Statistical Models for Single-Cell Data



Emma Pierson
St Anne's College
University of Oxford

A thesis submitted for the degree of
Master of Statistics by Research
Trinity 2015

For my grandfather, Herb Adelman, who claims this math is “black magic”
but put me through college anyway.

Acknowledgements

You can't write a thesis about hidden variables without acknowledging the hidden influences which have made it possible. Thanks first to my supervisor, Christopher Yau, who was invariably supportive, insightful, and definitely one of the best collaborators I have ever worked with. Thanks as well to the other Oxford scientists who talked through ideas with me and provided data, particularly Adam Mead, Supat Thongjuea, Justina Žurauskienė, Quin Wills, Jotun Hein, Michael Osborne, and the other members of the Yau lab. Thanks to the Rhodes Trust for more than financial support: moving to a new country is not easy, and they made it easier from the day when they collected us, sleep-starved, at the airport. To my friends at Oxford: it has been the honor of a lifetime to linger too long over meals with you. To my friends in the States: thank you for putting up with my stream-of-consciousness narrations of my research woes at 3 AM California time. To Nat: even computer scientists like being sent flowers from 5,000 miles away. *Οἱ μὲν ἱππων στρότον οἱ δὲ πέσδων, οἱ δὲ νάων φαῖσ' ἐπι γᾶν μέλαιναν, ἔμμεναι κάλλιστον ἔγω δὲ κῆν' ὅττω τὶς ἔπαται.*

To my teachers. To the men, especially Michael Stueben, John Dell, Chuong Do, Noah Goodman, Nick Eriksson, Joe Lipsick, Greg Wrenn, Greg Watkins, and David Mehler. Many of you supported me in male-dominated fields, and the more time passes the more important I realize that was. To the women, especially Christine Maginnis, Elizabeth King, Pat Burchat, and Daphne Koller. I used to think female role models weren't important; I realize now I was wrong. And to the women I may never get a chance to meet, but who inspire me to keep fighting: Suleika Jaouad, Marjorie Williams, and Marina Keegan.

Above all to my family. There aren't really any words. To my ferocious little sisters, who improve all my writing, outdo me in debate, and generally keep me on my toes. To my grandparents, who filled my childhood with books and arguments and equations. To my parents. Having spent a year studying heredity at 23andMe, I can scientifically say: you get a large fraction of the credit for whatever I am. Less scientifically: I love you.

Abstract

The growing use of single-cell gene expression data (scRNA-seq data) has made it possible to more precisely identify cellular subpopulations with distinct characteristics. This offers insight both into normal cellular function and into diseases such as cancer. However, single-cell data presents new challenges which standard clustering and dimensionality reduction methods are not designed to confront. Rather than being drawn from Gaussians as assumed by most models, single-cell datasets contain an overabundance of zeros. We use several scRNA-seq datasets to investigate how zeros are distributed in real biological data. Based on our findings, we perform a series of simulations to examine how standard clustering and dimensionality reduction methods perform on high-dimensional, zero-inflated data. We show that performance suffers as the fraction of zeros increases and that more complex non-linear models fail to alleviate this problem. Based on this, we then present two new models that account for zeros: one to perform dimensionality reduction, and one to perform clustering. For dimensionality reduction, we present an extension of the factor analysis model which accounts for zero-inflation by adding a single parameter which describes how the probability of technical zeros decreases with the expression level of a gene. We show that our model, ZIFA, outperforms factor analysis and probabilistic principal components analysis on simulated data. For clustering, we present a extension of the Gaussian mixture model which accounts for zero-inflation. We show that our model, ZIMM, outperforms k-means and Gaussian mixture models on simulated data. We also show that both ZIFA and ZIMM better fit real data from

several scRNA-seq experiments, and provide fast software implementations of both models. Lastly, we explore applications of our models to several other classes of non-biological datasets.

Contents

1 Motivation **1**

1.1 Background 1

1.2 Distribution of zeros in biological datasets 4

1.2.1 scRNA-seq in myoblasts 5

1.2.2 scRNA-seq in bone-marrow-derived dendritic cells 6

1.2.3 scRNA-seq in 11 cell populations 7

1.2.4 scRNA-seq in differentiating T cells 8

1.2.5 scRNA-seq in chronic myelogenous leukemia 9

1.3 Performance on zero-inflated data 9

1.3.1 Clustering algorithms 10

1.3.1.1 Data generation 10

1.3.1.2 Evaluation of performance 12

1.3.2 Dimensionality reduction algorithms 17

1.3.2.1 Data generation 17

1.3.2.2 Evaluation of performance 18

1.3.3 Conclusions 23

2 Zero-inflated dimensionality reduction **24**

2.1 Model 24

2.1.1 Initialization of parameters 25

2.1.2 Optimization of parameters 26

2.1.2.1 EM algorithm . . . 26
2.1.2.2 Complete data likelihood . . . 26
2.1.2.3 E-step . . . 28
2.1.2.4 M-step . . . 30
2.1.3 Computing the marginal likelihood of Y_i . . . 32
2.2 Performance in simulations . . . 34
2.2.1 Performance with perfect parameters . . . 38
2.3 Performance on biological datasets . . . 39
2.3.1 Performance metric . . . 39
2.3.2 Comparative performance . . . 40
2.4 Estimation of covariance matrices . . . 42
2.4.1 Identification of modules in biological datasets . . . 46

3 Zero-Inflated Mixture Models **49**
3.1 Model . . . 49
3.1.1 Initialization of parameters . . . 50
3.1.2 Optimization of parameters . . . 50
3.1.2.1 E-step . . . 51
3.1.2.2 M-step . . . 53
3.2 Performance in simulations . . . 54
3.3 Performance on biological datasets . . . 58

4 Broader Applicability and Future Work **59**
4.1 Transforming Y . . . 60
4.2 Application of models to other datasets . . . 61
4.2.1 Recommendation datasets . . . 62
4.2.2 Usage frequency datasets . . . 64
4.2.3 Network datasets . . . 65
4.3 Discussion and future work . . . 66

A Fast Matrix Inversions for Dimensionality Reduction **68**

B Closed Forms for Mixture Model Integrals **71**

Bibliography **73**

Chapter 1

Motivation

1.1 Background

Cells can accomplish diverse functions with identical DNA in part by controlling the quantity of RNA which is produced from each gene. For example, although cells throughout the human body have essentially the same DNA, most genes show statistically significant differences in gene expression across tissues (Schug et al., 2005), allowing tissues to perform very different functions. The growing use of high-throughput RNA sequencing (Ozsolak and Milos, 2011; Wang et al., 2009), which provides gene expression data for tens of thousands of genes, has deepened understanding of cellular function: for example, genes which show tissue-specific expression can be instrumental to tissue-specific function (Lonsdale et al., 2013; Pierson et al., 2015), and gene expression data in cancer cells predicts prognosis (van't Veer et al., 2002).

Advances in sequencing technology have made it possible to measure RNA expression levels within a single cell via single-cell RNA-sequencing (scRNA-seq) (Tang et al., 2010; Islam et al., 2011; White et al., 2011; Tischler and Surani, 2013), a technology that may "revolutionize whole-organism science" (Shapiro et al., 2013). In scRNA-seq, single cells are first isolated using techniques including micro-manipulation, fluorescence-activated cell sorting (FACS), laser-capture microdissection, or microfluidics; RNA from

single cells is then converted to cDNA and amplified using techniques including *in vitro* transcription, template switching, or homopolymer tailing (Shapiro et al., 2013; Hebenstreit, 2012). Single-cell resolution is important because combining data from multiple cells often conceals critical differences: not all cells within a tissue or a cancer express their DNA in the same way. Merely measuring the mean level of expression across a group of cells does not differentiate, for example, between uniform and bimodal distributions of gene expression, and in fact few cells express their genes at the mean level. Consequently, scRNA-seq has made it possible to more precisely identify cellular subpopulations with distinct characteristics.

This technology has numerous applications. For example, scRNA-seq has identified subgroups of T cells (Buettner et al., 2015), lung cells (Treutlein et al., 2014), and myoblasts (Trapnell et al., 2014) at different stages of differentiation. It has been used to identify subgroups of circulating tumor cells (Powell et al., 2012) and neurons (Pollen et al., 2014). It has been used to study cell-to-cell communication in bone marrow cells, identifying a core module of genes that is expressed early by a few cells, and then activated in the remaining cells by interferon-mediated signaling (Shalek et al., 2014). It has been used to study brain tumors, revealing that individual tumors contain mixtures of cell types from different types of brain cancers and identify genetic expression patterns that correlate with important functions in brain cancers, like proliferation, oncogenic signaling, and response to hypoxia (Patel et al., 2014). This diversity in cells within a tumor implies that therapeutic strategies that are effective against some cells may be confounded by others: indeed, brain tumors which are more heterogenous are are associated with worse outcomes.

These examples highlight the utility of scRNA-seq and the importance of heterogeneity within populations once treated as homogenous. But single-cell data presents new challenges which standard statistical methods are not designed to confront. One particularly large problem is that many statistical models assume that the data is Gaussian-distributed. For example, two standard statistical tasks are dimensionality reduction, in

which high-dimensional data (e.g., expression vectors for thousands of genes) is projected down into lower dimensional spaces in a way that preserves distances in the original space, and clustering, in which similar samples are grouped together. For dimensionality reduction, widely used models like probabilistic principal components analysis (PPCA) and factor analysis make Gaussian assumptions (Tipping and Bishop, 1999b). The same is true for widely used clustering models like Gaussian mixture models (GMMs) (Reynolds, 2009).

However, single-cell data is highly non-Gaussian: it contains an overabundance of zeros, in which no transcripts of a gene are detected. This may occur for both biological and technical reasons. Biological zeros occur when a gene truly is not expressed by a cell. Technical zeros occur when a gene is expressed but not detected. Most experimental methods require 5 - 10 molecules of mRNA to detect a transcript (Shapiro et al., 2013), meaning that genes expressed at lower levels are more likely to be read as zeros. In Figure 1.1, we depict read counts from 9 randomly sampled genes from the scRNA-seq data used in (Trapnell et al., 2014), making the overabundance of zeros clear. The presence of biological zeros means the zero-inflation problem is unlikely to be solved merely by the development of more sensitive detection methods.

Here, we use several scRNA-seq datasets to investigate how zeros are distributed in real biological data. Based on this, we perform a series of simulations to examine how standard clustering and dimensionality reduction methods perform on high-dimensional, zero-inflated data. We show that performance suffers as the fraction of zeros increases and that more complex non-linear models fail to alleviate this problem. We then present two new models that account for zeros: one to perform dimensionality reduction, and one to perform clustering. For dimensionality reduction, we present an extension of the factor analysis model which accounts for zero-inflation by adding a single parameter, λ, which describes how the probability of technical zeros decreases with the expression level of a gene. We show that our model outperforms factor analysis and PPCA on simulated data. For clustering, we present a extension of the GMM which accounts for

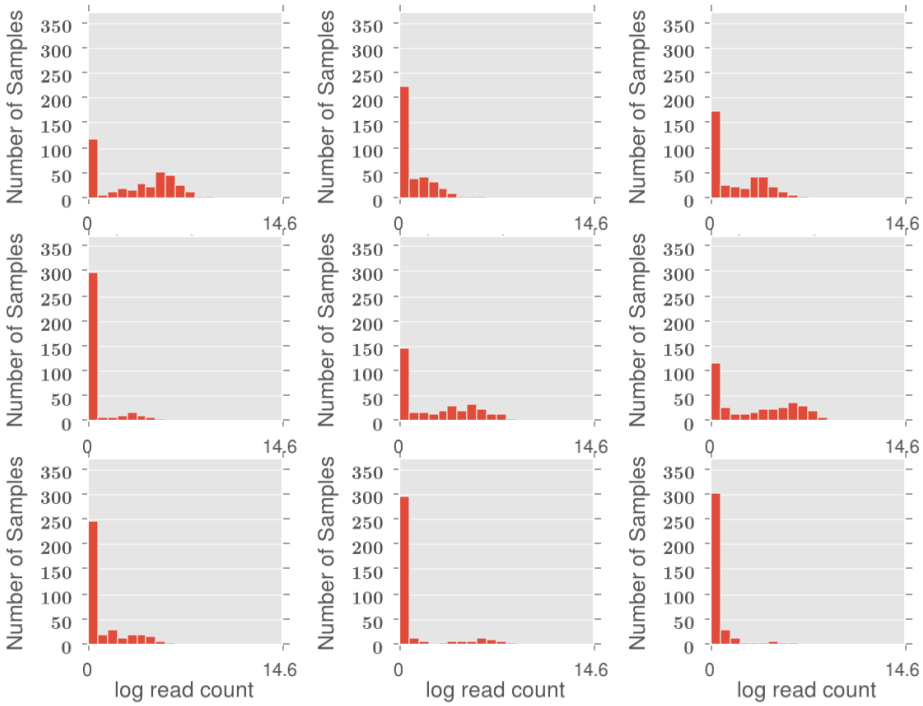


Figure 1.1: scRNA-seq data is highly non-Gaussian, containing an overabundance of zeros. Each plot is a histogram of log read counts from one randomly selected gene in a dataset of myoblasts (Trapnell et al., 2014).

zero-inflation. We show that our model outperforms K-means and GMMs on a synthetic dataset. We also show that our models better fit real data from five scRNA-seq datasets (Trapnell et al., 2014; Shalek et al., 2014; Pollen et al., 2014; Buettner et al., 2015; Mead, 2015). We explore applications of our models to several classes of non-biological datasets, and provide fast software implementations of both models.

1.2 Distribution of zeros in biological datasets

In order to build realistic simulations, we investigate the distribution of zeros on several scRNA-seq datasets. We find a negative correlation between a gene's mean log expression level μ in nonzero samples and the fraction of samples that are zero p_0, consistent with previous results (Kharchenko et al., 2014). Genes expressed at lower levels in circulating tumor cells are less likely to be detected (Ramsköld et al., 2012); genes expressed in lymphoblastoid cell lines at $>$ 100 FKPM in 10 ng RNA samples are almost always

detected, but only 30% of genes expressed at 10 FKPM are detected, and only 10% of genes expressed at 1 FKPM are detected. We investigate the relationship between μ and p_0 in several datasets, and find that $p_0 = \exp(-\lambda\mu^2)$ fits the data well. (We choose the value of λ by using non-linear least squares and minimizing the sum of squared errors.) We note that there are many other functional forms which could relate p_0 and μ; for example, one might consider a logistic or polynomial function. $p_0 = \exp(-\lambda\mu^2)$ does not fit the data perfectly: for example, in several datasets, the fitted curve is too large for small μ and too small for large μ. However, the major advantage of $p_0 = \exp(-\lambda\mu^2)$ is that it is analytically tractable for the models which we will develop in Chapters 2 and 3, while logistic and polynomial functions are not. Future work might also consider functions of the form $p_0 = \exp(-a\mu^2 - b\mu)$, which would still be analytically tractable and might fit the data better.

