noise corrected log transformed Model and Classical PCA on X and $\log(X)$, where $\mathbf{u} \sim \mathrm{N}(0.5, 0.5^2)$, $\mathbf{v} \sim \mathrm{N}(0.5, 0.5^2)$, $\mathbf{w} \sim \mathrm{N}(0, 2^2)$.

3.3 Discussion

We approximate $\log(\lambda) \approx p_{a,n_T}(\lambda)$ and use a Taylor series to obtain $g_k(X)$ as an estimator for $p_{a,n_T}(\lambda)$.

We need to choose values a and n_T of the Taylor series to achieve a good estimate of $\Sigma_{f(\lambda)}$. There are two criteria we need to consider. Firstly the difference between $p_{a,n_T}(\lambda)$ and $\log(\lambda)$ induces a bias in our estimate. For the bias, we know that the radius of convergence of the Taylor series is a, so for $\lambda > 2a$, the bias is very large. We therefore want to choose a large enough so that there are few or no data points with $\lambda > 2a$. Having chosen such an a, the Taylor series will converge to $\log(\lambda)$, so the larger n_T, the smaller the bias $\log(\lambda) - p_{a,n_T}(\lambda)$. However, larger n_T also increases the variance of $g_{a,n_T}(X)$, and of $u_{a,n_T}(X)$. Therefore we need to choose n_T to balance these two constraints.

Theoretically, if we can find suitable a and n_T for each OTU of the Microbiome data, $g_k(X)$ is an unbiased estimator for $p_{a,n_T}(\lambda)$. Empirically, the Microbiome data we observed from experiment contains sequencing depth and that may cause high variance for a single OTU. It is very likely to observe reads of one OTU that vary from 0 to 5000 with mean 1000. In this situation, we cannot find suitable values of a and n_T that work for all the reads in the form of $g_k(X)$ as an approximation of $\log(\lambda)$ to meet the requirements of $\lambda < 2a$ and the variance in $g_{a,n_T}(X)$ is not too large.

In order to solve the problem, we introduce a weight function $\mathrm{m}(\lambda) = \frac{1}{1+e^{-\frac{\lambda-\mu}{s}}}$, then the weighted average function $h(\lambda) = \mathrm{m}(\lambda)\log(\lambda) + (1 - \mathrm{m}(\lambda))\left(\log(a) - \sum_{m=1}^{n_T} \frac{1}{m}(1 - \frac{\lambda}{a})m\right)$ approximates $\log(\lambda)$. We choose parameters for $m(\lambda)$ and the values of a and n_T, and examine the approximation $h(\lambda)$ and the MSE of using $g_k(X)$ as an estimator of $\log(\lambda)$. The comparison can be found in Section 3.1, and the Figures 3.1 and 3.2 both illustrate that the weighted function $g_{a,n_T}(X)$ is a good estimator of $\log(\lambda)$.

Chapter 4

Poisson Error Corrected log transformed PCA with Sequencing Depth Correction

In Section 2.3, we have dealt with the Poisson error corrected PCA on Λ when sequencing depth is subject to large noise.

Now let $\Lambda = \mathbf{s}\Lambda_0$ for some random variable $\mathbf{s}$, where Λ_0 is a form of underlying abundance, and PCA of $\log(\Lambda_0)$ is of interest. (In this chapter we do not insist that Λ_0 be compositional, because that is incompatible with our assumption that the log-transformed scale is the important scale on which to view the data.) $\mathbf{s}$ represents the sequencing depth which is a random variable and unobserved, and Λ_0 is not necessarily compositional.

In Section 3.1, we have calculated an estimate for the Variance-Covariance matrix Σ of $\log(\Lambda)$, where Λ is the Poisson mean of X. That means we have estimated the total variance of $\log(\mathbf{s}\Lambda_0)$. When both $\mathbf{s}$ and Λ_0 are not observable, in principle, $\mathrm{Var}(\log(\Lambda_0))$ is not identifiable. We are going to derive two approaches to estimate the covariance matrix Σ_0 of $\log(\Lambda_0)$ from our estimate for Σ by adding two different types of constraints .

4.1 Method I: Composition Restricted Variance

It is natural to look for the Variance-Covariance matrix of $\log(\Lambda_0)$ to be the Variance-Covariance matrix of a compositional random vector. Thus we estimate the Variance-Covariance matrix Σ_c as the closest matrix which could be the Variance-Covariance matrix of a compositional random vector. Under such a condition, we can consider the estimate under the following constraints:

1. For any $\mathbf{v}\perp\mathbf{1}$ and $\mathbf{w}\perp\mathbf{1}$, we have $\mathbf{v}^T\Sigma_0\mathbf{w} = \mathbf{v}^T\Sigma\mathbf{w}$
2. Σ_c is a symmetric matrix
3. $\Sigma_c\mathbf{1} = 0$

It is straight forward that the variance covariance matrix Σ_c of compositional data has the properties 2 and 3. The property 3 means the space spanned by the matrix Σ_c is orthogonal to vector $\mathbf{1}$. We add a constraint that Σ_c and Σ induce the same norm in the space orthogonal to vector $\mathbf{1}$, which is expressed by property 1.

Proposition 4.1.1. *Under the 3 constraints, there is an unique solution for Σ_c and it is given by $\Sigma_c = \Sigma - \boldsymbol{1}\boldsymbol{a}^T - \boldsymbol{a}\boldsymbol{1}$ where $\boldsymbol{a} = \left(pI + \boldsymbol{1}\boldsymbol{1}^T\right)^{-1}\Sigma\boldsymbol{1}$.*

Proof. Suppose $\Sigma_c = \Sigma - A$ where A is some $p \times p$ symmetric matrix which will be defined later.

$$\mathbf{v}^T\Sigma_c\mathbf{w} = \mathbf{v}^T\Sigma\mathbf{w} - \mathbf{v}^T A\mathbf{w}$$
$$0 = \mathbf{v}^T A\mathbf{w}$$

Since $(A\mathbf{v})^T\mathbf{w} = 0$ holds for any $\mathbf{w}\perp\mathbf{1}$, so $A\mathbf{v} \propto \mathbf{1}$, thus $A\mathbf{v} = d\mathbf{1}$, where d is some scalar to make the equation holds and d is a linear function of $\mathbf{v}$, $d = \mathbf{a}^T\mathbf{v}$ for some $\mathbf{a}$.

For any $\mathbf{v}\perp\mathbf{1}$, we have $A\mathbf{v} = \mathbf{a}^T\mathbf{v}\mathbf{1} = \mathbf{1}\mathbf{a}^T\mathbf{v}$ for some $\mathbf{a}$ thus we have $(A - \mathbf{1}\mathbf{a}^T)\mathbf{v} = \mathbf{0}$. This gives $A = \mathbf{1}\mathbf{a}^T + W$ where $W = \phi\mathbf{1}^T$ for some vector ϕ. Due to the second constraint, A is a symmetric matrix, so $\phi = \mathbf{a}$. Thus $\Sigma_c = \Sigma - \mathbf{1}\mathbf{a}^T - \mathbf{a}\mathbf{1}^T$.

The next step is to solve for vector $\mathbf{a}$ in the above equation:

$$\begin{aligned}
\Sigma_c\mathbf{1} &= \Sigma\mathbf{1} - \mathbf{1}\mathbf{a}^T\mathbf{1} - \mathbf{a}\mathbf{1}^T\mathbf{1} \\
\mathbf{0} &= \Sigma\mathbf{1} - \mathbf{1}(\mathbf{a}^T\mathbf{1}) - \mathbf{a}(\mathbf{1}^T\mathbf{1}) \\
\left(I + \frac{\mathbf{1}\mathbf{1}^T}{p}\right)\mathbf{a} &= \frac{\Sigma\mathbf{1}}{p} \\
\mathbf{a} &= \left(pI + \mathbf{1}\mathbf{1}^T\right)^{-1}\Sigma\mathbf{1}
\end{aligned}$$

Plugging $\mathbf{a}$ into $\Sigma_c = \Sigma - \mathbf{1}\mathbf{a}^T - \mathbf{a}\mathbf{1}^T$ will give us the Variance-Covariance matrix of $\log(\Lambda_0)$ □

4.2 Method II: Minimum Variance Constraint

Assuming the sequencing depth variable is independent of random vector Λ_0, we have that

$$\begin{aligned}
\Sigma_\Lambda &= \mathrm{Var}(\log(s\Lambda_0)) \\
&= \mathrm{Var}(\log(s)\mathbf{1} + \log(\Lambda_0)) \\
&= \mathrm{Var}(\log(s)\mathbf{1}) + \mathrm{Var}(\log(\Lambda_0)) \\
&= \mathbf{1}\,\mathrm{Var}(\log(s))\mathbf{1}^T + \Sigma_0 \\
&= \sigma_s^2\mathbf{1}\mathbf{1}^T + \Sigma_0
\end{aligned}$$

where $\sigma_s^2 = \mathrm{Var}(\log(s))$

We cannot determine the true value of σ_s^2, but the constraint that Σ_0 is non-negative definite gives a maximum for σ_s^2, which gives the minimum total variance for Σ_0. This means we attribute as much as possible of the total variance to the sequencing depth. This maximum value of σ_s^2 occurs when there is an eigenvalue reduced to 0. That is σ_s^2 is the smallest solution to $|\Sigma - \sigma_s^2\mathbf{1}\mathbf{1}^T| = 0$.