1.2.1 scRNA-seq in myoblasts

We first investigate the relationship between μ and p_0 using data from (Trapnell et al., 2014) (Figure 1.2) using all genes that are expressed in at least 5% of samples. Data in (Trapnell et al., 2014) comes from scRNA-seq in myoblasts (precursors of muscle cells) captured at four time points as they differentiate into muscle cells. The data contains multiple groups of cells: proliferating cells, differentiating myoblasts, and interstitial mesenchymal cells. We find a negative relationship between μ and p_0 that is well-modeled by exponentially decaying fits, especially $p_0 = \exp(-\lambda\mu^2)$.

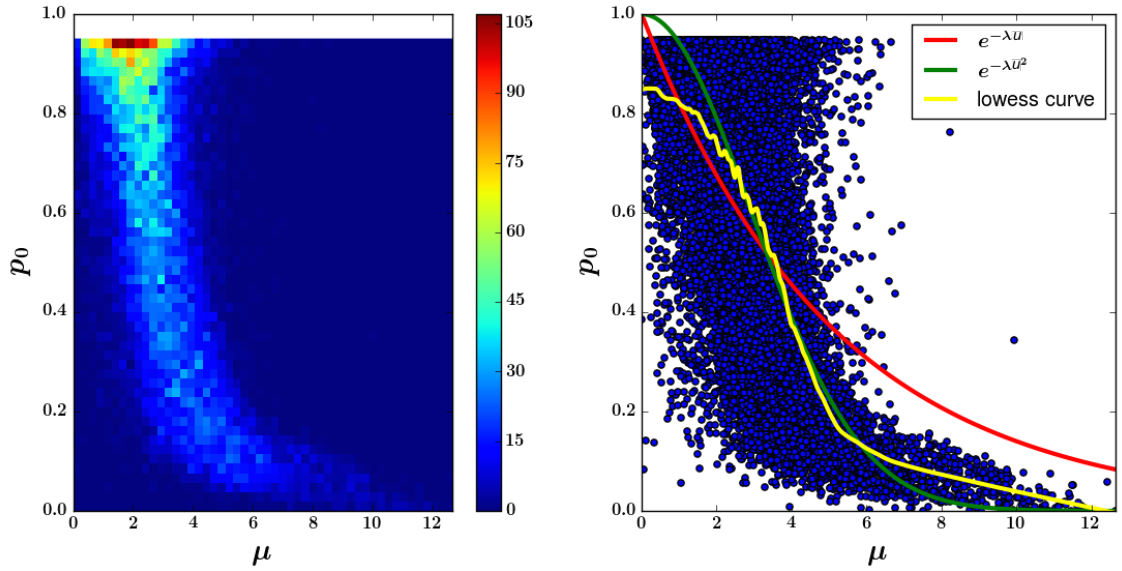


Figure 1.2: Correlation between μ and p_0 in myoblasts. The density map at left indicates the number of genes in each μ, p_0 bucket; the scatter plot at right shows the lowess curve, along with the $e^{-\lambda x}$ and $e^{-\lambda x^2}$ fits.

1.2.2 scRNA-seq in bone-marrow-derived dendritic cells

We next investigate the relationship between μ and p_0 using data from (Shalek et al., 2014) (Figure 1.3) using all genes that are expressed in at least 5% of samples. Data in (Shalek et al., 2014) comes from scRNA-seq in 1,700 mouse bone-marrow-derived dendritic cells, captured at 5 time points after stimulation with three different pathogens. Again, there is a negative relationship between μ and p_0 that is well-modeled by $p_0 = \exp(-\lambda\mu^2)$.

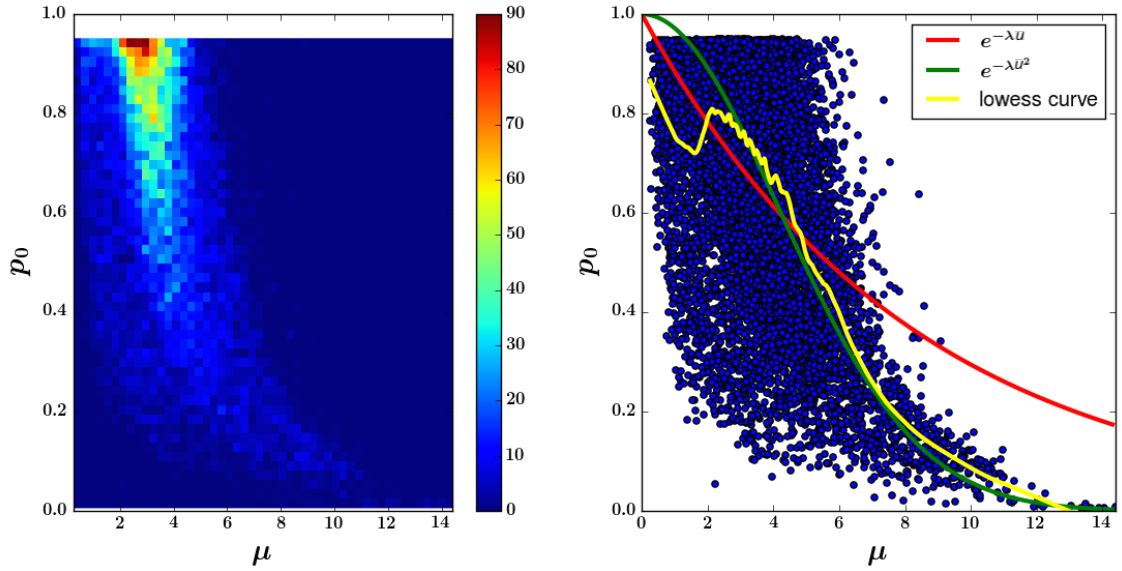


Figure 1.3: Correlation between μ and p_0 in bone-marrow-derived dendritic cells.

1.2.3 scRNA-seq in 11 cell populations

We find a similar relationship between μ and p_0 using data from (Pollen et al., 2014) (Figure 1.4), which comes from 11 different cell populations including neural cells and blood cells. This indicates that the relationship between μ and p_0 persists even in datasets with many cell populations with diverse expression patterns.

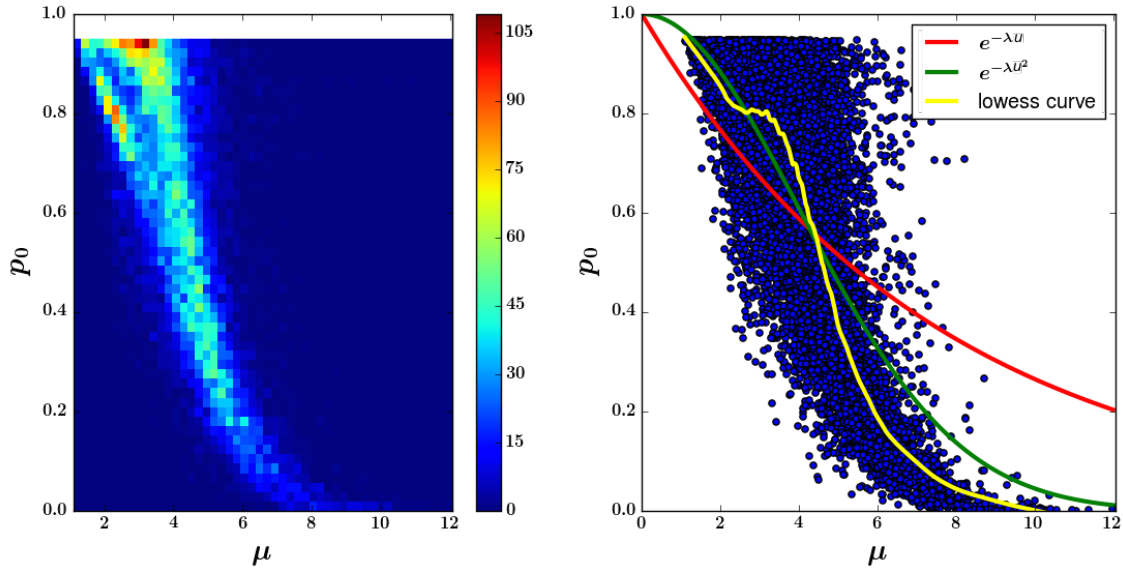


Figure 1.4: Correlation between μ and p_0 in 11 cell populations that include neural cells and blood cells.

1.2.4 scRNA-seq in differentiating T cells

We find a similar relationship between μ and p_0 using data from (Buettner et al., 2015) (Figure 1.5), which comes from differentiating T cells.

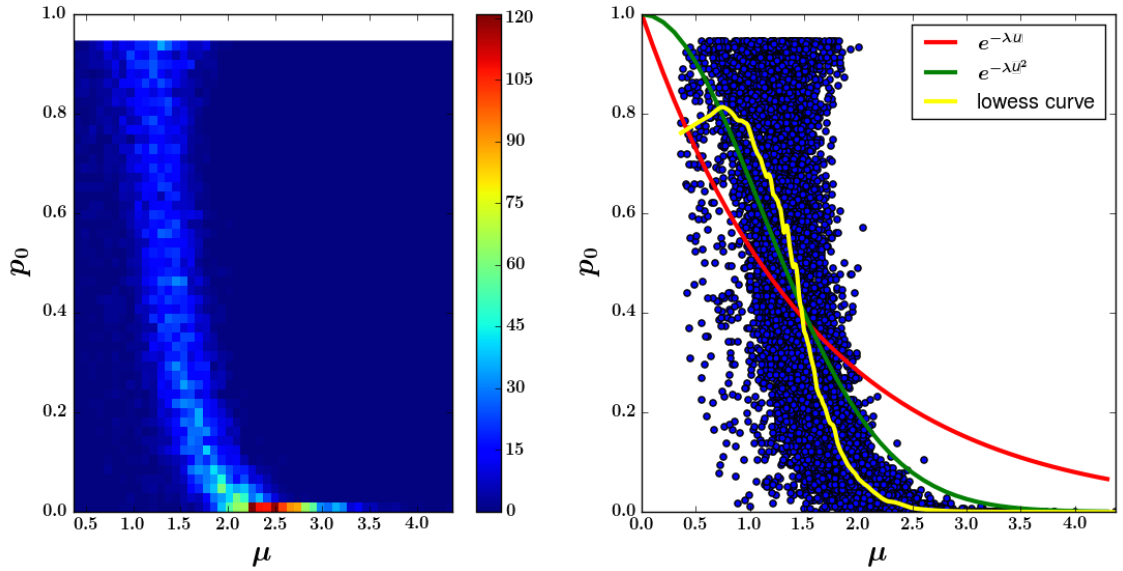


Figure 1.5: Correlation between μ and p_0 in differentiating T cells.

1.2.5 scRNA-seq in chronic myelogenous leukemia

Some datasets display more complex patterns. scRNA-seq data from blood cells in leukemia patients (Mead, 2015) provides one revealing example; we thank Adam Mead for providing this preliminary data. The dataset contains two groups of genes: one which is consistently expressed at very low expression levels, one which is less consistently expressed but at higher expression levels. The former group is enriched for membrane proteins. The two-cluster effect may occur as an artifact of how the cells are selected: they are sorted based on membrane proteins including CD90 and CD34. This example highlights the fact that the relationship between μ and p_0 is often a decaying squared exponential, it may differ depending on the nature of the biological experiment, and should be confirmed on a case-by-case basis.

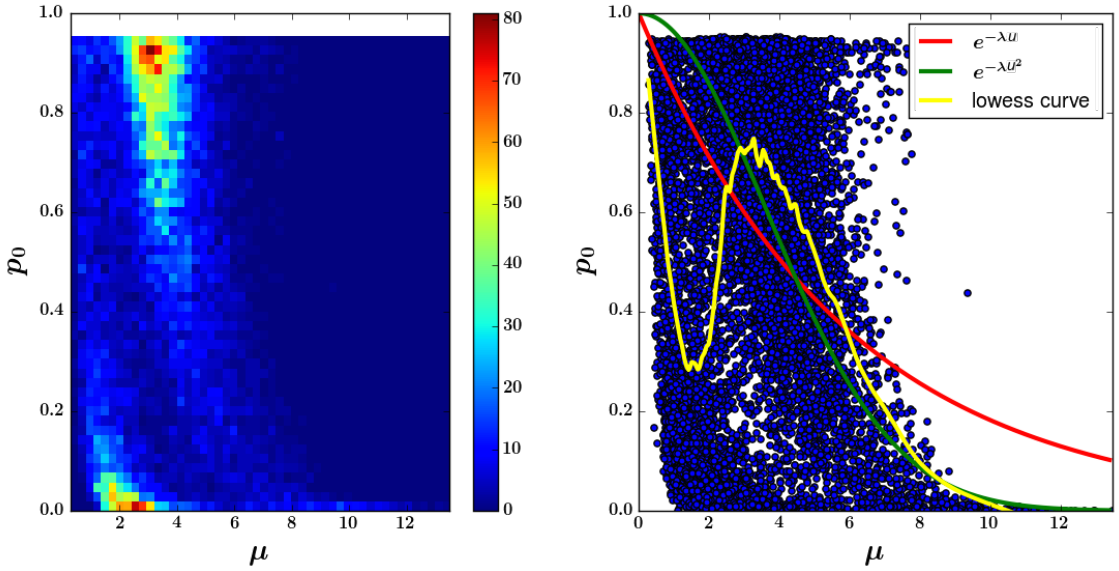


Figure 1.6: Correlation between μ and p_0 in the CML dataset.

1.3 Performance on zero-inflated data

We analyzed the performance of standard clustering and dimensionality reduction algorithms on simulated zero-inflated data. Based on our finding that $p_0 = \exp(-\lambda\mu^2)$

provided a reasonable model for zero-inflation on biological datasets, we created simulated data to which zeros were added with probability that decayed exponentially with the level of gene expression.

1.3.1 Clustering algorithms

We evaluated the performance of two standard clustering models: K-means (Hartigan and Wong, 1979) and GMMs (Reynolds, 2009) on zero-inflated data.

1.3.1.1 Data generation

We drew each sample from a mixture of Gaussians and then added zeros to the data. We index samples by $i = 1, ..., N$, genes by $j = 1, ..., D$, and clusters by $k = 1, ..., K$. For each sample i, we drew a cluster Z_i from a multinomial distribution parameterized by ϕ. For each gene j in the sample, we drew an expression value X_{ij} from a Gaussian parameterized by mean μ_{jk} and standard deviation σ_{jk}. (Throughout this thesis, we use the notation $X \sim \mathcal{N}(\mu, \sigma^2)$ to indicate that X is drawn from a normal distribution with mean μ and variance σ^2; we use $N(x; \mu, \sigma^2)$ to denote the value of the probability density function at x.) We then set the sample to zero with probability $f(X_{ij})$, where f was a non-increasing function of X_{ij}, so samples with larger X_{ij} had lower probabilities of being set to zero. The graphical model for the data generation process is shown in Figure 1.7.