Proposition 4.2.1. *The largest σ_s^2 for Σ_0 to be non-negative definite is given by $\sigma_s^2 = \frac{|\Sigma|}{p|\Sigma^*|}$, where $\Sigma^* = \Sigma_c + \frac{\mathbf{1}\mathbf{1}^T}{p}$ and Σ_c is as calculated in Proposition 4.1.1*

Proof. For any matrix B, $|B(\Sigma - \sigma_s^2\mathbf{1}\mathbf{1}^T)B^{-1}| = 0$. We choose B so that:

1.$B\mathbf{1} = (\sqrt{p}, 0, \cdots, 0)^T = \sqrt{p}\mathbf{e}_1$.

2.$B^T = B^{-1}$

Let $M = B\Sigma B^{-1}$, and let the (i, j)th element of M be M_{ij}.

$$
\begin{aligned}
0 &= \left| B\Sigma B^{-1} - p\sigma_s^2 \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \right| = \begin{vmatrix} M_{11} - p\sigma_s^2 & M_{12} & \cdots & M_{1p} \\ M_{21} & M_{22} & \cdots & M_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ M_{p1} & M_{p2} & \cdots & M_{pp} \end{vmatrix} \\
&= |B\Sigma B^{-1}| - p\sigma_s^2 \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 0 & M_{22} & \cdots & M_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & M_{p2} & \cdots & M_{pp} \end{vmatrix} = |\Sigma| - p\sigma_s^2 |M - \mathbf{e}_1\mathbf{b}^T - \mathbf{b}\mathbf{e}_1^T + \mathbf{e}_1\mathbf{e}_1^T| \\
&= |\Sigma| - p\sigma_s^2 |M^*| \qquad (4.1)
\end{aligned}
$$

where $\mathbf{b}$ is chosen so that $(M - \mathbf{e}_1\mathbf{b}^T - \mathbf{b}\mathbf{e}_1^T)\mathbf{e}_1 = \mathbf{0}$, so $\mathbf{b} = (\frac{M_{11}}{2}, M_{12}, \cdots, M_{1p})$ and $M_{1j} = M_{j1}$ for any $j = 1, 2, \cdots, p$.

Then we can solve for σ_s^2 in equation (4.1): $\sigma_s^2 = \frac{|\Sigma|}{p|M^*|}$.

Define $\Sigma^* = B^{-1}M^*B$, thus $|M^*| = |\Sigma^*|$

$$
\begin{aligned}
\Sigma^* &= B^{-1}MB - B^{-1}\mathbf{e}_1\mathbf{b}^T B - B^{-1}\mathbf{b}\mathbf{e}_1^T B + B^{-1}\mathbf{e}_1\mathbf{e}_1^T B \\
&= \Sigma - \frac{1}{\sqrt{p}}\mathbf{1}(\mathbf{b}^T B) - \frac{1}{\sqrt{p}}(B^{-1}\mathbf{b})\mathbf{1}^T + (\frac{1}{\sqrt{p}}\mathbf{1})(\mathbf{1}^T\frac{1}{\sqrt{p}}) \\
&= \Sigma - \mathbf{1}\frac{(B^{-1}\mathbf{b})^T}{\sqrt{p}} - \frac{(B^{-1}\mathbf{b})}{\sqrt{p}}\mathbf{1}^T + \frac{1}{p}\mathbf{1}\mathbf{1}^T \\
&= \Sigma - \mathbf{1}\mathbf{c}^T - \mathbf{c}\mathbf{1}^T + \frac{1}{p}\mathbf{1}\mathbf{1}^T
\end{aligned}
$$

where $\mathbf{c} = \frac{B^{-1}\mathbf{b}}{\sqrt{p}}$.

$$
\begin{aligned}
(M - \mathbf{e}_1\mathbf{b}^T - \mathbf{b}\mathbf{e}_1^T)\mathbf{e}_1 &= 0 \\
B^{-1}(M - \mathbf{e}_1\mathbf{b}^T - \mathbf{b}\mathbf{e}_1^T)BB^{-1}\mathbf{e}_1 &= 0 \\
\left(\Sigma - \frac{\mathbf{1}(\mathbf{b}^T B)}{\sqrt{p}} - \frac{(B^{-1}\mathbf{b})\mathbf{1}^T}{\sqrt{p}}\right)\mathbf{1} &= 0 \\
\frac{1}{\sqrt{p}}\left(\mathbf{1}(\mathbf{b}^T B\mathbf{1}) + (B^{-1}\mathbf{b})\mathbf{1}^T\mathbf{1}\right) &= \Sigma\mathbf{1} \\
\mathbf{1}\mathbf{c}^T\mathbf{1} + \mathbf{c}\mathbf{1}^T\mathbf{1} &= \Sigma\mathbf{1}
\end{aligned}
$$

thus $\mathbf{1}\mathbf{1}^T\mathbf{c} + p\mathbf{c} = \Sigma\mathbf{1}$. Solving for $\mathbf{c}$, we will get $\mathbf{c} = (\mathbf{1}\mathbf{1}^T + pI)^{-1}\Sigma\mathbf{1}$, thus

$$
\begin{aligned}
\Sigma^* &= \Sigma - \mathbf{1}\mathbf{c}^T - \mathbf{c}\mathbf{1}^T + \frac{\mathbf{1}\mathbf{1}^T}{p} \\
&= \Sigma_c + \frac{\mathbf{1}\mathbf{1}^T}{p}
\end{aligned}
$$

where Σ_c is the same as that defined in Proposition 4.1.1.

□

4.3 Simulation for Principal Component Analysis of log transformed PCA with Sequencing Depth Correction

We simulate under 2 scenarios, all with $p = 10$, $\mathbf{s} \sim \mathrm{N}(2, 1^2)$, $\mathbf{u} \sim \mathrm{N}(0.5, 0.1^2)$, $\mathbf{v} \sim \mathrm{N}(1, 0.5^2)$. In the first scenario we simulate $w \sim N(0, 1^2)$, while in the second scenario, we let $w \sim N(0, 2^2)$.

For each scenario, we simulate 100 Λ matrices for sample sizes 10, 50, 100, 300, 500, 1000 by $\Lambda = S e^{\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T}$, where S is the diagonal matrix of $\mathbf{s}$. Then we apply both models on the data and compare the result with that from classical PCA on X and on log of the compositional form of X separately.

Figures 4.1 and 4.2 show the average absolute cosine of the angle between the truth and the estimated PC1 for the 2 scenarios. The red line shows the results for method I of the Poisson error corrected log transformed PCA with sequencing depth correction, while the blue line shows the results for classical PCA on X and the black line shows the results for classical PCA on log of the composition form of X. The shaded area is the 95% Confidence Interval of the average cosine of the angle.

From Figures 4.1 and 4.2, we can also observe that the performances of these two classical methods vary a lot for different sample sizes and variations of $\mathbf{w}$. However, for the large sample size, applying PCA on log of the compositional form of X seem to capture the most information of the true PC1 especially for the case when the standard deviation of $\mathbf{w}$ is large. If we consider three different methods from all the situations as a whole, the Poisson error corrected log transformed PCA with sequencing depth correction outperforms the classical methods and this pattern is consistent.

Similar plots for method II are presented in Figure 4.3

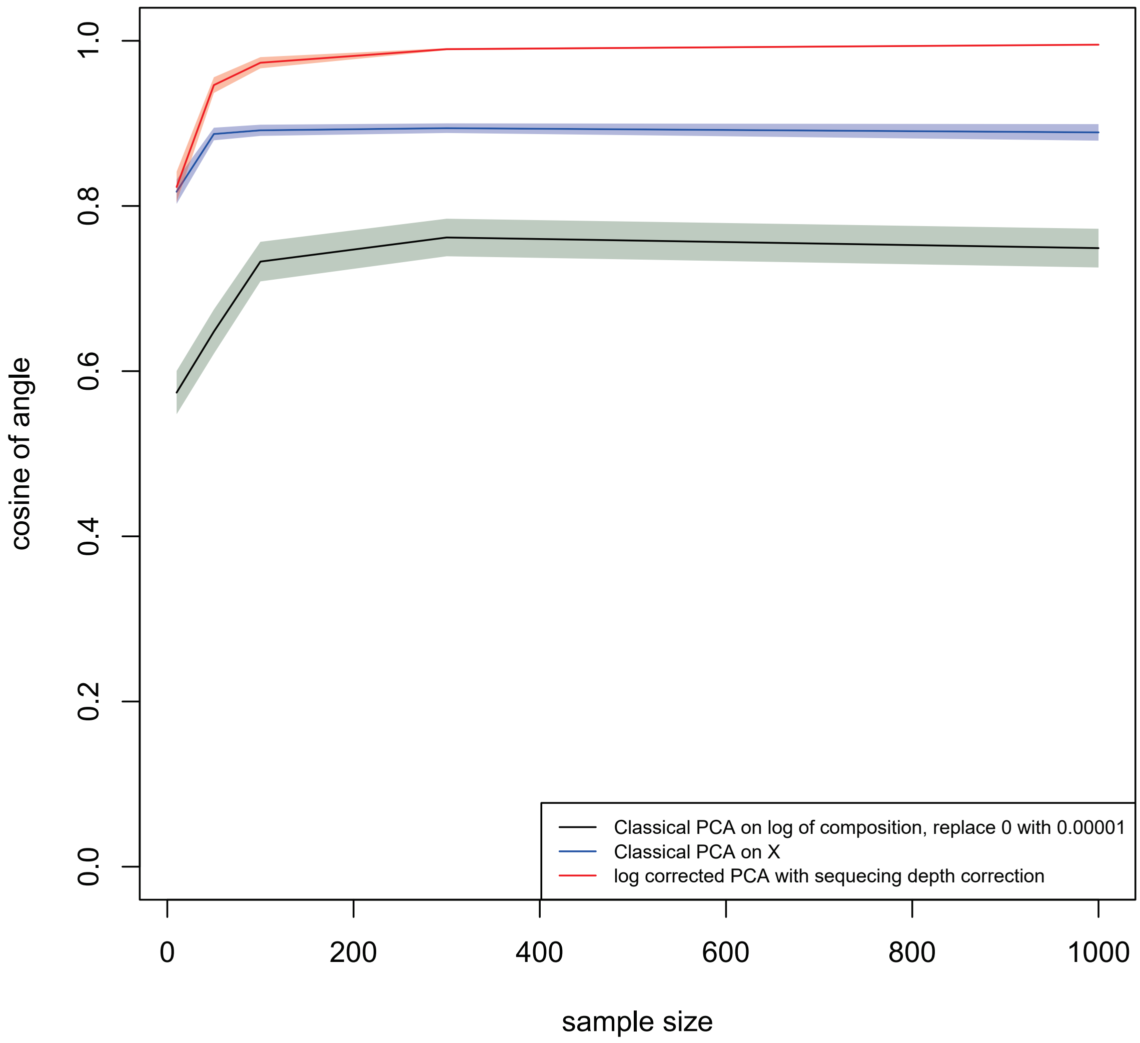


Figure 4.1: Comparison of Poisson error corrected log transformed PCA with sequencing depth correction (Method I) and Classical PCA where $\mathbf{u} \sim \mathrm{N}(0.5, 0.1^2)$, $\mathbf{v} \sim \mathrm{N}(1, 0.5^2)$, $\mathbf{w} \sim \mathrm{N}(0, 1^2)$.

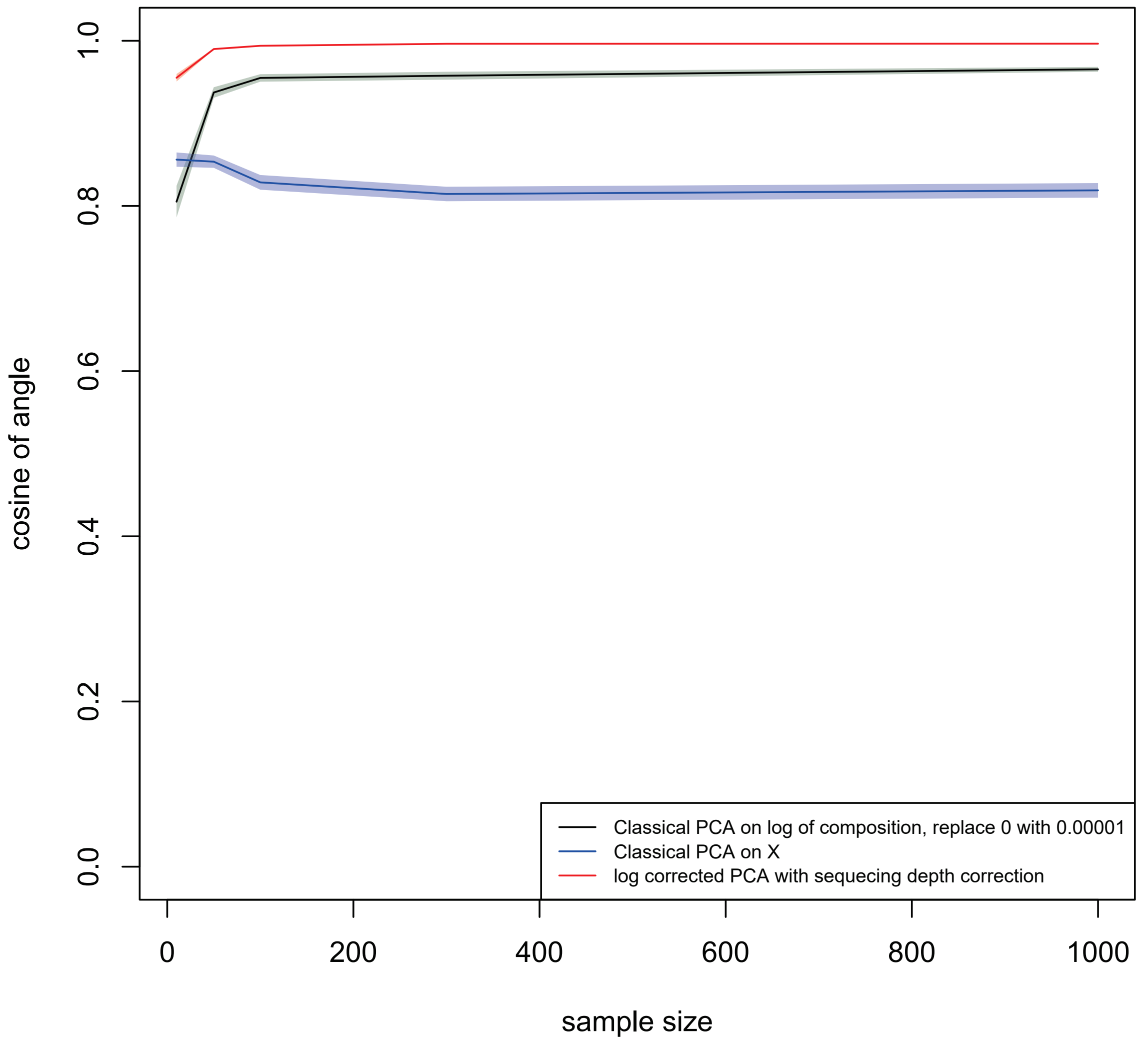


Figure 4.2: Comparison of Poisson error corrected log transformed PCA with sequencing depth correction (Method I) and Classical PCA where $\mathbf{u} \sim N(0.5, 0.1^2)$, $\mathbf{v} \sim N(1, 0.5^2)$, $\mathbf{w} \sim N(0, 2^2)$.

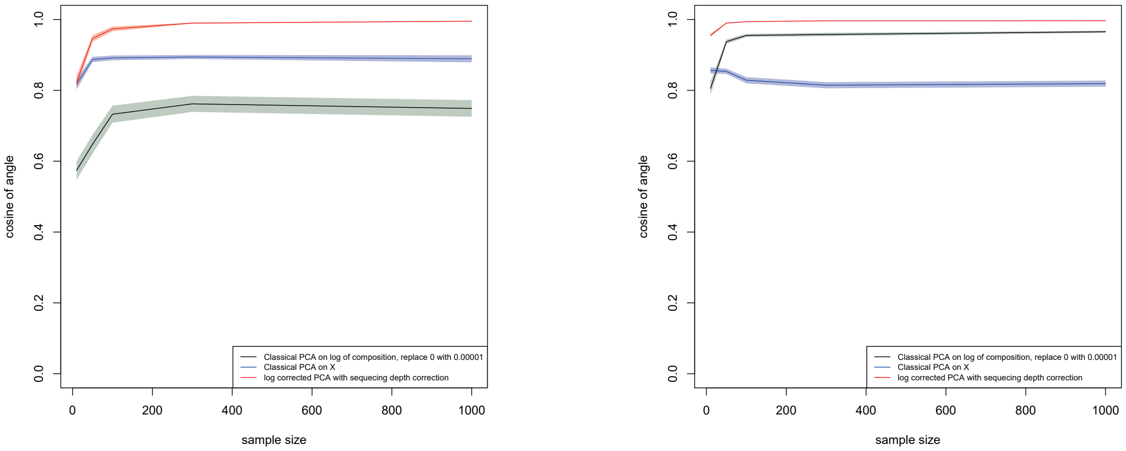


Figure 4.3: Comparison of Poisson error corrected in log transformed PCA with sequencing depth correction (Method II) and Classical PCA where $\mathbf{u} \sim \mathrm{N}(0.5, 0.1^2)$, $\mathbf{v} \sim \mathrm{N}(1, 0.5^2)$, $\mathbf{w} \sim \mathrm{N}(0, 1^2)$ and, $\mathbf{w} \sim \mathrm{N}(0, 2^2)$.

Chapter 5

Projecting $\log(\Lambda)$ onto Principal Component Space

5.1 Projection Method

Suppose we have data matrix $X_{n\times p} = (x_1, x_2, \cdots, x_n)^T$, its corresponding mean matrix $\Lambda = (\Lambda_1, \Lambda_2, \cdots, \Lambda_n)^T$, where each Λ_i $(i = 1, 2, \cdots, n)$ is a vector of dimension p. We have estimated the PC's from our estimated Variance Covariance matrix $\widehat{\Sigma_{\log(\Lambda)}}$.

We would like to project $\log\widehat{(\Lambda_i)} - \boldsymbol{\mu}$ to the first k PC's, where PC's are already given by $\widehat{\Sigma_{\log(\Lambda)}} = VDV^T$, where $V = (V_1, V_2, \cdots, V_p)$ and $D = \mathrm{diag}(d_1, d_2, \cdots, d_p)$. If $\log(\Lambda)$ were known, then the mean of $\log(\Lambda)$ would be $\mu = \frac{\log(\Lambda)^T\mathbf{1}}{n}$ and the scores of $\log(\Lambda_i)$ would be $\frac{V^T(\log(\Lambda_i)-\boldsymbol{\mu})}{|V|} = V^T(\log(\Lambda_i) - \boldsymbol{\mu})$. Thus the true projection of $\log(\Lambda_i)$ on the first k principal components is:

$$P_i(k) = \sum_{j=1}^{k} V_j({V_j}^T V_j)^{-1} {V_j}^T(\log(\Lambda_i) - \boldsymbol{\mu}) = \sum_{j=1}^{k} V_j {V_j}^T(\log(\Lambda_i) - \boldsymbol{\mu}),\ k = 1, 2, \cdots, p$$

Since Λ and thus $\log(\Lambda)$ is not observed, we will need to estimate it so that it maximizes the likelihood of the data on one hand and minimizes the difference between $\log(\Lambda_i) - \boldsymbol{\mu}$ and its projection on the first k PC's space on the other hand. We will continue to refer to the quantities $V^T(\log(\Lambda_i) - \mu)$ for our estimated Λ_i as the scores. (Note that these scores depend on k: where additional clarity is required, we will refer to them as k-scores.)