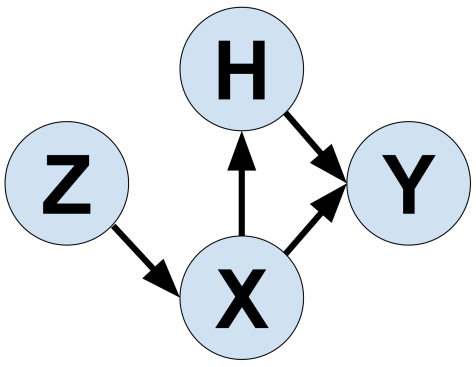


Figure 1.7: Graphical model for the generative process: Z generates X, which generates H; Y depends on X and H.

The full model for the observed data Y was as follows:

$$Z_i \sim Multinomial(\phi),$$

$$X_{ij}|(Z_i = k) \sim \mathcal{N}(\mu_{jk}, \sigma^2_{jk}),$$

$$H_{ij}|X_{ij} \sim Bernoulli(f(X_{ij})),$$

$$Y_{ij} = \begin{cases} \max(X_{ij}, 0) & \text{if } H_{ij} = 0 \\ 0 & \text{if } H_{ij} = 1 \end{cases}$$

where σ_{jk} for each cluster were drawn from a uniform distribution $U(0.9, 1.1)\sigma$, where σ was a simulation parameter; μ_{jk} for each cluster were drawn from $U(2.5, 3.5)$; and ϕ_k were drawn from $U(0, 1)$ and normalized to sum to 1. We experimented with three choices of $f(X_{ij})$: decaying squared exponential, $f(X_{ij}) = \exp(-\lambda X^2_{ij})$; uniform, $f(X_{ij}) = 1 - \lambda$; and linear, $f(X_{ij}) = 1 - \lambda X_{ij}$. (In all cases, if f was greater than 1, we set it to 1; if f was less than 0, we set it to 0.) We used a base setting of $N = 150, K = 2, D = 10, \sigma = 0.3, \lambda = 0.3$ and explored the effects of altering the decay parameter λ, the number of clusters[1] K, the cluster spread σ, the number of dimensions D, and the number of samples N. We evaluated clustering algorithms using the true number of clusters in the simulated data. In analysis of real biological data, the true number of clusters might be unknown; in such a setting, the number of clusters could be chosen using a Dirichlet process mixture model (Dahl, 2006), or heuristics like the elbow method or BIC score (Chen and Gopalakrishnan, 1998).

We discuss two nuances of this model because models of this form reappear throughout this thesis, and are the basis for the two new algorithms we develop in Chapters 2 and 3. First, is it reasonable to model Y as noiselessly observed from X if H is zero? While we could add an additional noise parameter σ_Y (for example, $Y_{ij} \sim \mathcal{N}(X_{ij}, \sigma^2_Y)$ if $H_{ij} = 0$) we do not do so for two reasons. First, this would fail to capture the intuition behind

[1] We note that while our base setting of $K = 2$ is a small number of clusters, many biological datasets can be thought of as having two clusters: for example, normal cells and cancer cells.

our model: that gene expression X is correctly observed unless it is zeroed out. X already contains a measurement noise term so adding another noise process on top seems redundant: there is no physical justification. Second, it would make parameter estimation for the model much more difficult: as we will show in our later derivations, the analytic tractability of the model is dependent on the fact that if Y is non-zero, X is known.

Second, in our simulated data, Y is never negative, because we set negative gene expression values to zero. This is biologically realistic, since gene expression cannot be negative, and makes sense for generating simulated data. However, when we try to fit similar models to actual data (Chapters 2 and 3) we find that assuming that negative values are truncated yields expressions which are not analytically tractable. We discuss our solutions to this problem in Chapters 2 and 3, but also note that models which can generate negative values are often used to model non-negative data. This is not generally a problem as long as the model does not place too much mass on negative values, and we find that is true in our data as well.

1.3.1.2 Evaluation of performance

All clustering algorithms yielded a clustering $\hat{z}$ that was compared to the ground truth clustering z using the adjusted Rand index (Hubert and Arabie, 1985; Rand, 1971). The Rand index of two clusterings $z, \hat{z}$ is:

$$R(z, \hat{z}) = \frac{n_{\text{same}} + n_{\text{different}}}{n_{\text{pairs}}}$$

where n_{same} is the number of pairs of points that are assigned to the same cluster in z and $\hat{z}$, $n_{\text{different}}$ is the number of pairs of points that are assigned to different clusters in z and $\hat{z}$, and n_{pairs} is the number of pairs of points. This captures the idea that clusterings A and B are similar if a pair of points is that assigned to the same cluster in A tends to be assigned to the same cluster in B. The adjusted Rand index compares this score to the score for a random partition, yielding a value between -1 and 1, with higher scores indicating better clusterings and zero indicating performance no better than chance.

We report the results of all simulations in Figures 1.8-1.10. We evaluated both how the algorithms performed on the original data and how they performed on resampled data in which we set each Y_{ij} that was equal to zero to a randomly chosen nonzero value from the jth column of Y. (We did this because the spike at zero often caused the clustering algorithms to perform very badly by centering one cluster at zero otherwise.) For each parameter setting, we report the average performance over 5 random simulations.

We drew several conclusions from our simulations. The performance of the algorithms improved as the number of datapoints increased (because there was more data with which to estimate the same number of parameters) and the number of dimensions increased (because there were more dimensions to separate clusters). Performance worsened as the number of clusters increased (because there were more parameters to estimate with the same amount of data), as λ decreased and the fraction of zeros increased (because the data deviated more dramatically from the model assumptions), and as σ increased (because clusters were more likely to overlap). We observed similar trends regardless of whether we added zeros exponentially, linearly, or uniformly. Performance showed the strongest dependence upon λ and σ. For sufficiently few zeros (e.g., $\lambda = 0.4$ in the linear simulation) the data became essentially Gaussian and both algorithms performed well. For sufficiently large σ, both algorithms performed badly because it was impossible to discriminate between clusters. There were no large performance differences between the two algorithms. Both performed badly on the raw data; imputing zeros dramatically improved performance.

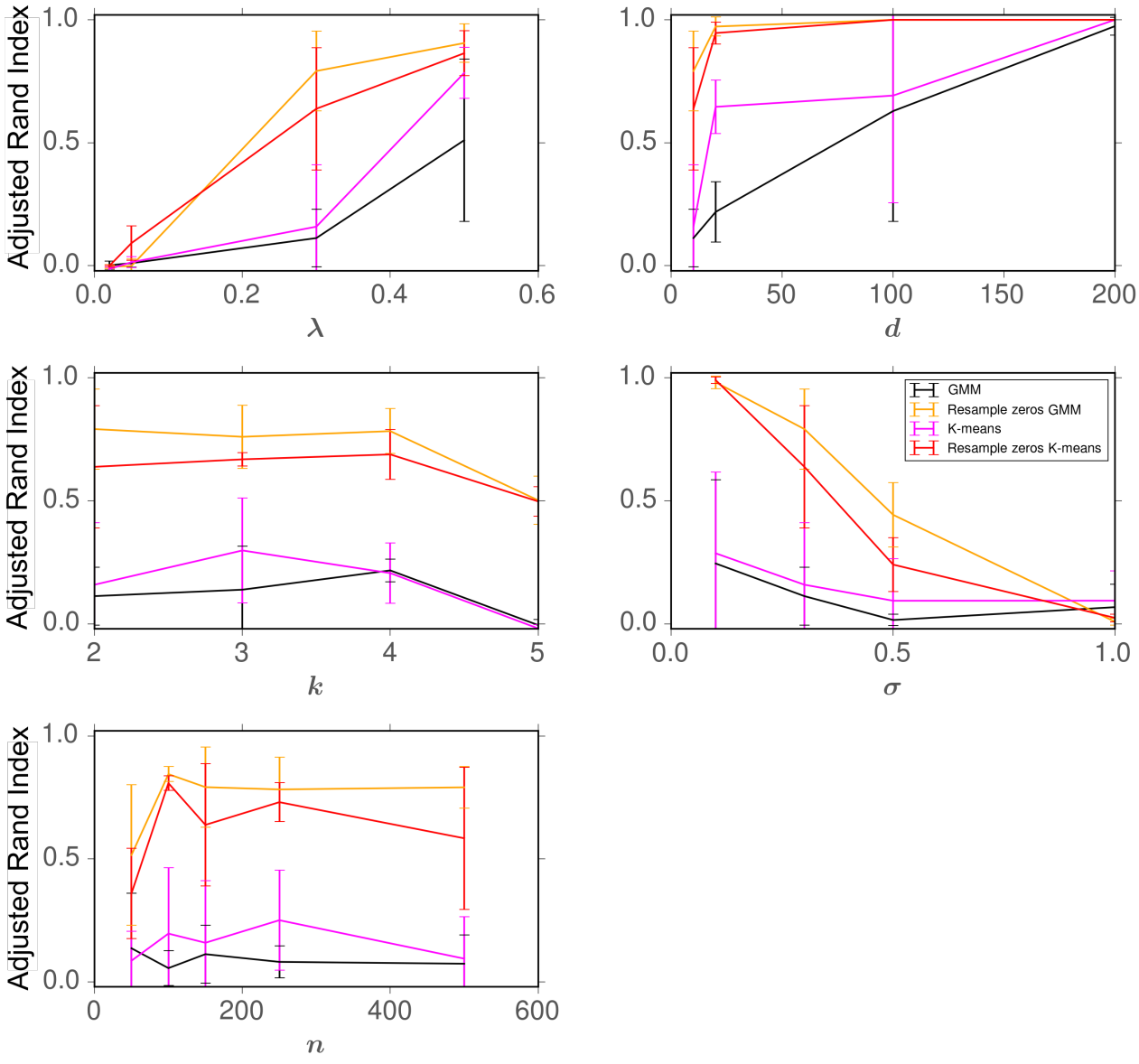


Figure 1.8: Performance of standard clustering algorithms on simulated data that is zeroed out with exponentially decaying probability $\exp(-\lambda X^2)$.

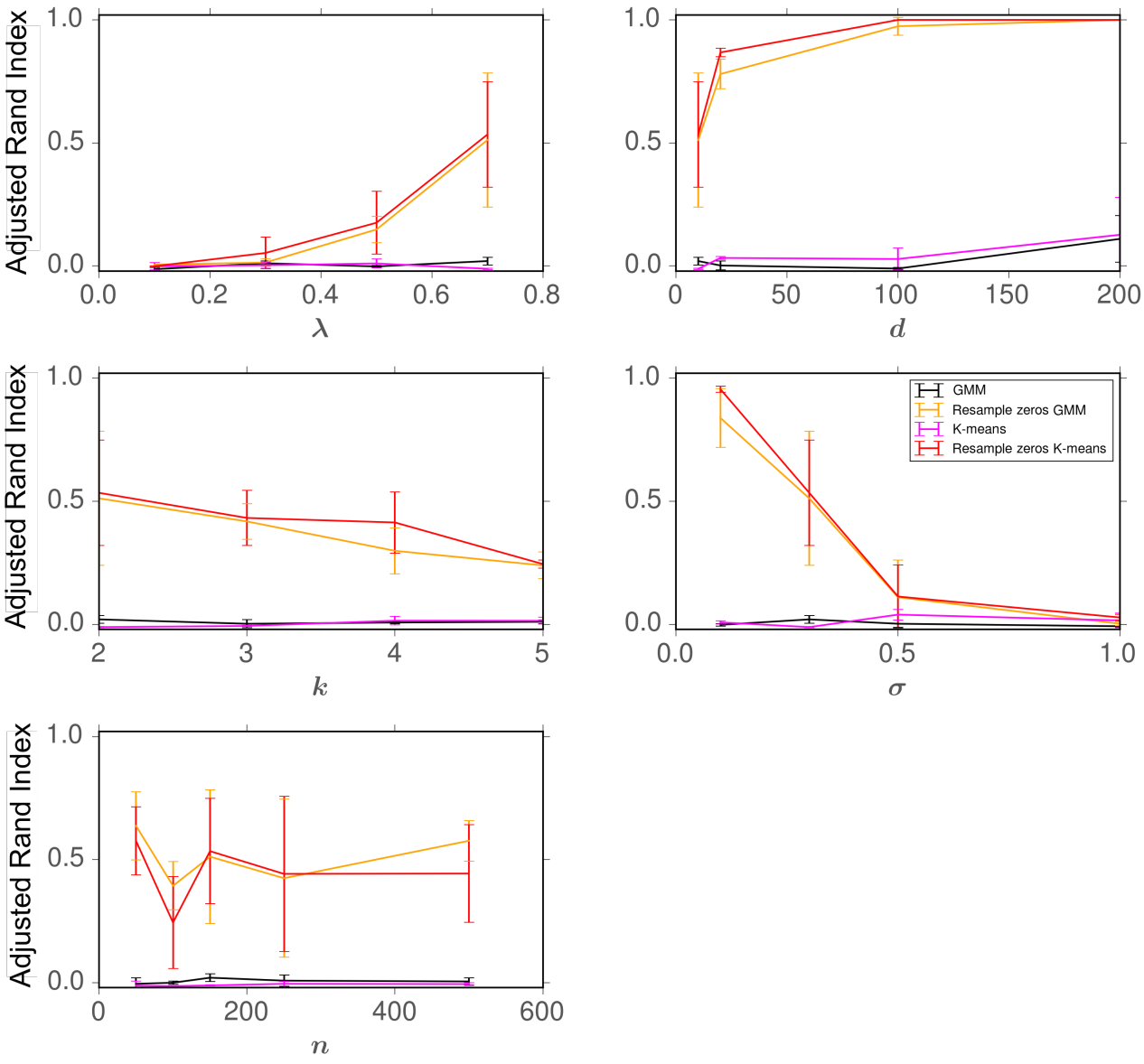


Figure 1.9: Performance of standard clustering algorithms on simulated data that is zeroed out with uniform probability $1 - \lambda$.