The penalized log-likelihood is given by

$$l_p(X_i; \Lambda_i, V) = \sum_{i=1}^{n} \left[X_i^T \log(\Lambda_i) + \mathbf{1}^T\Lambda_i - (\log(\Lambda_i) - \boldsymbol{\mu} - P_i(k))^T\, \Sigma^{-1} \left(\log(\Lambda_i) - \boldsymbol{\mu} - P_i(k)\right) \right] \tag{5.1}$$

where $P_i(k) = \sum_{j=1}^{k} V_j {V_j}^T(\log(\Lambda_i) - \boldsymbol{\mu})$ and $\boldsymbol{\mu}$ is the mean vector of the $\log(\Lambda)$. The idea here is that if our latent $\log(\hat{\Lambda})$ follows a normal distribution with the estimated Variance-Covariance matrix Σ, then the prior log-likelihood of Λ conditional on its projection is given by the penalty term, so the penalized log-likelihood is the Bayesian posterior likelihood. We maximize $l_p(X_i; \Lambda_i, V)$ over the vector Λ_i, assuming V has been estimated using the methods in Chapters 3 and 4.

To simplify the problem from optimizing l_p on all elements of Λ simultaneously to a much lower dimensional problem, we plug in the mean vector estimate based on

$$\mathbb{E}[\mathbb{E}[g(X)|\Lambda]] = \mathbb{E}[f(\Lambda)] = \mathbb{E}[\log(\Lambda)]$$

An estimator $\hat{\boldsymbol{\mu}}$ for $\boldsymbol{\mu}$ is given in Section 3.1. Recall that

$$g(X) = \sum_{m=0}^{\infty} m! \binom{X}{m} p_m$$

where the expansion term P_m is defined by some parameter a and truncation point n, and a weight function $m(X)$. Then $\hat{\boldsymbol{\mu}} = \frac{(g(X))^T\mathbf{1}}{n}$ is an estimate for $\boldsymbol{\mu}$, where $g(X)$ is the matrix of observed data X after the above transformation is performed elementwise.

Thus for each observation X_i, we maximize (5.2) over Λ_i

$$
\begin{aligned}
l_p(X_i, \Lambda_i, V) &= X_i^T \log(\Lambda_i) - \mathbf{1}^T\Lambda_i - (\log(\Lambda_i) - \widehat{\boldsymbol{\mu}} - P_i(k))^T \, \Sigma^{-1} \, (\log(\Lambda_i) - \widehat{\boldsymbol{\mu}} - P_i(k)) \\
&= X_i^T \log(\Lambda_i) - \mathbf{1}^T\Lambda_i - \left(\sum_{j=k+1}^{p} V_j^T(\log(\Lambda) - \widehat{\boldsymbol{\mu}})V_j \right)^T VP^{-1}V^T \left(\sum_{j=k+1}^{p} V_j^T(\log(\Lambda) - \widehat{\boldsymbol{\mu}})V_j \right) \\
&= X_i^T \log(\Lambda_i) - \mathbf{1}^T\Lambda_i - \sum_{j=k+1}^{p} \frac{\left(V_j^T(\log(\Lambda) - \widehat{\boldsymbol{\mu}})\right)^2}{d_j} \qquad (5.2)
\end{aligned}
$$

where $\log(\Lambda) - \widehat{\boldsymbol{\mu}} = \sum_{j=1}^{p} V_j^T(\log(\Lambda) - \widehat{\boldsymbol{\mu}})V_j$.

The total weighted squared scores on the last $(p - k)$ principal space is used as penalty. In the case that $\widehat{\Sigma}$ is not of full rank, say it is of rank $r < p$, then for $j > r$, we have $d_j = 0$ which forces the corresponding score to be zero. We therefore implement the solution as a (constrained) optimization with the sum running from $j = k + 1$ to r and constraints $V_j^T(\log(\Lambda) - \boldsymbol{\mu}) = 0$ for $j > r$.

When the eigenvalue d_j is small, the principal score $V_j^T(\log(\Lambda_i) - \boldsymbol{\mu})$ will naturally be very small. This forces $\log(\Lambda) - \boldsymbol{\mu}$ to stay in the space where the data have larger variances, so the variance of the estimated $\log(\hat{\Lambda}) - \boldsymbol{\mu}$ is close to our estimate for the variance of the latent Λ.

For estimating the penalized maximum likelihood of $\Lambda_i = (\Lambda_{i1}, \Lambda_{i2}, \cdots, \Lambda_{ip})$, the ith score function is:

$$
\begin{aligned}
\frac{\partial l_p(X_i, \Lambda_i, V)}{\partial \Lambda_{ij}} &= \frac{X_{ij}}{\Lambda_{ij}} - 1 - \sum_{l=k+1}^{p} \frac{2[V_l^T(\log(\Lambda_i - \widehat{\boldsymbol{\mu}}))]}{d_l} V_l(j) \frac{1}{\Lambda_{ij}} \qquad (5.3) \\
&= \frac{X_{ij}}{\Lambda_{ij}} - 1 - \sum_{l=k+1}^{p} \frac{2V_l(j)[V_l^T(log(\Lambda_i) - \widehat{\boldsymbol{\mu}}_i)]}{d_l \Lambda_{ij}} \qquad (5.4)
\end{aligned}
$$

where $V_l(j)$ is the jth element of V_l.

In order to better deal with the constraints of $V_j^T(\log(\Lambda_i) - \widehat{\boldsymbol{\mu}}) = 0$ for $j > r$ when $\text{Rank}(\widehat{\Sigma}) = r$, we re-parameterise Λ_{ij} as follows. Denote $\log(\Lambda_i) - \widehat{\boldsymbol{\mu}} = \sum_{j=1}^{r} a_{ij}V_j$, where $a_{ij} = V_j^T \, (\log(\Lambda_i) - \widehat{\boldsymbol{\mu}})$ is the jth score of $\log(\Lambda_i) - \widehat{\boldsymbol{\mu}}$, $a_{ij} = 0$ for $j > r$. Thus the penalized log-likelihood can be written as:

$$
\begin{aligned}
l_p(X_i; a_i, V) &= X_i^T \left(\widehat{\boldsymbol{\mu}} + \sum_{j=1}^{p} a_{ij} V_j \right) - \mathbf{1}^T e^{\left(\widehat{\mu} + \sum_{j=1}^{p} a_{ij} V_j\right)} \\
&\quad - \left(\sum_{j=k+1}^{p} a_{ij} V_j \right)^T \Sigma^{-1} \left(\sum_{j=k+1}^{p} a_{ij} V_j \right) \\
&= X_i^T \left(\widehat{\boldsymbol{\mu}} + \sum_{j=1}^{r} a_{ij} V_j \right) - \mathbf{1}^T e^{\left(\widehat{\mu} + \sum_{j=1}^{p} a_{ij} V_j\right)} - \sum_{j=k+1}^{r} \frac{a_{ij}^2}{d_j}
\end{aligned}
$$

The score functions relative to a_{ij} will be

$$
\begin{aligned}
\frac{\partial l_p}{\partial a_{ij}} &= \begin{cases} X_i^T V_j - V_j^T e^{\left(\widehat{\mu} + \sum a_{ij} V_j\right)}, & 1 \leqslant j \leqslant k \\ X_i^T V_j - V_j^T e^{\left(\widehat{\mu} + \sum a_{ij} V_j\right)} + \frac{2a_{ij}}{d_j}, & k < j \leqslant r \end{cases} \\
&= \begin{cases} V_j^T (X_i - \Lambda_i), & 1 \leqslant j \leqslant k \\ V_j^T (X_i - \Lambda_i) + \frac{2a_{ij}}{d_j}, & k < j \leqslant r \end{cases}
\end{aligned} \tag{5.5}
$$

In this thesis, we solve these equations to maximize l_p using the `BB` package in `R`. The R function `BBsolve` uses Barzilai-Borwein spectral methods to solve nonlinear system of equations.

Maximizing the penalized likelihood over these principal component scores for a fixed k will directly give us the principal component projection.

When sequencing depth is not informative, but treated as noise, we have developed two different methods by imposing two different sets of constraints to calculate the principal components. The composition restricted variance method will compute the principal component directions that are orthogonal to vector $\mathbf{1}$. The variance due to sequencing depth difference can be computed by projecting $\log(\Lambda_i) - \widehat{\boldsymbol{\mu}}$ to direction $\mathbf{1}$. Thus when considering projection of data onto principal component space, we only need to include the $\mathbf{1}$ vector as the first vector. The penalized log-likelihood is the same as (5.1) with $P_i(k)$ defined as $P_i(k) = \sum_{j=0}^{k} V_j^T (\log(\Lambda_i) - \widehat{\boldsymbol{\mu}}) V_j$, where $V_0 = \mathbf{1}$. We then project this MLE orthogonally onto the principal component space without the vector $\mathbf{1}$.

5.2 Simulation for projecting $\log(\Lambda)$ onto Principal Component Space

Data are simulated with $\mathbf{u} \sim \mathrm{N}(0, 0, 5^2)$, $\mathbf{v}_t \sim \mathrm{N}(0.5, 0.5^2)$, and $\mathbf{w}_t \sim \mathrm{N}(0.5, 1^2)$, where $t = (1, 2, \cdots, k)$ indicates the number of principal components we generate for the synthetic data set. We set the number of observations as 10, 50, 100, 200, 500 and 1000 respectively, and the number of variables is 10. We then simulate one Λ matrix by $\Lambda = e^{\mathbf{1u}^T + (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k)(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k)^T}$ for each sample size and we generate 100 X matrices.

The simulation follows the procedure below:

1. First generate $\Lambda = e^{\mathbf{1u}^T + (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k)(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k)^T}$

(a) $\{u_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_u and standard deviation σ_u

(b) $\{(v_t)_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_v and standard deviation σ_v

(c) $\{(w_t)_i\}_{i=1}^n$ are i.i.d. normal random variables with mean μ_w and standard deviation σ_w

(d) The Gram-Schmidt process is used to find $\mathbf{w}_{t_1} \perp \mathbf{w}_{t_2}$, where $t_1 = (1, 2, \cdots, k)$ and $t_2 = (1, 2, \cdots, k)$, $t_1 \neq t_2$.
Note that we only generate 2 Λ matrices for each sample size. The first one is $\Lambda = e^{\mathbf{1}\mathbf{u}^T + \mathbf{v}\mathbf{w}^T}$ and the second one is $\Lambda = e^{\mathbf{1}\mathbf{u}^T + (\mathbf{v}_1, \mathbf{v}_2)(\mathbf{w}_1, \mathbf{w}_2)^T}$.

2. Simulate 100 Poisson sample X matrices from Λ

3. Apply Poisson error corrected log transformed PCA on each X to get a corrected variance covariance matrix Σ_c

4. Compute $\widehat{a_{ij}}$ $(i = 1, 2, \cdots, n; j = 1, 2, \cdots, p)$ by maximizing the penalized log-likelihood (5.5), and thus get the estimated $\widehat{\Lambda}$. Note that $\hat{\boldsymbol{\mu}} = \frac{g(X)^T \mathbf{1}}{n}$.

We then use Mean Squared Error of the estimated $\log(\Lambda)$ to assess the accuracy of the penalized maximum log-likelihood estimation for the scores. We expect the estimated scores to capture the true information. That is to compute $\sum_{j=1}^p \|\widehat{\log(\Lambda_{ij}})-\log(\Lambda_{ij})\|^2$ for each row i, and find the average of the 100 simulations.

We can compare our projection method with the restricted maximum likelihood estimation (rMLE), where we restrict $\log(\Lambda)$ to lie on the first k principal component space. The restricted log-likelihood function is

$$l_r(X_i, \Lambda_i, V) = X_i^T \log(\Lambda_i) - \mathbf{1}^T \Lambda_i \tag{5.6}$$

where $\log(\Lambda_i) - \widehat{\boldsymbol{\mu}} = \sum_{j=1}^k V_j^T (\log(\Lambda_i) - \widehat{\boldsymbol{\mu}}) V_j$.

The score functions relative to a_{ij} $(j \leq k)$ can be computed by:

$$\begin{aligned} \frac{\partial l_r}{\partial a_{ij}} &= X_i^T V_j - V_j^T e^{(\widehat{\boldsymbol{\mu}} + \sum a_{ij} V_j)} \\ &= V_j^T (X_i - \Lambda_i) \end{aligned} \tag{5.7}$$

where V_j is the jth eigenvector of our estimated Poisson noise corrected variance covariance matrix and $j = 1, 2, \cdots, k$.

We compare a third method, the orthogonal projection of $\log(X)$ onto the principal component space. This can be seen as coming from our penalized likelihood function where the penalty in (5.1) becomes infinitesimal, or the restricted log-likelihood function in (5.6) where $\log(\Lambda_i) - \widehat{\boldsymbol{\mu}} = \sum_{j=1}^p V_j^T (\log(\Lambda_i) - \widehat{\boldsymbol{\mu}}) V_j$. The three methods are based on the same eigen structure, which comes from Poisson noise corrected log transformed PCA. We compute MSE of $\widehat{\log(\Lambda)}$ and compare among the three methods. Note that in our simulation, we set $k = 1$ and $r = 2$.

For two Λ matrices generated by $\Lambda = e^{\mathbf{1u}^T+\mathbf{vw}^T}$ and $\Lambda = e^{\mathbf{1u}^T+(\mathbf{v}_1,\mathbf{v}_2)(\mathbf{w}_1,\mathbf{w}_2)^T}$, we first check the performance of Poisson noise corrected log transformed model. In Table 5.1, we measure the average cosine of the angle between the estimated PC1 and the true PC1 by applying classical PCA on X, $\log(X)$, and the third column refers to our method. The second $\Lambda = e^{\mathbf{1u}^T+(\mathbf{v}_1,\mathbf{v}_2)(\mathbf{w}_1,\mathbf{w}_2)^T}$ is generated by 2 principal components, so we compare the average cosine of the angle of two estimated PCs with the truth for the three methods, which can be found in Table 5.3. The Poisson noise corrected log transformed model outperforms both classical methods for the two Λ matrices we randomly generated. Furthermore, for $\Lambda = e^{\mathbf{1u}^T+(\mathbf{v}_1,\mathbf{v}_2)(\mathbf{w}_1,\mathbf{w}_2)^T}$, the third eigenvalue of our model is much more to 0 compared with the classical PCA approaches in all the simulations. All the evidence suggests that Poisson noise corrected log transformed model better reveals the latent information. Tables 5.2 and 5.4 measure the MSE of estimated $\widehat{\log(\Lambda)}$ for two Λ matrices by applying three projection methods onto the same Poisson noise corrected PC space.

Table 5.1: Average cosine of the angle in simulations for $\Lambda = e^{\mathbf{1u}^T+\mathbf{vw}^T}$.

sample size	PC1 of X	PC1 of log(X)	PC1 of $\widehat{\Sigma}_{\log(\Lambda)}$
10	0.907419	0.603683	0.896323
50	0.908442	0.911651	0.966031
100	0.903882	0.924421	0.972581
200	0.902934	0.959206	0.97886
500	0.904976	0.976776	0.982043
1000	0.898269	0.98062	0.983057

Table 5.2: MSE of simulations for $\Lambda = e^{\mathbf{1u}^T+\mathbf{vw}^T}$.

sample size	rMLE	log(X)	penalized MLE
10	4.005166	10.49233	1.759245
50	2.957265	12.52773	1.447736
100	3.004657	11.76455	1.329299
200	3.126116	12.20281	1.332634
500	2.801252	12.26772	1.4465
1000	2.7283	12.76184	1.492862

Table 5.3: Average cosine of the angle in simulations for $\Lambda = e^{\mathbf{1}\mathbf{u}^T + (\mathbf{v}_1,\mathbf{v}_2)(\mathbf{w}_1,\mathbf{w}_2)^T}$.

sample size	PC1 of X	PC2 of X	PC1 of log(X)	PC2 of log(X)	PC1 of $\widehat{\Sigma}_{\log(\Lambda)}$	PC2 of $\widehat{\Sigma}_{\log(\Lambda)}$
10	0.605491	0.487335	0.619698	0.324089	0.757472	0.595476
50	0.921586	0.828291	0.687107	0.672328	0.945374	0.933076
100	0.924367	0.741037	0.661069	0.659361	0.959238	0.955071
200	0.92996	0.759048	0.680626	0.682302	0.971504	0.972154
500	0.933418	0.644119	0.853567	0.855021	0.979798	0.981086
1000	0.932935	0.603131	0.903078	0.90241	0.981731	0.984

Table 5.4: MSE of simulations for $\Lambda = e^{\mathbf{1}\mathbf{u}^T + (\mathbf{v}_1,\mathbf{v}_2)(\mathbf{w}_1,\mathbf{w}_2)^T}$.

sample size	rMLE	log(X)	penalized MLE
10	3.488864	11.62662	0.916661
50	1.985524	14.03555	0.726366
100	1.981952	14.11767	0.562823
200	2.130901	14.5734	0.598255
500	1.971641	14.26033	0.484051
1000	1.960451	14.23148	0.489428

Chapter 6

PCA on the Moving Picture Data

The moving picture data is a human microbiota time series that covers two individuals at four body sites over 396 timepoints [21]. In this chapter we apply our Poisson noise corrected PCA methods on the tongue data of both individuals.

We will compare the analysis results of our method with several other methods popularly used in practice on this data with several different levels, i.e. genus level, family level, order level, and class level [22]. We avoid using the species level because at species level there are many more variables than samples, and this presents problems for PCA. There are different methods developed for PCA when the number of variables is much higher than the number of observations. It is however not the focus of this thesis to deal with that aspect of PCA. The proposed method in this thesis can be further developed to suit the situation of high dimensional problems.

After removing the variables for which all observations are 0's, the total number of different OTUs (variables) at species level for the tongue data is around 2000. The sample sizes are 134 and 374 respectively for two people. The numbers of OTUs are 916, 268, 138, and 92 for genus level, family level, order level, and class level respectively.

Through a quick check of the data, it is immediately apparent that sequencing depth is much higher for one of the individuals. This is most likely the result of differences in the experimental factors, rather than interesting biological signals. Figure 6 shows the classical PCA and our Poisson error corrected log transformed PCA on all the data of two individuals on genus level. The clear separation in the 1st PC is mostly due to the sequencing depth differences between two separated groups of data. The second PC of classical PCA doesn't show any separation between the data of two individuals. Our method shows slight separation of the two individuals within the subgroup of the data with similar sequencing depths.