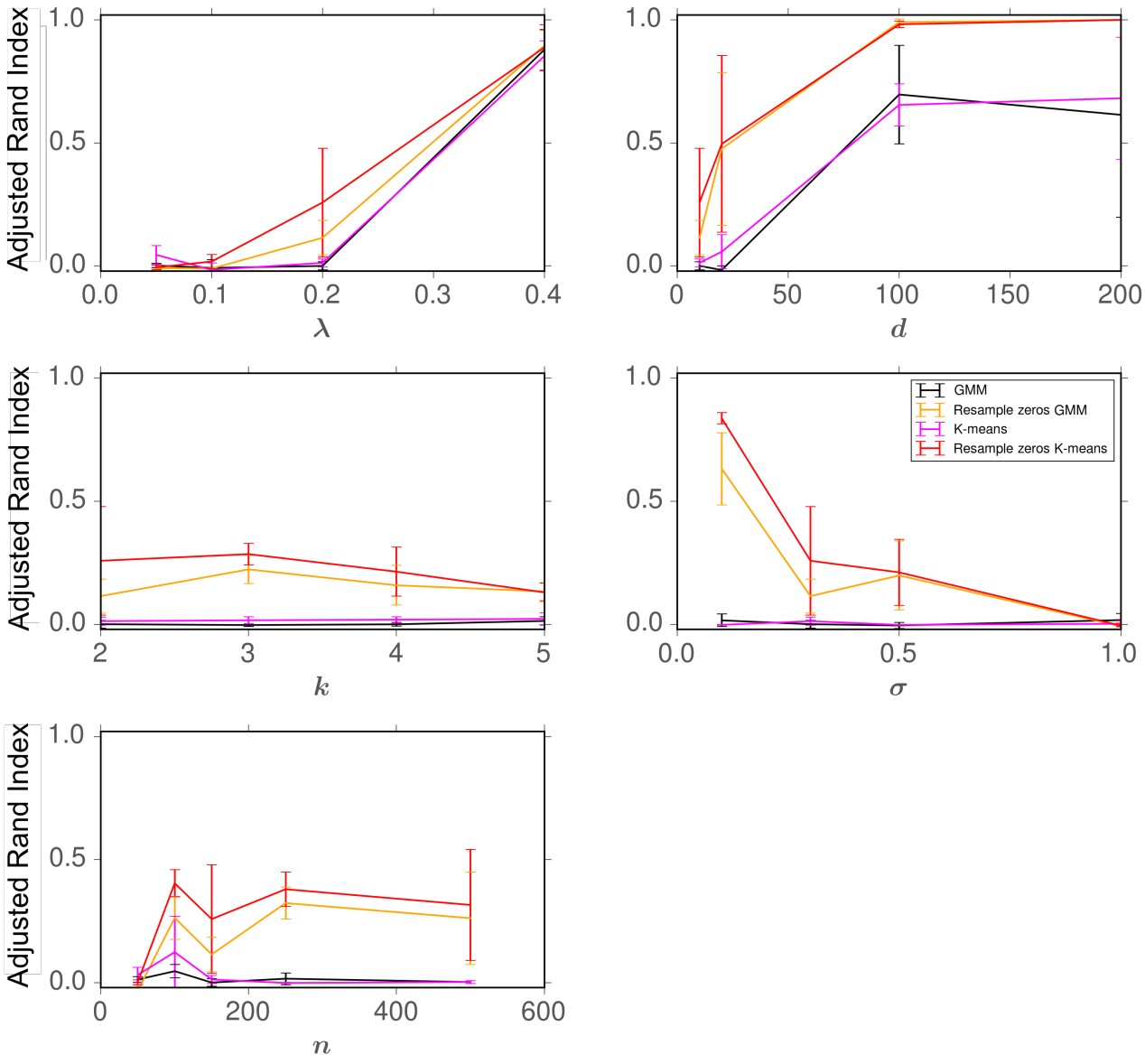


Figure 1.10: Performance of standard clustering algorithms on simulated data that is zeroed out with linearly decaying probability $1 - \lambda X$.

1.3.2 Dimensionality reduction algorithms

We evaluated the performance of several standard dimensionality reduction methods on zero-inflated data: principal components analysis (PCA) and probabilistic principal components analysis (PPCA) (Tipping and Bishop, 1999b); factor analysis (Tipping and Bishop, 1999b); multi-dimensional scaling (MDS) (Kruskal, 1964); isomap (Tenenbaum et al., 2000); and t-SNE (Van der Maaten and Hinton, 2008). We also evaluated the performance of PPCA and factor analysis on data in which zeros had been resampled as described above.

1.3.2.1 Data generation

As with our clustering simulations, we first generated data using a standard generative model (in this case, the factor analysis model) and then added zeros. We index samples by $i = 1, ..., N$, genes by $j = 1, ..., D$, and latent dimensions by $k = 1, ..., K$. We drew each sample from a low-dimensional space then added zeros to the data. For each sample i, we drew a latent $K \times 1$ vector Z_i from a Gaussian with zero mean and identity variance. We then drew a gene expression vector X_i from a Gaussian with mean $AZ_i + \mu$ and covariance W, where A was a $D \times K$ matrix, W was a $D \times D$ diagonal matrix and μ was a $D \times 1$ vector. We then set the sample to zero with probability $f(X_{ij})$, where f was a non-increasing function of X_{ij}, so samples with larger X_{ij} had lower probabilities of being set to zero. The graphical model for the data generation process is identical to that used for clustering (Figure 1.7) except that Z is now a low-dimensional projection rather than a clustering assignment. The full model for the observed data Y was as follows:

$$Z_i \sim \mathcal{N}(0, I),$$

$$X_i | Z_i \sim \mathcal{N}(AZ_i + \mu, W),$$

$$H_{ij} | X_{ij} \sim Bernoulli(f(X_{ij})),$$

$$Y_{ij} = \begin{cases} \max(X_{ij}, 0) & \text{if } H_{ij} = 0 \\ 0 & \text{if } H_{ij} = 1 \end{cases}$$

where A_{jk} was drawn from a uniform distribution $U(-0.5, 0.5)$, μ_j were drawn from $U(2.7, 3.3)$, and $W_{jj} = \sigma_j^2$, with σ_j drawn from the uniform distribution $U(0.9, 1.1)\sigma$, where σ is a simulation parameter. We experimented with three choices of $f(X_{ij})$: decaying squared exponential, $f(X_{ij}) = \exp(-\lambda X_{ij}^2)$; uniform, $f(X_{ij}) = 1 - \lambda$; linear, $f(X_{ij}) = 1 - \lambda X_{ij}$ We used a base setting of $N = 150, K = 10, D = 50, \sigma = 0.3, \lambda = 0.1$ and explored the effects of altering the decay parameter λ, the number of latent dimensions K, the cluster spread σ, the number of dimensions D, and the number of samples N. Because some biological datasets contain multiple clusters (e.g., tissue types or diseased and normal cells), we also ran simulations that contained multiple clusters in which each sample came from a cluster c_i and Z_i was drawn from $N(\mu_{c_i}, I)$.

1.3.2.2 Evaluation of performance

As a measure of algorithm performance, we compared the true Z to the $\hat{Z}$ estimated by the algorithms as follows. We computed the true distance between each pair of points i_1, i_2 and defined a pairwise distance matrix D such that $D_{i_1 i_2} = ||Z_{i_1} - Z_{i_2}||_2$. We compared this to the estimated distance matrix $\hat{D}$ with $\hat{D}_{i_1 i_2} = ||\hat{Z}_{i_1} - \hat{Z}_{i_2}||_2$. We scored the correspondence between the two distance matrices using Spearman correlation ρ_s. By comparing D and $\hat{D}$ rather than Z and $\hat{Z}$, we account for the fact that dimensionality reduction algorithms may rotate the points but ought to preserve the relative distances between them.

We report the results of all simulations in Figures 1.11 - 1.13. Based on our simulations, we drew several conclusions. Nonlinear dimensionality reduction algorithms were generally outperformed by simpler, more widely used algorithms like PPCA and factor analysis. (Because the nonlinear dimensionality reduction algorithms did not produce clear improvements on our data and are not as widely used as PPCA and factor analysis,

we dropped them from further consideration in our analysis.) As with clustering, the performance of the algorithms improved as N and D increased and worsened as the fraction of zeros increased, σ increased, and the number of clusters increased. Performance also worsened as K increased, because the algorithms had to infer more parameters with the same amount of information, increasing noise.

Resampling zeros did not yield the dramatic performance boosts for standard dimensionality reduction algorithms that it did for clustering: in fact, it yielded inconsistent gains. When zeros were added uniformly, resampling zeros improved performance (because the zeros were uninformative and the algorithms are not designed to accommodate zero-inflated data). But when the zero probability was dependent on the original value, resampling zeros worsened performance because it ignored the information in the zeros.

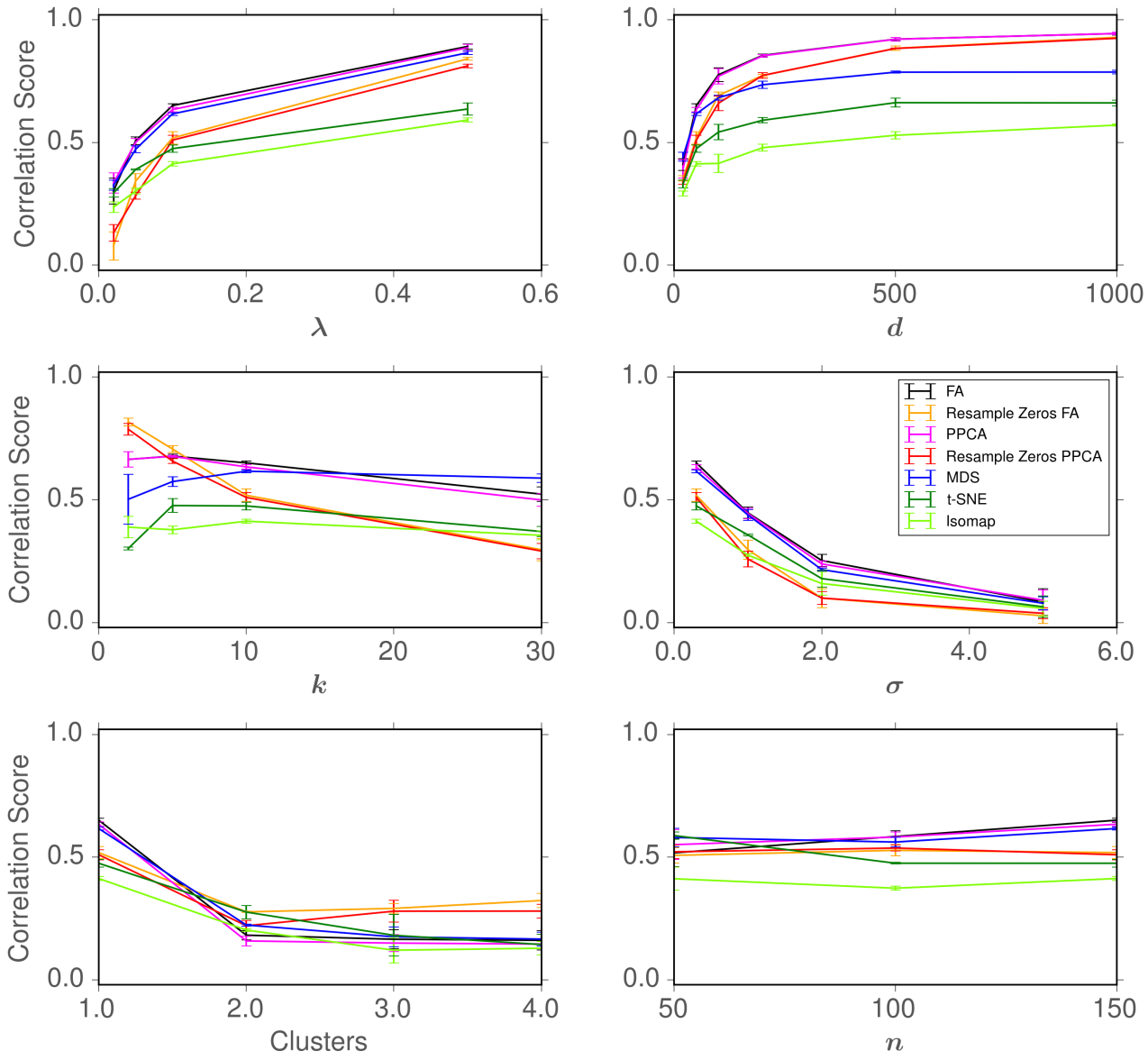


Figure 1.11: Performance of standard dimensionality reduction algorithms on simulated data that is zeroed out with exponentially decaying probability $\exp(-\lambda X^2)$.

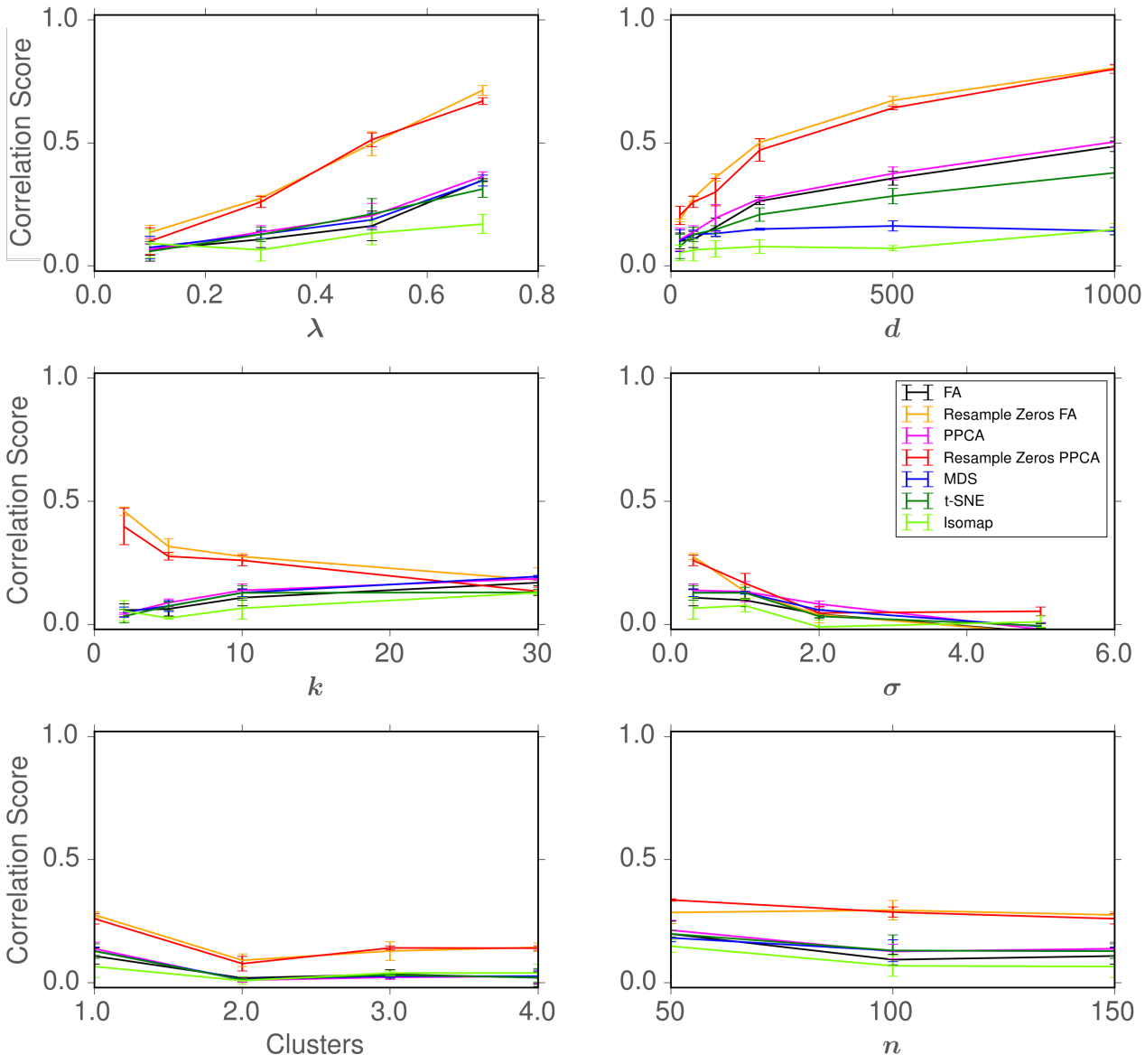


Figure 1.12: Performance of standard dimensionality reduction algorithms on simulated data that is zeroed out with uniform probability $1 - \lambda$.

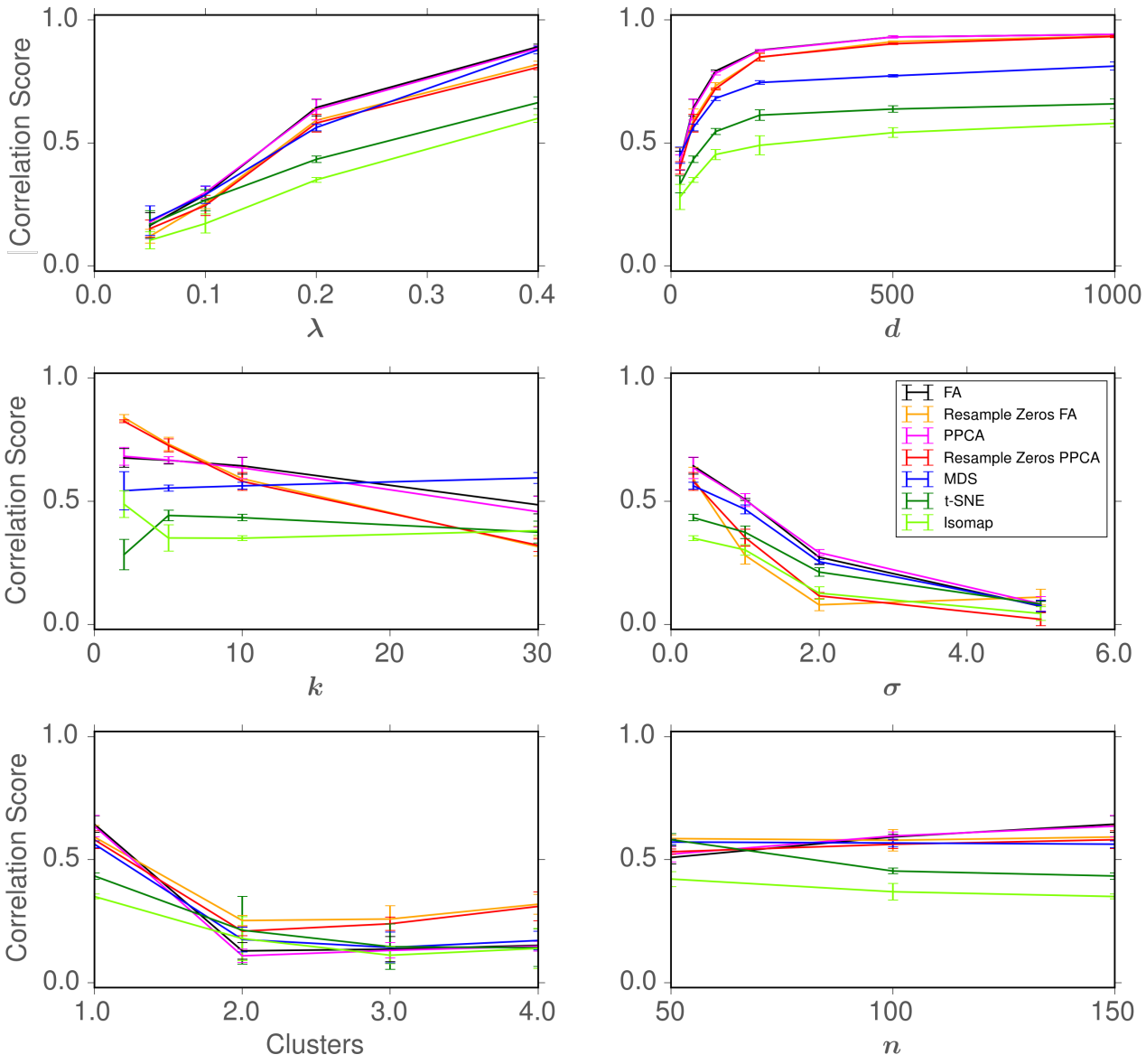


Figure 1.13: Performance of standard dimensionality reduction algorithms on simulated data that is zeroed out with linearly decaying probability $1 - \lambda X$.

1.3.3 Conclusions

For both dimensionality reduction and clustering algorithms, performance worsened considerably as zeros were added to the data, regardless of whether zeros were added with exponentially decreasing, linearly decreasing, or uniform probability. In the case of dimensionality reduction, more complex nonlinear algorithms failed to alleviate the problem, and did not yield performance increases over PPCA and factor analysis. The harmful effect of the zeros is not surprising because standard algorithms are not designed to accommodate zero-inflated data, which is strongly non-Gaussian. Our simulations therefore imply that methods that model zeros may yield performance increases over standard methods. We now develop two such methods.

Chapter 2

Zero-inflated dimensionality reduction

We have shown that standard dimensionality reduction and clustering models do not perform well on simulated, biologically realistic zero-inflated data. We now develop two models which confront this problem: a zero-inflated dimensionality reduction model and a zero-inflated clustering model. In this chapter, we discuss the zero-inflated dimensionality reduction model. Given a high-dimensional $N \times D$ matrix Y, the goal is to produce a $N \times K$ matrix Z with $K \ll D$ that accounts for zeros in which points that are far apart in Y remain far apart in Z. We call our model zero-inflated factor analysis (ZIFA). In all derivations below, we use use $i = 1, ..., N$ to index over samples, $j = 1, ..., D$ to index over genes, and $k = 1, ..., K$ to index over latent dimensions.

2.1 Model

Our generative model is similar to the one used for our dimensionality reduction simulations (Section 1.3.2). Each sample Y_i is drawn independently:

$$Z_i \sim \mathcal{N}(0, I),$$

$$X_i|Z_i \sim \mathcal{N}(AZ_i + \mu, W),$$

$$H_{ij}|X_{ij} \sim Bernoulli(\exp(-\lambda X_{ij}^2)),$$

$$Y_{ij} = \begin{cases} X_{ij} & \text{if } H_{ij} = 0 \\ 0 & \text{if } H_{ij} = 1 \end{cases}$$

where the parameters of the model are the $D \times K$ matrix A, the $D \times D$ covariance matrix W, which we constrain to be diagonal with $W_{jj} = \sigma_j^2$, the $D \times 1$ mean vector μ, and the exponential decay parameter λ. Note that λ is shared across genes: this reduces the number of parameters to be estimated and captures the fact that technical noise should have similar effects across genes. This model is identical to the one used to generate data in Section 1.3.2 except that negative values of X are not set to zero; thus, the model can produce negative gene expression values. While negative gene expression values are not biologically realistic, truncating X at 0 would not yield an analytically tractable model. We optimize the data log-likelihood using the EM algorithm following (Scheffler, 2008; Dempster et al., 1977); we summarize the algorithm in the box below and then describe the algebraic details.

Algorithm 1: EM for zero-inflated dimensionality reduction

```
initialize model parameters A, μ, W, λ;
while parameters not converged do
    E-step: given A, μ, W, λ, compute p(Z, X_0|Y) and
    E[Z], E[ZZ^T], E[X_0], E[X_0^2], E[X_0 Z];
    M-step: compute analytic updates for A, μ, W and optimize λ numerically;
end
```

2.1.1 Initialization of parameters

We initialize A, μ, and W using factor analysis: we fit a factor analysis model to Y, and set A, μ, and W to the model parameters. To initialize λ, for each gene j, we compute the mean of nonzero samples μ and the fraction of samples that are zero, p_0; we then fit a decaying squared exponential curve to the pairs (μ, p_0), and set λ to the fitted exponential

parameter.

2.1.2 Optimization of parameters

2.1.2.1 EM algorithm

The EM algorithm, as applied to our model, seeks to find the maximum likelihood estimate of the parameters Θ given observed data Y which is generated by latent variables Z, X, and H. The EM algorithm increases the log likelihood by iteratively updating Θ; it does not guarantee convergence to a global maximum, but it does guarantee monotonic improvement of the log likelihood at each step. The value of Θ at the nth iteration, Θ_n, is the value that maximizes

$$Q(\Theta_n) = E_{Z,X,H|Y,\Theta_{n-1}} \log p(Z, X, H, Y|\Theta_n), \tag{2.1}$$

$$= \int_{Z,X,H} p(Z, X, H|Y, \Theta_{n-1}) \log p(Z, X, H, Y|\Theta_n) dZ dX dH \tag{2.2}$$

In other words, it is the value of the parameters that maximizes the expected value of the complete log likelihood $p(Z, X, H, Y|\Theta_n)$ under the conditional distribution over the latent variables given the observed data and the parameters at the last iteration. Computing Θ_n requires two steps: the expectation step (E-step) and the maximization step (M-step). In the E-step, we derive an expression for the complete log likelihood $p(Z, X, Y|\Theta_n)$ and compute all necessary expectations under the distribution $p(Z, X|Y, \Theta_{n-1})$. In the M-step, we maximize the expected value $Q(\Theta_n)$ with respect to Θ_n.

2.1.2.2 Complete data likelihood

Because the samples are generated independently, the likelihood of all samples is the product of the likelihoods of each sample. The complete data likelihood for the ith

sample is

$$P(Z_i, X_i, H_i, Y_i|\Theta) = P(Z_i)p(X_i|Z_i)p(H_i|X_i)p(Y_i|H_i, X_i), \tag{2.3}$$

$$= \prod_{k=1}^{d} p(Z_{ik}) \prod_{j=1}^{p} p(X_{ij}|Z_i)p(H_{ij}|X_{ij})p(Y_{ij}|X_{ij}, H_{ij}) \tag{2.4}$$

There are two different cases to consider: $Y_{ij} \neq 0, H_{ij} = 0$, and $Y_{ij} = 0, H_{ij} = 1$. Note the case where $Y_{ij} = 0, H_{ij} = 0$ has probability zero because for $Y_{ij} = 0$ when $H_{ij} = 0$ we must have $X_{ij} = 0$, which means that $H_{ij} = 1$ with probability $\exp(-\lambda X_{ij}^2) = 1$. The case where $Y_{ij} \neq 0, H_{ij} = 1$ has probability zero because $H_{ij} = 1 \rightarrow Y_{ij} = 0$. Thus,

$$P(Z_i, X_i, H_i, Y_i|\Theta) = \prod_{k=1}^{d} p(Z_{ik}) \prod_{j=1}^{p} p(X_{ij}|Z_i)p(H_{ij}|X_{ij})p(Y_{ij}|X_{ij}, H_{ij}), \tag{2.5}$$

$$= \prod_{k=1}^{d} p(Z_{ik}) \prod_{j:Y_{ij}=0} p(X_{ij}|Z_i)p(H_{ij} = 1|X_{ij}) \prod_{j:Y_{ij}\neq 0} p(X_{ij} = Y_{ij}|Z_i)p(H_{ij} = 0|X_{ij}) \tag{2.6}$$

Taking the log of this and substituting in the expressions from the generative model yields the log likelihood.

$$\sum_{k=1}^{d} \log N(Z_{ik}; 0, 1) + \sum_{j:Y_{ij}=0} \log N(X_{ij}; AZ_i + \mu, W) - \lambda X_{ij}^2$$

$$+ \sum_{j:Y_{ij}\neq 0} \log N(Y_{ij}; AZ_i + \mu, W) + \log(1 - \exp(-\lambda Y_{ij}^2)) \tag{2.7}$$

For the first sum we need to compute expectations for Z_{ik} and Z_{ik}^2; for the second sum, we need to additionally compute expectations for X_{ij}, X_{ij}^2, and all cross terms $Z_{ik_1}Z_{ik_2}$ and $Z_{ik}X_{ij}$ under the distribution $p(Z_i, X_i|Y_i, \Theta)$. (In derivations below, we do not include H_i in our latent variables because it is determined given Y_i.)

2.1.2.3 E-step

The E-step requires us to compute expectations for Z_{ik}, Z_{ik}^2, X_{ij}, X_{ij}^2, $Z_{ik_1}Z_{ik_2}$, and $X_{ij}Z_{ik}$ under the conditional distribution $p(Z_i, X_i|Y_i, \Theta)$. While we have $X_i \sim \mathcal{N}(AZ_i + \mu, W)$, we do not have $E[X_i|Y_i] = AE[Z_i|Y_i] + \mu$ because the expectation of the error term given Y_i is nonzero. Thus, we must compute the distribution over X_i and Z_i, not just Z. We use Y_{i+} to denote the elements of Y_i that are nonzero and Y_{i0} to denote the elements of Y_i that are zero (similarly for X_i). We are only interested in the distributions over X_{i0}, since X_{i+} is observed; thus, we need the distribution $p(Z_i, X_{i0}|Y_i)$. We show that this is multivariate normal.

Algorithm 2: Computing $p(Z_i, X_{i0}|Y_i)$

1 Compute the prior distribution $p(Z_i, X_{i0})$;
2 Condition on Y_{i+} to compute $p(Z_i, X_{i0}|Y_{i+})$, parameterized by Σ_c and μ_c;
3 Multiply by $p(Y_{i0}|X_{i0})$;

We can compute the $p(Z_i, X_{i0}|Y_i)$ as follows.

$$p(Z_i, X_{i0}|Y_i) = p(Z_i, X_{i0}|Y_{i0}, Y_{i+}), \tag{2.8}$$

$$\propto p(Y_{i0}, Z_i, X_{i0}|Y_{i+}), \tag{2.9}$$

$$= p(Z_i, X_{i0}|Y_{i+})p(Y_{i0}|X_{i0}) \tag{2.10}$$

The second term is easy to compute: we have $p(Y_{i0}|X_{i0}) = \exp(-\lambda X_{i0}^2)$. We can compute $p(Z_i, X_{i0}|Y_+)$ as follows. We know the prior distribution $p(Z_i, X_i)$ is a multivariate normal (Zhao et al., 2008) with prior mean $\mu^{(p)}$ given by the vector of length $(K + D)$

$$\mu^{(p)} = \begin{pmatrix} 0 \\ \mu \end{pmatrix} \tag{2.11}$$

where the upper 0 denotes the first K entries and the μ denotes the last D entries. The

prior covariance $\Sigma^{(p)}$ is given by the $(K+D)\times(K+D)$ matrix

$$\Sigma^{(p)} = \begin{pmatrix} I & A^T \\ A & AA^T + W \end{pmatrix} \tag{2.12}$$

So we just want the distribution of Z_i, X_{i0} conditional on $X_{i+} = Y_{i+}$. The conditional expectation of a multivariate normal is a multivariate normal with conditional mean μ_c and conditional variance Σ_c given by (Eaton, 1983). Let $\mu_+^{(p)}$ denote the entries of $\mu^{(p)}$ corresponding to X_{i+}, and $\mu_0^{(p)}$ denote the other entries; partition $\Sigma^{(p)}$ similarly. Then we have:

$$\mu_c = \mu_0^{(p)} + \Sigma_{0+}^{(p)}(\Sigma_{++}^{(p)})^{-1}(Y_{i+} - \mu_+^{(p)}), \tag{2.13}$$

$$\Sigma_c = \Sigma_{00}^{(p)} - \Sigma_{0+}^{(p)}(\Sigma_{++}^{(p)})^{-1}\Sigma_{+0}^{(p)} \tag{2.14}$$

Thus, $p(Z_i, X_{i0}|Y_i)$ is the product of an exponential term $\exp(-\lambda X_{i0}^2)$ and a multivariate normal parameterized by μ_c and Σ_c. This product is a multivariate normal. Denote by I_x the diagonal matrix with ones on diagonal elements corresponding to entries of X_{i0} and zeros everywhere else. Then the covariance matrix of $p(Z_i, X_{i0}|Y_i)$ is

$$(\Sigma_c^{-1} + 2\lambda I_x)^{-1} \tag{2.15}$$

and the mean is

$$(\Sigma_c^{-1} + 2\lambda I_x)^{-1}\Sigma_c^{-1}\mu_c \tag{2.16}$$

Although these formulae involve inverses of high-dimensional matrices, the expressions can be simplified such that no high-dimensional inversions are required; we describe how to do this in Appendix A using the Woodbury identity (Petersen et al., 2008) and block matrix inversion.