The full tongue data in genus level can be clustered into two groups using the K-means algorithm. The first group contains 198 observations while the number of observations for the second group is 310. From the boxplots of sequencing depths for both groups in Figure 6.2, we find that the clustering is mainly based on sequencing depth rather than any biologically interesting information.

In order to demonstrate that our method is able to pick up relevant biological signals, we sub-sample the data so that sequencing depth is not the dominant factor to separate the data of these two individuals. Through sub-sampling, we make the distributions of the sequencing depths for the two individuals the same. (Note that this is not a recommended analysis technique for this data set. Rather we are ensuring that any failure to adequately account for sequencing depth does not give our method an unfair advantage.) Note that we have retained the within sample variation in sequencing depths. Thus the sequencing depth correction is still important for our analysis. Among individual 1's data, there are a

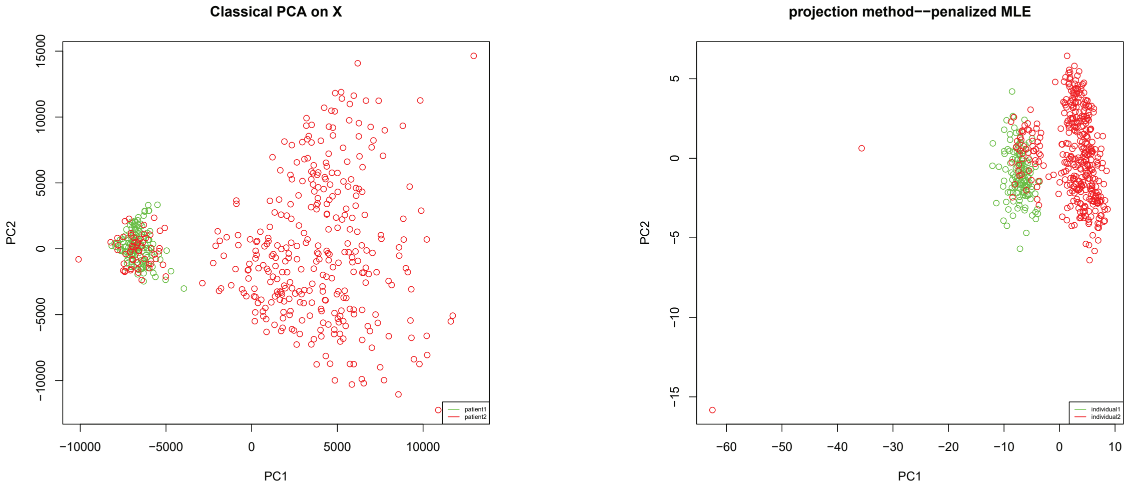


Figure 6.1: Applying classical PCA and Poisson error corrected log transformed PCA without data preprocessing.

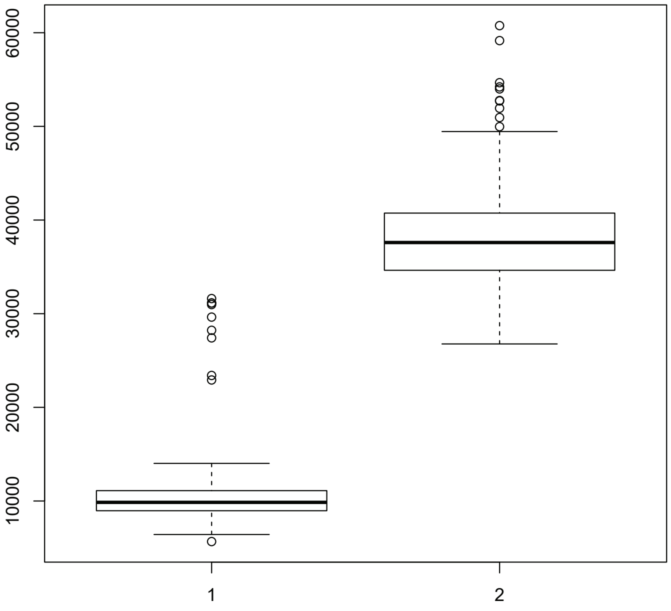


Figure 6.2: Boxplots of sequencing depths for two clusters of the data by K-means clustering on the tongue data.

large number of samples with much higher sequencing depths, thus for each sample from individual 1, we pick a sequencing depth s from a sample from individual 2, and sub-sample the corresponding individual 1's sample down to the sequencing depth s.

We apply the Poisson noise corrected log transformed PCA on the preprocessed data X to get a corrected variance covariance matrix to estimate the latent structure $\Sigma_{\log(\Lambda)}$. Then we apply the projection method, i.e. the penalized MLE method to estimate the latent Λ. The eigen-decomposition of corrected covariance matrix is $VDV^T = \sum_{j=1}^{p} d_j V_j V_j^T$, and we set r to be the number of eigenvalues greater than 10^{-2}. In practice, projecting the data onto the first two principal components is often used as the visualization method, so here we set $k = 2$. We use $\frac{\mathbf{1}^T g(X)}{n}$ as our estimate of the latent column mean $\boldsymbol{\mu}$. We plot the estimated scores on the first two corrected principal components and differentiate the two individuals with two colors.

To examine the performance of Poisson noise corrected log transformed PCA with sequencing depth correction for the tongue data, we define the projection as $P_i(k)$ and $P_i(k) = \sum_{j=0}^{k} V_j^T(\log(\Lambda_i) - \widehat{\boldsymbol{\mu}})V_j$, where $V_0 = \mathbf{1}$. Again, applying penalized MLE method with this new $P_i(k)$, we can get the estimated scores on the first two principal components V_1 and V_2. We can also project the centered X, centered $\log(X)$, and the centered compositional form of X onto their corresponding classical PCs space separately. By checking how well two groups are separated we can roughly assess the performance of each method.

Recall that PCA is not a supervised method, so optimal classification performance is not the main objective here. However, measuring the amount of latent variance explained is not straightforward. Since we have good reason to believe that the difference between the two individuals should account for a large proportion of the true latent variance, looking at the separation between the classes after applying PCA serves as a good proxy for the methods ability to remove the measurement error noise, while retaining the biologically important signal.

Figures 6.3, 6.4, 6.5 and 6.6 illustrate the performances of the Poisson noise corrected PCA and classical PCAs on different forms of the preprocessed data organized at genus level, family level, order level, and class level. For genus level and family level, log transformation without Poisson error correction (middle left) reveals the patterns of the data and differentiates the two groups significantly better than the classical PCA on X (top left). By applying our method and projecting the estimated latent scores on the Poisson noise corrected first two principal components space, we get slightly better results.

However, log transformation (left middle) without Poisson error correction does not produce as much separation between the two individuals for data at the order level and class level. Our method (left bottom) differentiates the two groups much better than classical PCAs on X, $\log(X)$, and the compositional form of X.

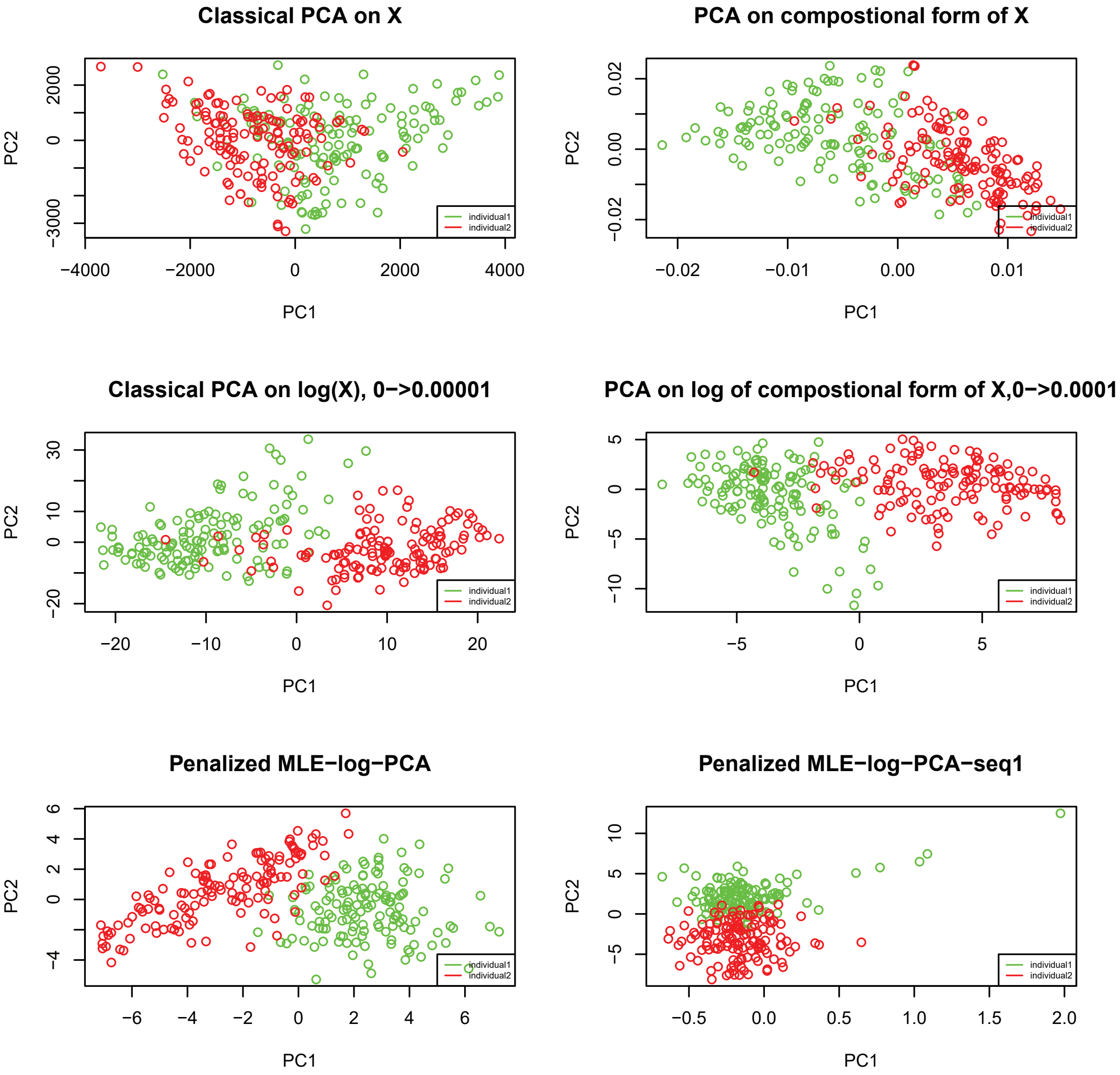
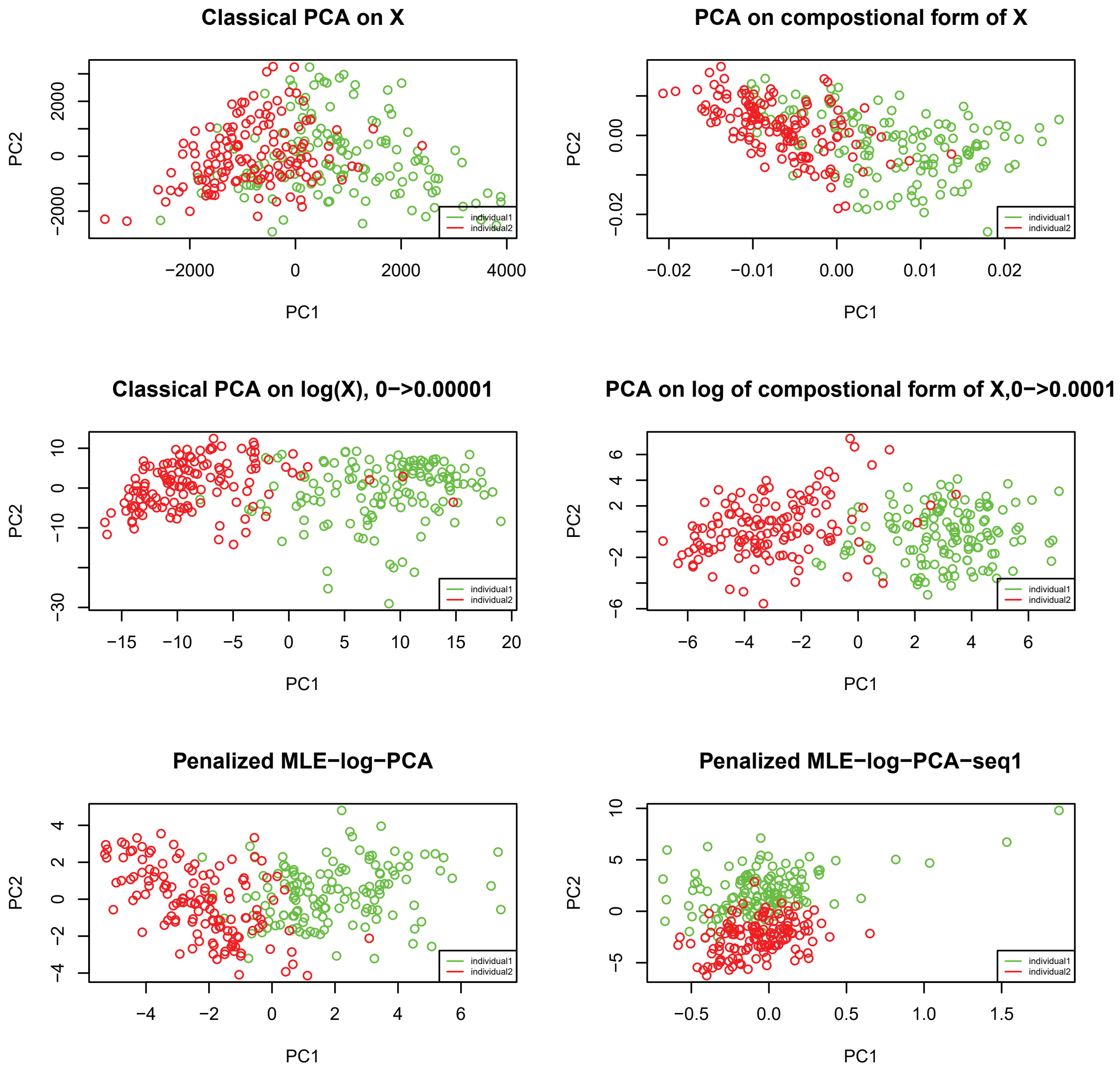


Figure 6.3: Different PCA analyses on tongue OTU data from two individuals: genus level.



Classical PCA on X

PCA on compostional form of X

Classical PCA on log(X), 0−>0.00001

PCA on log of compostional form of X,0−>0.0001

Penalized MLE−log−PCA

Penalized MLE−log−PCA−seq1

Figure 6.4: Different PCA analyses on tongue OTU data from two individuals: family level.

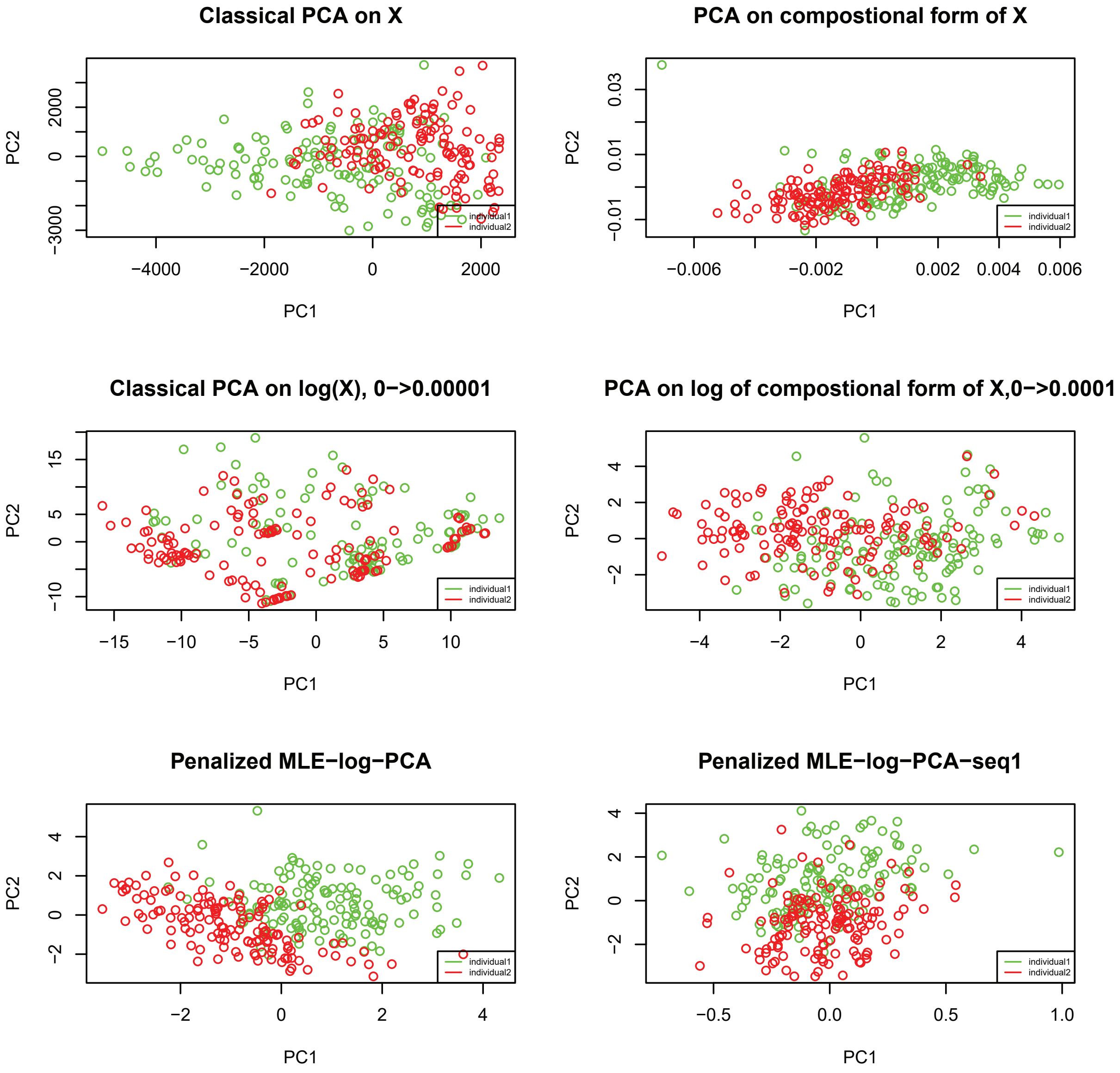


Figure 6.5: Different PCA analyses on tongue OTU data from two individuals: order level.