2.1.2.4 M-step

In the M-step we first update A and μ; then we update W, whose value depends on A and μ; then we update λ. All updates are analytic except for λ which we optimize numerically. We want to optimize the expected value of the complete log likelihood:

$$\sum_{i=1}^{n}\Bigg(\sum_{k=1}^{d}\log N(Z_{ik};0,1)+\sum_{j:Y_{ij}=0}\log N(X_{ij};AZ_i+\mu,W)-\lambda X_{ij}^2 \\ +\sum_{j:Y_{ij}\neq 0}\log N(Y_{ij};AZ_i+\mu,W)+\log(1-\exp(-\lambda Y_{ij}^2))\Bigg) \tag{2.17}$$

Because the Gaussian log likelihood is concave in its mean and precision matrix (Boyd and Vandenberghe, 2004), and A, μ and W appear only in the Gaussian log likelihood terms, we can optimize the values of A, μ and W by setting the first derivatives of the log likelihood to zero. Collecting the terms that contain μ_j and setting $\frac{d}{d\mu_j}=0$ yields

$$\mu_j=\frac{1}{N}\Bigg[\sum_{i:Y_{ij}=0}\Bigg(E[X_{ij}]-\sum_{k=1}^{K}A_{jk}E[Z_{ik}]\Bigg)+\sum_{i:Y_{ij}>0}\Bigg(Y_{ij}-\sum_{k=1}^{K}A_{jk}E[Z_{ik}]\Bigg)\Bigg] \tag{2.18}$$

Doing the same for A_{jk} yields

$$A_{jk}=\frac{1}{\sum_{i=1}^{N}E[Z_{ik}^2]}\Bigg[\sum_{i:Y_{ij}=0}\Bigg(E[X_{ij}Z_{ik}]-\mu_j E[Z_{ik}]-\sum_{k'\neq k}A_{jk'}E[Z_{ik}Z_{ik'}]\Bigg)+ \\ \sum_{i:Y_{ij}>0}\Bigg(Y_{ij}E[Z_{ik}]-\mu_j E[Z_{ik}]-\sum_{k'\neq k}A_{jk'}E[Z_{ik}Z_{ik'}]\Bigg)\Bigg] \tag{2.19}$$

The optimal value for μ_j depends on $A_{j1}, A_{j2}, ..., A_{jK}$ and vice versa. We can express these constraints as a matrix equation for each j and solve them for each j independently. We have $B_j u_j - c_j = 0$, where B_j is a $(D+1)\times(D+1)$ matrix and u_j and c_j are vectors of

length $(D+1)$:

$$u_j = \begin{pmatrix} A_{j1} \\ A_{j2} \\ \vdots \\ A_{jD} \\ \mu_j \end{pmatrix} \tag{2.20}$$

$$B_j = \begin{pmatrix} 1 & \frac{\sum_i E[Z_{i1}Z_{i2}]}{\sum_i E[Z_{i1}^2]} & \dots & \frac{\sum_i E[Z_{i1}]}{\sum_i E[Z_{i1}^2]} \\ \frac{\sum_i E[Z_{i2}Z_{i1}]}{\sum_i E[Z_{i2}^2]} & 1 & \dots & \frac{\sum_i E[Z_{i2}]}{\sum_i E[Z_{i2}^2]} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N}\sum_i E[Z_{i1}] & \frac{1}{n}\sum_i E[Z_{i2}] & \dots & 1 \end{pmatrix} \tag{2.21}$$

$$c_j = \begin{pmatrix} \frac{\sum_{i:Y_{ij}=0} E[X_{ij}Z_{i1}] + \sum_{i:Y_{ij}>0} Y_{ij}E[Z_{i1}]}{\sum_i E[Z_{i1}^2]} \\ \frac{\sum_{i:Y_{ij}=0} E[X_{ij}Z_{i2}] + \sum_{i:Y_{ij}>0} Y_{ij}E[Z_{i2}]}{\sum_i E[Z_{i2}^2]} \\ \vdots \\ \frac{1}{N}\left(\sum_{i:Y_{ij}=0} E[X_{ij}] + \sum_{i:Y_{ij}>0} Y_{ij} \right) \end{pmatrix} \tag{2.22}$$

Having solved for μ_j and $A_{j1}, A_{j2}, ..., A_{jK}$, we use the updated values to update σ_j. Defining $m_{ij} = \sum_{k=1}^{K} A_{jk}Z_{ik} + \mu_j$, we have

$$\sigma_j^2 = \frac{1}{N}\left(\sum_{i:Y_{ij}=0} (E[X_{ij}^2] - 2E[X_{ij}m_{ij}] + E[m_{ij}^2]) + \sum_{i:Y_{ij}>0} (Y_{ij}^2 - 2Y_{ij}E[m_{ij}] + E[m_{ij}^2]) \right) \tag{2.23}$$

This expression is very similar to the expression for the variance in factor analysis, but with Y_{ij} replaced with the expected value of X_{ij} for zero values of Y_{ij}. This makes sense because when Y_{ij} is zero, X_{ij} is unobserved, but X_{ij} is otherwise equal to Y_{ij}. We note

that constraining all σ_j to be equal yields zero-inflated PPCA rather than zero-inflated factor analysis: we have

$$\sigma = \frac{1}{ND} \sum_j \left(\sum_{i:Y_{ij}=0} (E[X_{ij}^2] - 2E[X_{ij}m_{ij}] + E[m_{ij}^2]) + \sum_{i:Y_{ij}>0} (Y_{ij}^2 - 2Y_{ij}E[m_{ij}] + E[m_{ij}^2]) \right) \tag{2.24}$$

We then optimize λ numerically using BFGS (bounding below at 0) (Liu and Nocedal, 1989) and the M-step is complete. We confirm (Figure 2.1) that this procedure yields a value of λ which maximizes the objective function.

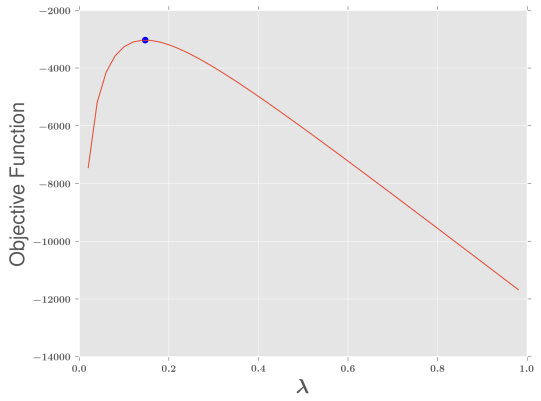


Figure 2.1: The value of the objective function as a function of λ (red line) with the value of λ chosen by BFGS (blue dot) on simulated data (Section 1.3.2.1), illustrating that the optimal value of λ is correctly found.

2.1.3 Computing the marginal likelihood of Y_i

We now describe how to compute the marginal likelihood of Y_i (the model evidence) given the model parameters A, μ, W, and λ.

To compute the marginal likelihood $p(Y_i)$, we have

$$p(Y_i) = \int_{X_i} p(X_i)p(Y_i|X_i)dX_i, \tag{2.25}$$

$$= \int_{X_i} p(X_{i+})p(X_{i0}|X_{i+}) \prod_j p(Y_{ij}|X_{ij})dX_i, \tag{2.26}$$

$$= \int_{X_i} p(X_{i+})p(X_{i0}|X_{i+}) \prod_{j:Y_{ij}\neq 0} p(Y_{ij}|X_{ij}) \prod_{j:Y_{ij}=0} p(Y_{ij}|X_{ij})dX_i, \tag{2.27}$$

$$= \int_{X_i} p(X_{i+})p(X_{i0}|X_{i+}) \prod_{j:Y_{ij}\neq 0} \mathbb{1}(X_{ij} = Y_{ij})(1 - \exp(-\lambda X_{ij}^2)) \prod_{j:Y_{ij}=0} \exp(-\lambda X_{ij}^2)dX_i, \tag{2.28}$$

$$= N(Y_{i+}; \mu_+, A_+A_+^T + W_+)(1 - \exp(-\lambda Y_{i+}^2)) \int_{X_{i0}} p(X_{i0}|X_{i+} = Y_{i+}) \exp(-\lambda X_{i0}^2)dX_{i0} \tag{2.29}$$

where the last step follows because the integrand is zero except where $Y_+ = X_+$. Note because the X are not independent, $p(X_{i0}|X_{i+} = Y_{i+}) \neq p(X_{i0})$. But the conditional distributions of a multivariate normal are multivariate normal as discussed above. Thus, denoting the mean and covariance of the distribution $p(X_{i0}|X_{i+} = Y_{i+})$ by μ_0, Σ_0:

$$p(Y_i) = N(Y_{i+}; \mu_+, A_+A_+^T + W_+)(1 - \exp(-Y_{i+}^2\lambda)) \int_{X_{i0}} N(X_{i0}|\mu_0, \Sigma_0) \exp(-\lambda X_{i0}^2)dX_{i0} \tag{2.30}$$

The integral term can be written as the integral of a multivariate normal times a constant, yielding

$$\int_{X_{i0}} N(X_{i0}|\mu_0, \Sigma_0) \exp(-\lambda X_{i0}^2)dX_0 = \sqrt{\frac{\det(\Sigma')}{\det(\Sigma_0)}} \exp\left(\frac{1}{2}\mu_0^T(\Sigma_0^{-1}\Sigma'\Sigma_0^{-1} - \Sigma_0^{-1})\mu_0\right) \tag{2.31}$$

with $\Sigma' = (\Sigma_0^{-1} + 2\lambda I)^{-1}$. Combining all expressions, we have

$$p(Y_i) = N(Y_{i+}; \mu_+, A_+A_+^T + W_+)(1 - \exp(-Y_{i+}^2\lambda))\sqrt{\frac{\det(\Sigma')}{\det(\Sigma_0)}}\exp\left(\frac{1}{2}\mu_0^T(\Sigma_0^{-1}\Sigma'\Sigma_0^{-1} - \Sigma_0^{-1})\mu_0\right) \tag{2.32}$$

2.2 Performance in simulations

In Figures 2.2-2.4, we compare the performance of ZIFA to the alternatives discussed in Chapter 1. Simulated data for Figure 2.2 is drawn using the same generative model used to run simulations in Section 1.3.2 with exponentially decaying zero probability $f(X_{ij}) = \exp(-\lambda X_{ij}^2)$; data for Figure 2.3 is drawn with linearly decaying zero probability $f(X_{ij}) = 1 - \lambda X_{ij}$, and data for Figure 2.4 is drawn with uniform zero probability $f(X_{ij}) = 1 - \lambda$.

We evaluate performance, as described in Section 1.3.2, using the correlation ρ_c between the true distance matrix and the estimated distance matrix. We find that ZIFA consistently outperforms factor analysis and PPCA regardless of whether we resample zero values. ZIFA's superior performance is also robust to model misspecification: it continues to outperform factor analysis and PPCA even if there are multiple clusters in the data (which the ZIFA model does not account for) or if zeros are added linearly or uniformly. ZIFA's performance degrades to that of factor analysis and PPCA in the limit of large σ, when all models perform badly because there is too much noise to infer parameters. In the limit of large λ (for which there are very few zeros) all models perform similarly well.

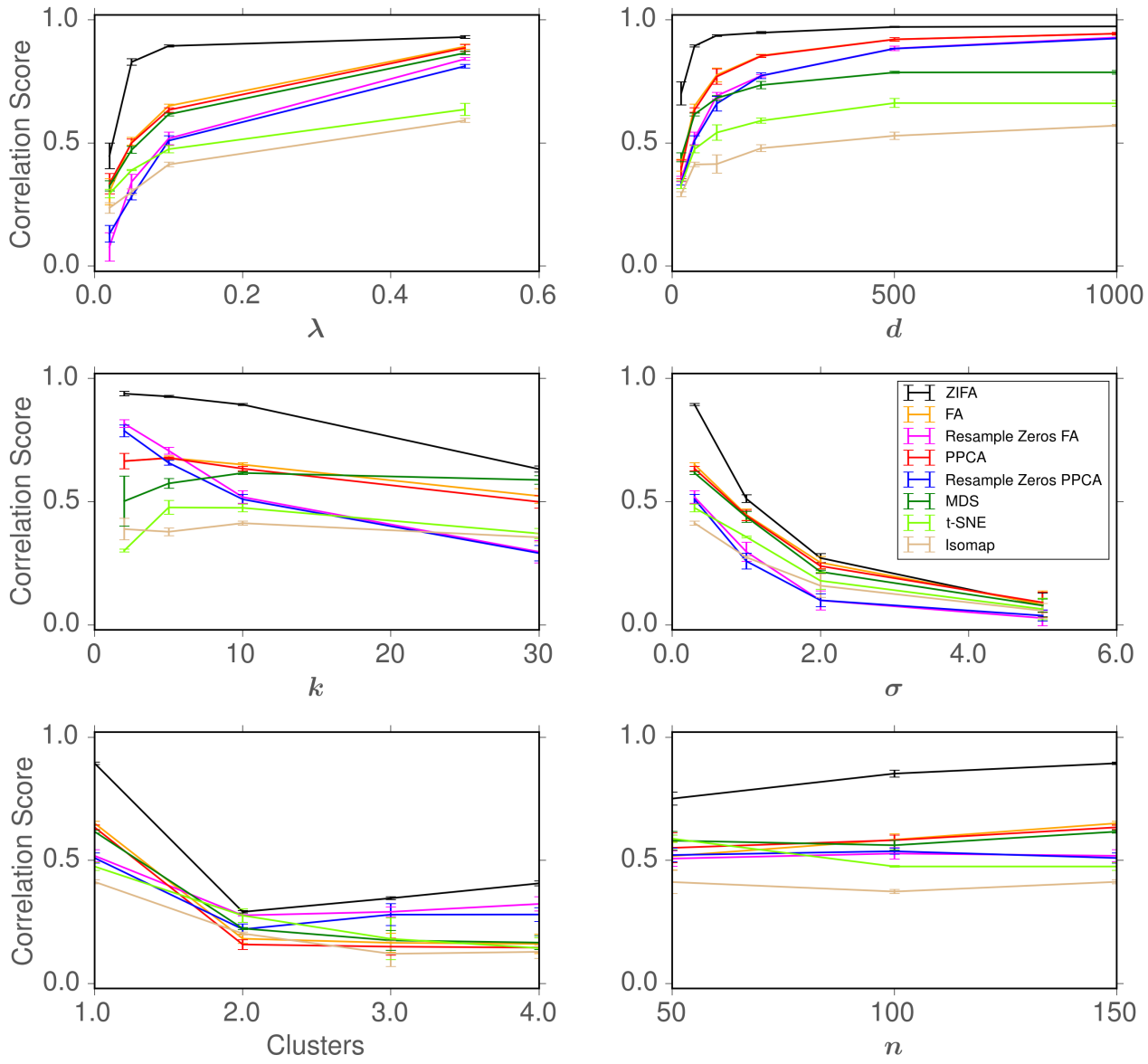


Figure 2.2: Performance of dimensionality reduction algorithms on simulated data that is zeroed out with exponentially decaying probability $\exp(-\lambda X^2)$.

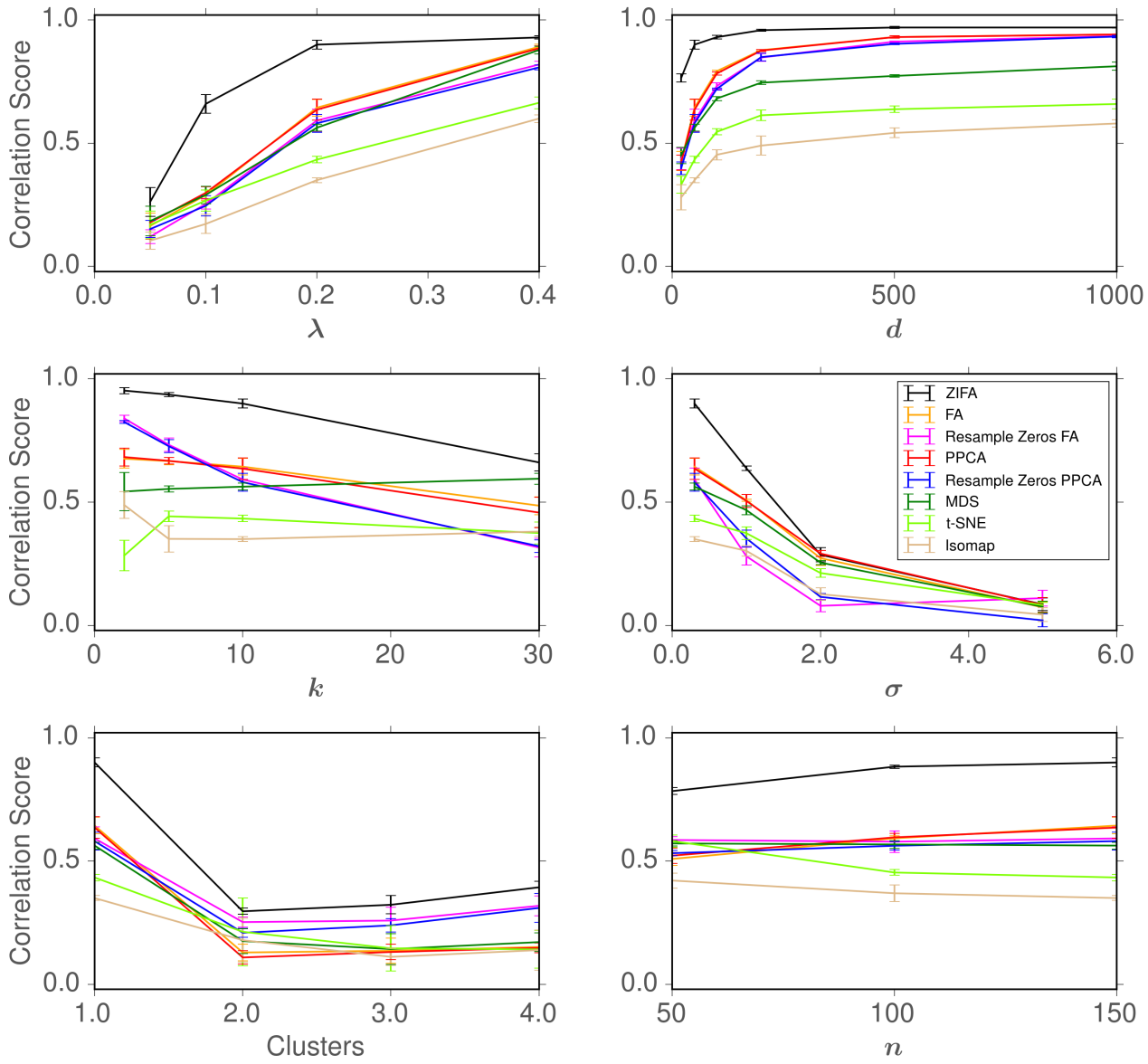


Figure 2.3: Performance of dimensionality reduction algorithms on simulated data that is zeroed out with linearly decaying probability $1 - \lambda X$

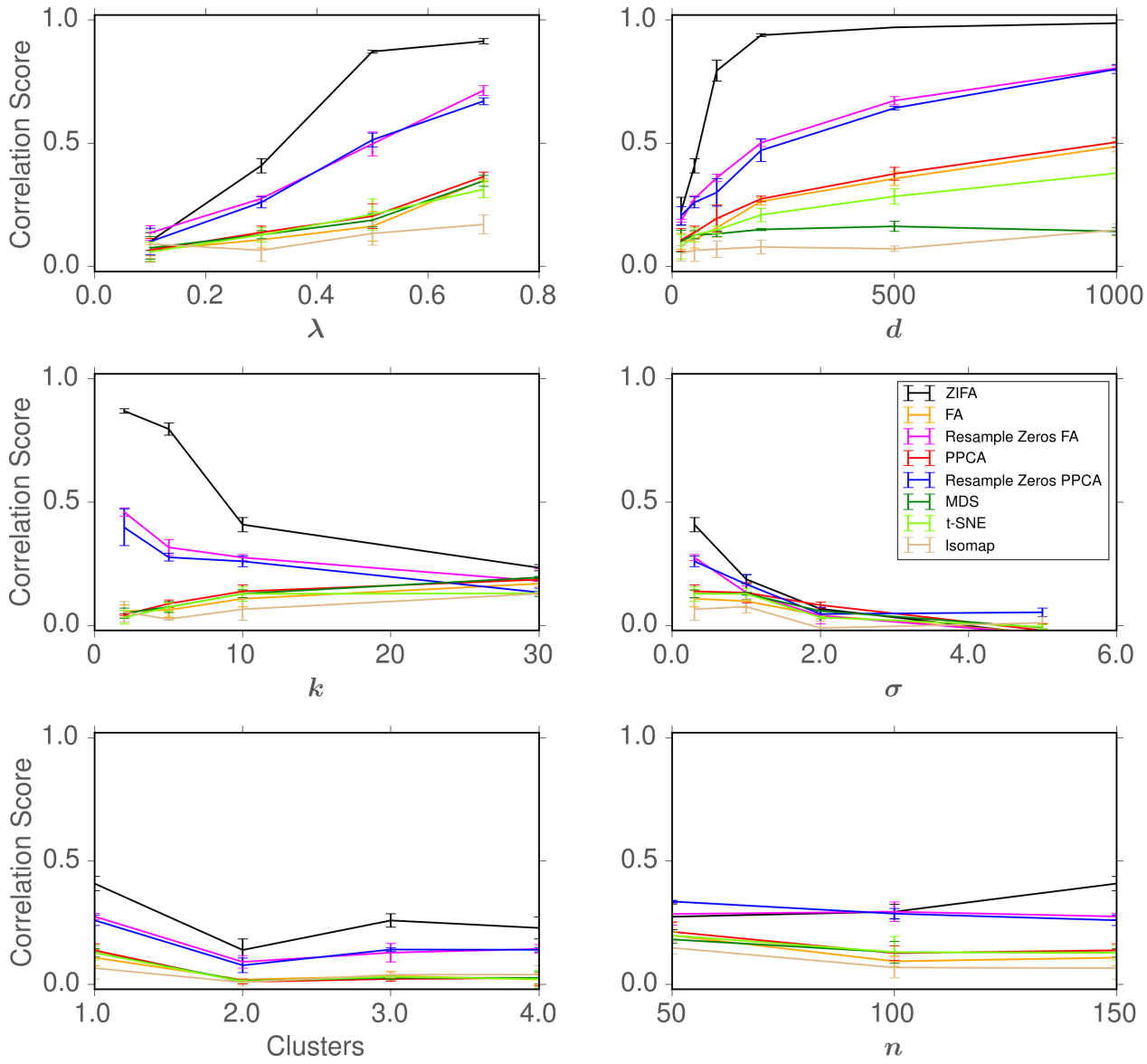


Figure 2.4: Performance of dimensionality reduction algorithms on simulated data that is zeroed out with uniform probability $1 - \lambda$.

2.2.1 Performance with perfect parameters

To quantify how effectively the EM algorithm was optimizing parameters, we investigated how much performance improvement could be achieved with perfect estimation of the model parameters. We found that using the true parameters used to generate the data yielded considerably better performance than using the parameters estimated via the EM algorithm (Figure 2.5). This suggests that initializing the algorithm repeatedly, or using different initial conditions, may yield improved performance. (To assess which initialization yielded the best fit to the data when the true parameters were unknown, a metric such as held-out test log-likelihood could be used.) Unsurprisingly, the gap between the EM-estimated model and the true model is smallest when there are few zeros, fewer parameters to estimate (K small) or as the number of samples increases; all these factors make the task of parameter estimation easier.

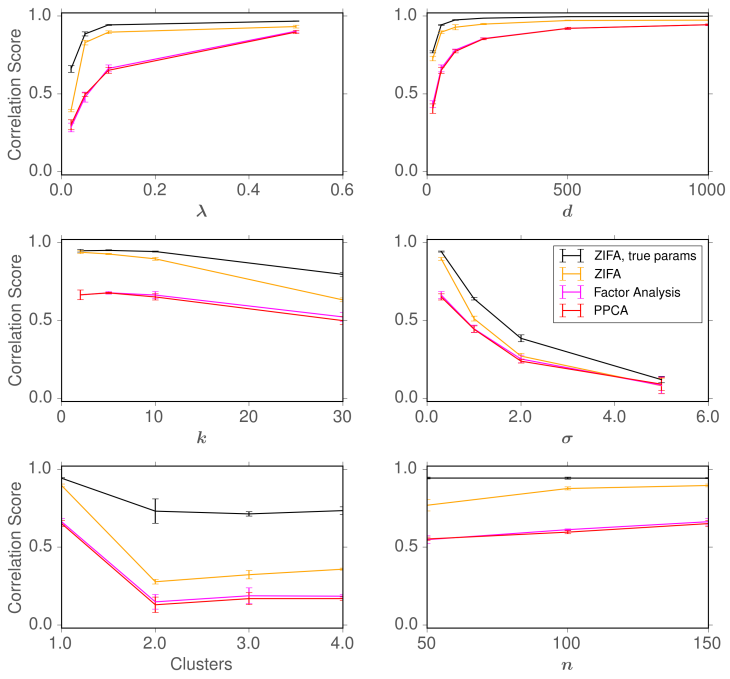


Figure 2.5: ZIFA performance using the true generative model parameters A, μ, W, λ (black line) is considerably better than ZIFA performance using the EM-estimated model parameters (orange line).

2.3 Performance on biological datasets

We have shown that ZIFA outperforms other dimensionality reduction algorithms on biologically realistic simulated data. But the validity of this test is contingent on the validity of our simulation assumptions. We therefore test how ZIFA performs on five single-cell datasets (Trapnell et al., 2014; Shalek et al., 2014; Pollen et al., 2014; Buettner et al., 2015; Mead, 2015) in comparison to factor analysis and PPCA, which performed best in our simulations and are more widely used than the other algorithms.

2.3.1 Performance metric

The goal of ZIFA is to more accurately model the distributions of scRNA-seq data. We therefore evaluate how data sampled from our model compares to the true data. For each gene j, we compare the predicted distribution $\hat{Y}_j$ to the observed distribution Y_j as follows: we compute the proportion of values in Y_j and $\hat{Y}_j$ that fall into 30 bins, then compute the divergence between the histograms Δ_j. If h_n is the proportion of values in bin n for the true distribution, and $\hat{h}_n$ for the predicted distribution, then the histogram divergence is

$$\sum_{n=1}^{30} |h_n - \hat{h}_n| \tag{2.33}$$

We compute the fraction of genes for which Δ_j from our method is less than Δ_j from standard methods. To prevent overfitting, we assess fit on a test set. We fit the model for each dataset on a training set containing 70% of the datapoints, and compute the difference between the histograms on the remaining 30% of datapoints.

We note that comparing ZIFA's predicted distributions to those from factor analysis or PPCA is somewhat difficult for two reasons.

1. The ZIFA model yields a nonzero probability that a gene will be *exactly* zero, while the factor analysis and PPCA models do not. This disparity is reminiscent of the

so-called "Indian GPA Problem" (Wood, 2014): if GPAs in India range from 0.0 - 10.0, and GPAs in America range from 0.0 - 4.0, with some nonzero fraction of students earning perfect GPAs in both countries, what is the probability that a student with a GPA of exactly 4.0 is American? The answer – 1 – highlights the unintuitive behavior that can result from point masses in probability distributions, as occurs in the ZIFA model.

2. The ZIFA model contains an additional hidden variable – X – between the low-dimensional latent Z and the observed Y. This produces unintuitive behavior when the ZIFA log likelihood is compared to the factor analysis log likelihood. For example, we would expect that as $\lambda \to \infty$, the ZIFA log likelihood converges to the factor analysis log likelihood (assuming the other parameters are the same). In fact, however, this does not occur, because as $\lambda \to \infty$, the integral in equation 2.29 vanishes except at zero, making the likelihood of zero Y go to zero. This unintuitive effect is not specific to our model. Consider a much simpler case. In model A, a one-dimensional Y is drawn from the uniform distribution $U(0, 1)$. In model B, the "zero-inflated model", a one-dimensional X is drawn from the uniform distribution $U(0, 1)$; H is then drawn from $Bernoulli(\exp(-\lambda X^2))$; if $H = 1$, $Y = 0$, and if $H = 0$, $Y = X$. Consider the likelihood under both models for $Y = 0$. Under model A, the likelihood is 1. Under model B, it is $\int_0^1 \exp(-\lambda X^2) dX$. Again, as $\lambda \to \infty$, the likelihood of $Y = 0$ under model B goes to 0.