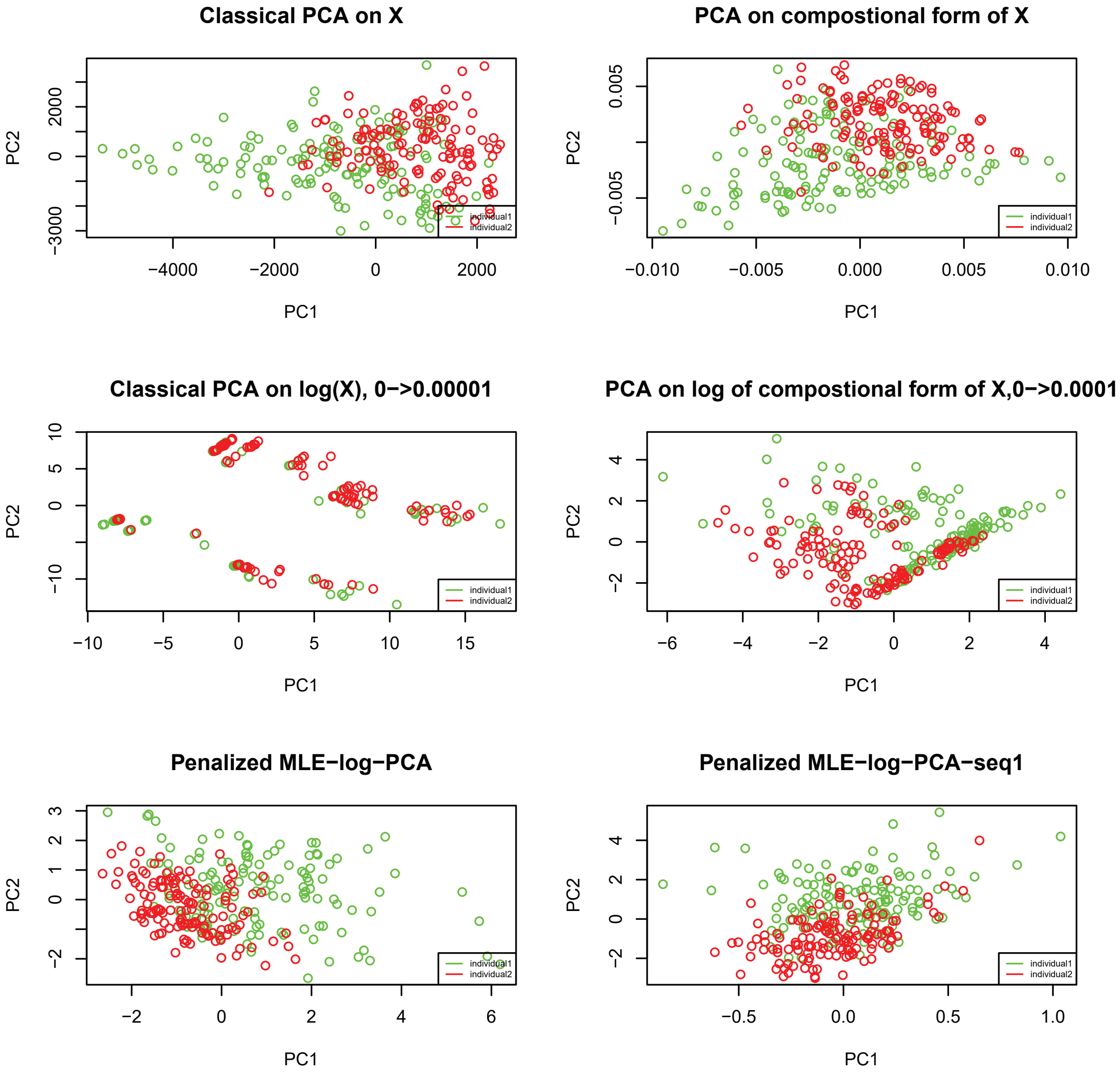


Figure 6.6: Different PCA analyses on tongue OTU data from two individuals: class level.

Chapter 7

Conclusion

In this thesis, we first proposed Poisson noise corrected PCA of the observed OTUs data to estimate the latent truth. The sequencing depth difference among the samples is further considered to be corrected in the PCA. By applying the corrected PCA on synthetic data sets and comparing the result with that of classical method, Poisson noise corrected PCA with sequencing depth correction outperforms classical PCAs both on observed data and the compositional form of data.

Due to the fact that bacteria grow exponentially, we build a logarithm transformed PCA that is to correct for the Poisson noise and find the latent truth which is on log scale. From the simulation results, we find that our method outperforms classical PCA both on observed data and on log transformation of the observed data, this is especially true when sample size is large. We then proposed two methods to correct for the sequencing depth noise for the Poisson noise corrected log transformed PCA. Our method captures the most information comparing to classical PCA on different forms of the observed data.

We further developed a penalized MLE method to find the projection of the latent data, with the assumption that the latent Poisson means lie close to the principal component space derived by our method. The mean squared error of the estimated latent Poisson means from the penalized MLE is smaller than that from rMLE or log transformation of data. We can make the conclusion that the projection method based on Poisson noise corrected log transformed PCA can find latent truth from observed data which may reduce noise to a large extent.

Bibliography

[1] Ian T Jolliffe. Principal component analysis and factor analysis. *Principal component analysis*, pages 150–166, 2002.

[2] Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[3] Neal S Holter, Madhusmita Mitra, Amos Maritan, Marek Cieplak, Jayanth R Banavar, and Nina V Fedoroff. Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proceedings of the National Academy of Sciences*, 97(15):8409–8414, 2000.

[4] G. L. Smith. Error propagation through principal components. *18th Conf. on Probability and Statistics in the Atmospheric Sciences, Atlanta, GA, Amer. Meteor. Soc., 9.3.*, 2006.

[5] Paul Gustafson. *Measurement error and misclassification in statistics and epidemiology: impacts and Bayesian adjustments*. CRC Press, 2003.

[6] Stephen Bailey. Principal component analysis with noisy and/or missing data. *Publications of the Astronomical Society of the Pacific*, 124(919):1015, 2012.

[7] Peter D Wentzell, Darren T Andrews, David C Hamilton, Klaas Faber, and Bruce R Kowalski. Maximum likelihood principal component analysis. *Journal of Chemometrics*, 11(4):339–366, 1997.

[8] Kristoffer Herland Hellton and Magne Thoresen. The impact of measurement error on principal component analysis. *Scandinavian Journal of Statistics*, 41(4):1051–1063, 2014.

[9] Joshua Lederberg and Alexa T McCray. Ome sweetomics–a genealogical treasury of words. *The Scientist*, 15(7):8–8, 2001.

[10] Luke K Ursell, Jessica L Metcalf, Laura Wegener Parfrey, and Rob Knight. Defining the human microbiome. *Nutrition reviews*, 70(suppl_1):S38–S44, 2012.

[11] Ron Sender, Shai Fuchs, and Ron Milo. Revised estimates for the number of human and bacteria cells in the body. *PLoS biology*, 14(8):e1002533, 2016.

[12] Ilseung Cho and Martin J Blaser. The human microbiome: at the interface of health and disease. *Nature reviews. Genetics*, 13(4):260, 2012.

[13] Monika A Gorzelak, Sandeep K Gill, Nishat Tasnim, Zahra Ahmadi-Vand, Michael Jay, and Deanna L Gibson. Methods for improving human gut microbiome data by reducing variability through sample processing and storage of stool. *PloS one*, 10(8):e0134802, 2015.

[14] Jane Peterson, Susan Garges, Maria Giovanni, Pamela McInnes, Lu Wang, Jeffery A Schloss, Vivien Bonazzi, Jean E McEwen, Kris A Wetterstrand, Carolyn Deal, et al. The nih human microbiome project. *Genome research*, 19(12):2317–2323, 2009.

[15] Susannah Green Tringe, Christian Von Mering, Arthur Kobayashi, Asaf A Salamov, Kevin Chen, Hwai W Chang, Mircea Podar, Jay M Short, Eric J Mathur, John C Detter, et al. Comparative metagenomics of microbial communities. *Science*, 308(5721):554–557, 2005.

[16] Hongzhe Li. Microbiome, metagenomics, and high-dimensional compositional data analysis. *Annual Review of Statistics and Its Application*, 2:73–94, 2015.

[17] Siddhartha Mandal, Will Van Treuren, Richard A White, Merete Eggesbø, Rob Knight, and Shyamal D Peddada. Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microbial ecology in health and disease*, 26(1):27663, 2015.

[18] Fan Xia, Jun Chen, Wing Kam Fung, and Hongzhe Li. A logistic normal multinomial regression model for microbiome compositional data analysis. *Biometrics*, 69(4):1053–1063, 2013.

[19] Kelvin Li, Monika Bihan, Shibu Yooseph, and Barbara A Methe. Analyses of the microbial diversity across the human microbiome. *PloS one*, 7(6):e32118, 2012.

[20] Williams Turpin, Osvaldo Espin-Garcia, Wei Xu, Mark S Silverberg, David Kevans, Michelle I Smith, David S Guttman, Anne Griffiths, Remo Panaccione, Anthony Otley, et al. Association of host genome with intestinal microbial composition in a large healthy cohort. *Nature*, 201:6, 2016.

[21] J Gregory Caporaso, Christian L Lauber, Elizabeth K Costello, Donna Berg-Lyons, Antonio Gonzalez, Jesse Stombaugh, Dan Knights, Pawel Gajer, Jacques Ravel, Noah Fierer, et al. Moving pictures of the human microbiome. *Genome biology*, 12(5):R50, 2011.

[22] Zheng-Zheng Tang, Guanhua Chen, Alexander V Alekseyenko, and Hongzhe Li. A general framework for association analysis of microbial communities on a taxonomic tree. *Bioinformatics*, 33(9):1278–1285, 2017.