The histogram divergence metric circumvents these subtleties and yields an intuitive basis for comparing ZIFA to factor analysis and PPCA.

2.3.2 Comparative performance

We compared ZIFA to standard factor analysis and PPCA on five biological datasets. For each dataset, we selected 25, 100, 250, or 1,000 genes at random, filtered out genes that were zero in more than 95% of samples, and used five latent dimensions. We averaged

performance across 100 replicates. ZIFA yielded better fits than factor analysis and PPCA to at least 80% of genes on all five datasets (Table 2.1). This superior performance was especially evident on genes with small μ and large p_0 (Figure 2.6): this is expected because these are the genes for which modeling zero-inflation is particularly important. ZIFA performed better on some datasets than others; this is likely because some datasets better fit the model assumptions. For example, the slightly inferior performance on the CML dataset may occur because the dataset contained two clusters of genes (Figure 1.6) and was less well fit by the $p_0 = \exp(-\lambda X^2)$ model.

Dataset	**Method**	**Number of Genes**			
		25	*100*	*250*	*1000*
Differentiating T cells	Factor analysis	$86 \pm 6.6\%$	$84 \pm 4.6\%$	$82 \pm 4.9\%$	$84 \pm 8.8\%$
	PPCA	$88 \pm 6.3\%$	$87 \pm 4.1\%$	$89 \pm 4.5\%$	$100 \pm 0.3\%$
11 Populations	Factor analysis	$97 \pm 3.7\%$	$96 \pm 2.5\%$	$96 \pm 2.1\%$	$95 \pm 2.7\%$
	PPCA	$98 \pm 3.2\%$	$97 \pm 2.0\%$	$97 \pm 1.5\%$	$99 \pm 0.6\%$
Myoblasts	Factor analysis	$97 \pm 3.3\%$	$97 \pm 2.4\%$	$96 \pm 2.7\%$	$95 \pm 2.7\%$
	PPCA	$97 \pm 3.2\%$	$96 \pm 2.3\%$	$96 \pm 2.1\%$	$99 \pm 1.7\%$
CML	Factor analysis	$85 \pm 7.9\%$	$84 \pm 5.4\%$	$84 \pm 5.6\%$	$86 \pm 6.3\%$
	PPCA	$88 \pm 6.8\%$	$88 \pm 5.4\%$	$92 \pm 5.3\%$	$99 \pm 1.6\%$
Bone Marrow	Factor analysis	$98 \pm 3.0\%$	$97 \pm 2.0\%$	$97 \pm 1.7\%$	$97 \pm 1.7\%$
	PPCA	$98 \pm 3.1\%$	$97 \pm 1.8\%$	$97 \pm 1.4\%$	$97 \pm 1.3\%$

Table 2.1: Comparison of ZIFA to PPCA and factor analysis on five biological datasets. Columns are the number of genes in the dataset. Genes are selected at random. Percentages denote the proportion of genes for which ZIFA had a smaller histogram divergence than factor analysis/PPCA, averaged across 100 replicates.

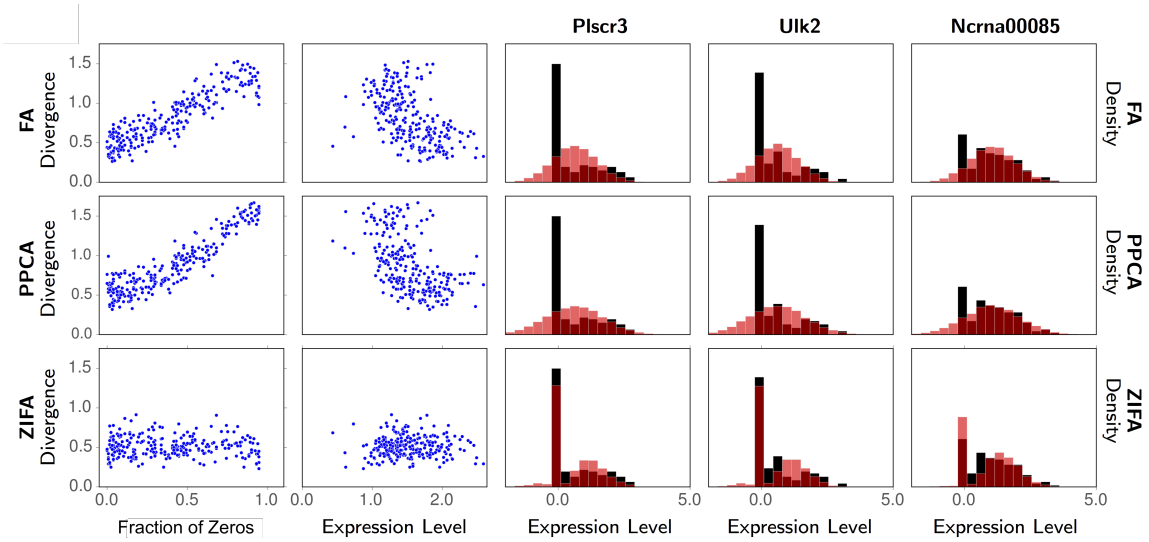


Figure 2.6: **Scatter plots, left**: Factor analysis divergences (top two plots), PPCA divergences (middle two plots), and ZIFA divergences (bottom two plots) for the T cell dataset. Each point is a gene; the y-axis plots the divergence between the predicted distribution and the actual distribution, with higher scores denoting a worse fit. Factor analysis and PPCA best model genes with a small fraction of zeros (left plots) and high expression level (right plots) whereas ZIFA has smaller divergences with little dependence on fraction of zeros or expression level. **Histograms, right**: Illustration of fits to three genes. Black shows the actual distribution; red shows the predicted distribution. Factor analysis and PPCA are unable to fit genes with a large zero spike.

2.4 Estimation of covariance matrices

When analyzing biological data it is often useful to compute the gene coexpression matrix C, where C_{ij} is the correlation between the expression levels of genes i and j. C can then be clustered using methods including affinity propagation (Frey and Dueck, 2007) or spectral clustering (Ng et al., 2002) to reveal modules of coexpressed genes. Such modules provide insight into cellular function because they may represent groups of genes that work together or are coregulated. Genes whose expression levels are correlated are more likely to share common functions, and modules that are upregulated within a cell type may be important to cell type specific function (Pierson et al., 2015).

The factor analysis and PPCA models yield an estimate of the gene coexpression matrix: $C = AA^T + W$. However, estimates of A and W will be distorted because the models do not account for zeros. We investigated whether ZIFA would yield a more

accurate estimate of the true covariance matrix using the simulated data described in 1.3.2. To evaluate performance, we used the Spearman correlation between the true covariance matrix and the estimated covariance matrix. We found that ZIFA improved performance (Figures 2.7 - 2.9).

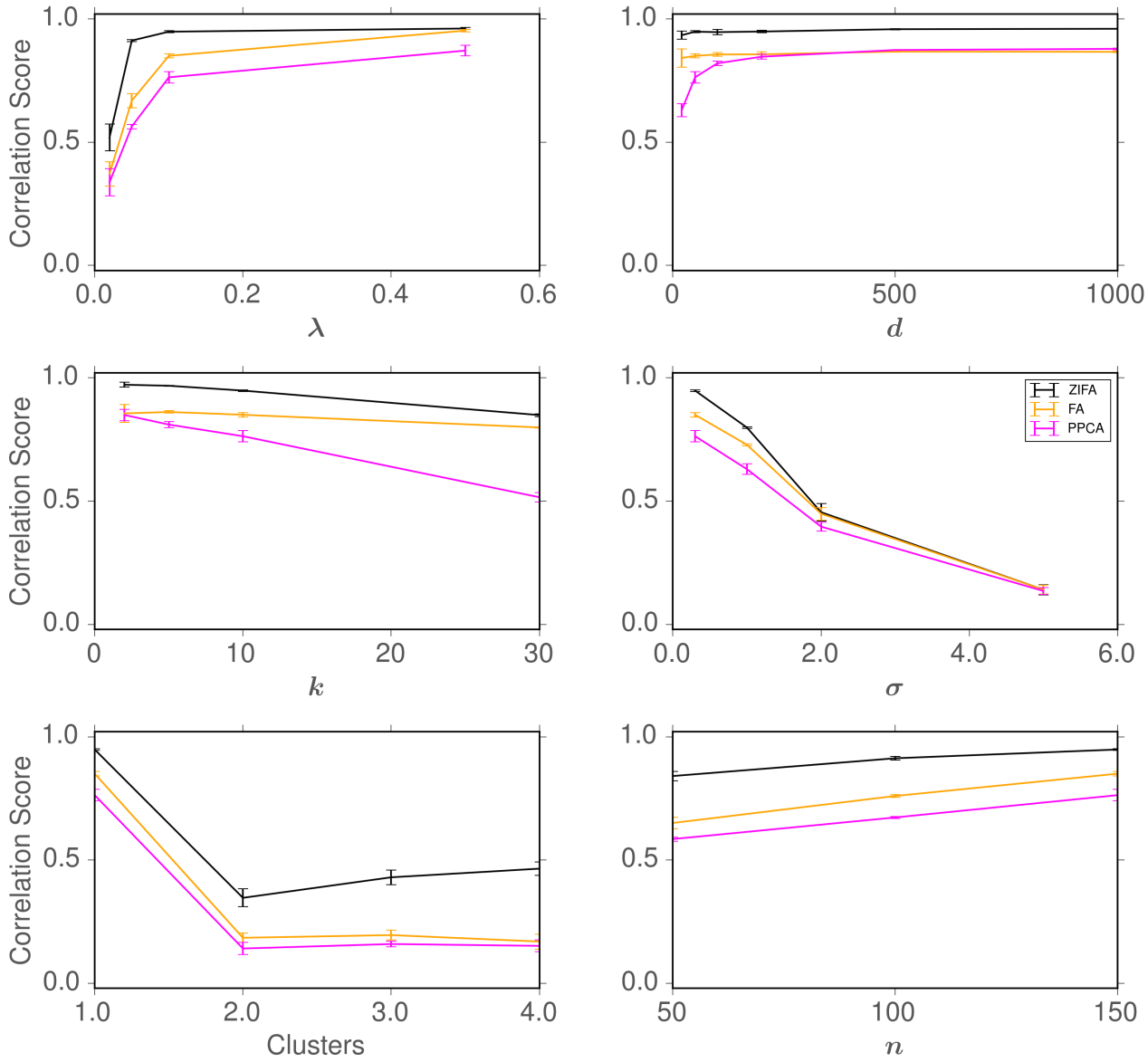


Figure 2.7: Correlation between the true correlation matrix and the estimated correlation matrix on simulated data that is zeroed out with exponentially decaying probability $\exp(-\lambda X^2)$.

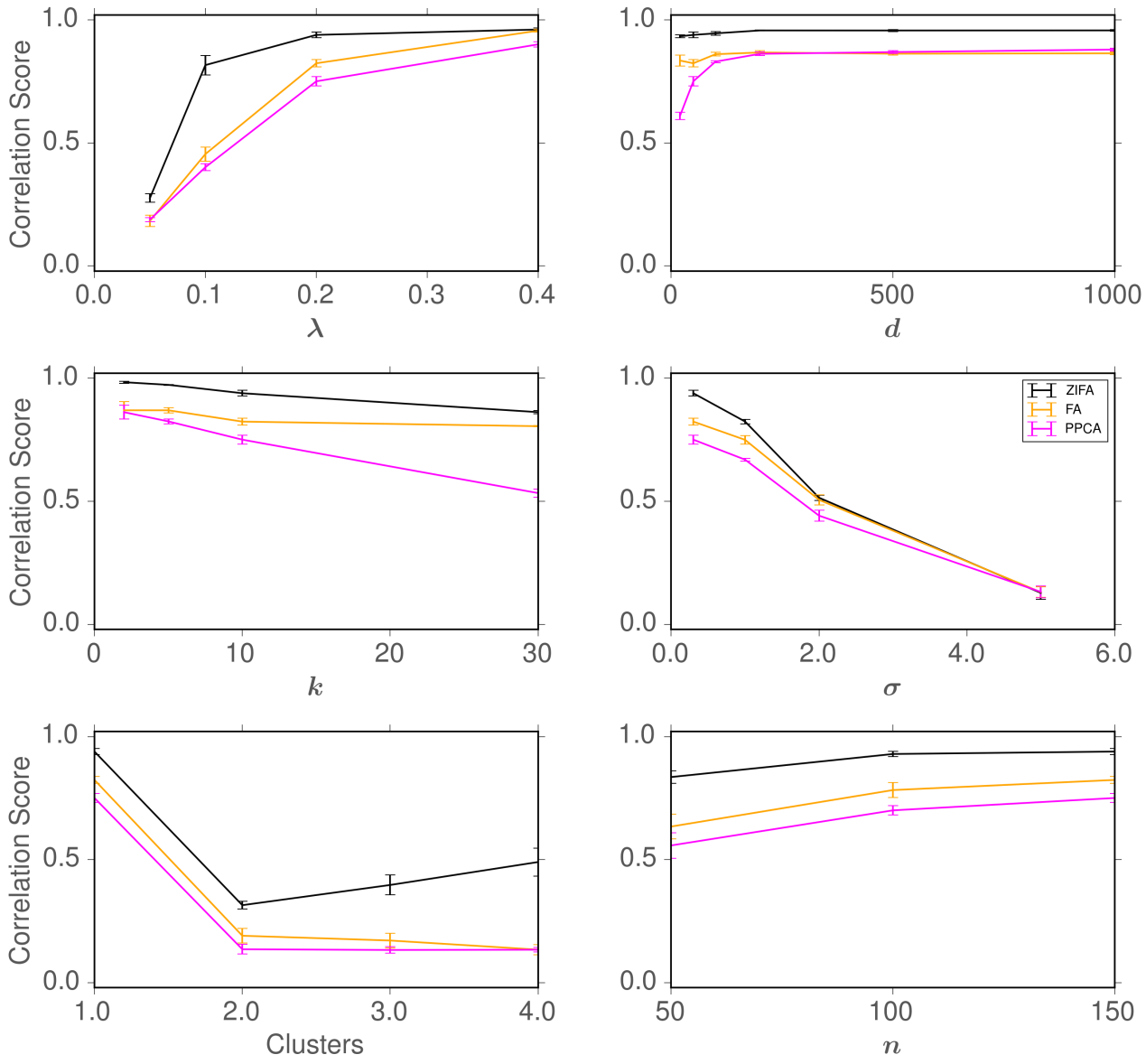


Figure 2.8: Correlation between the true correlation matrix and the estimated correlation matrix on simulated data that is zeroed out with linearly decaying probability $1 - \lambda X$.

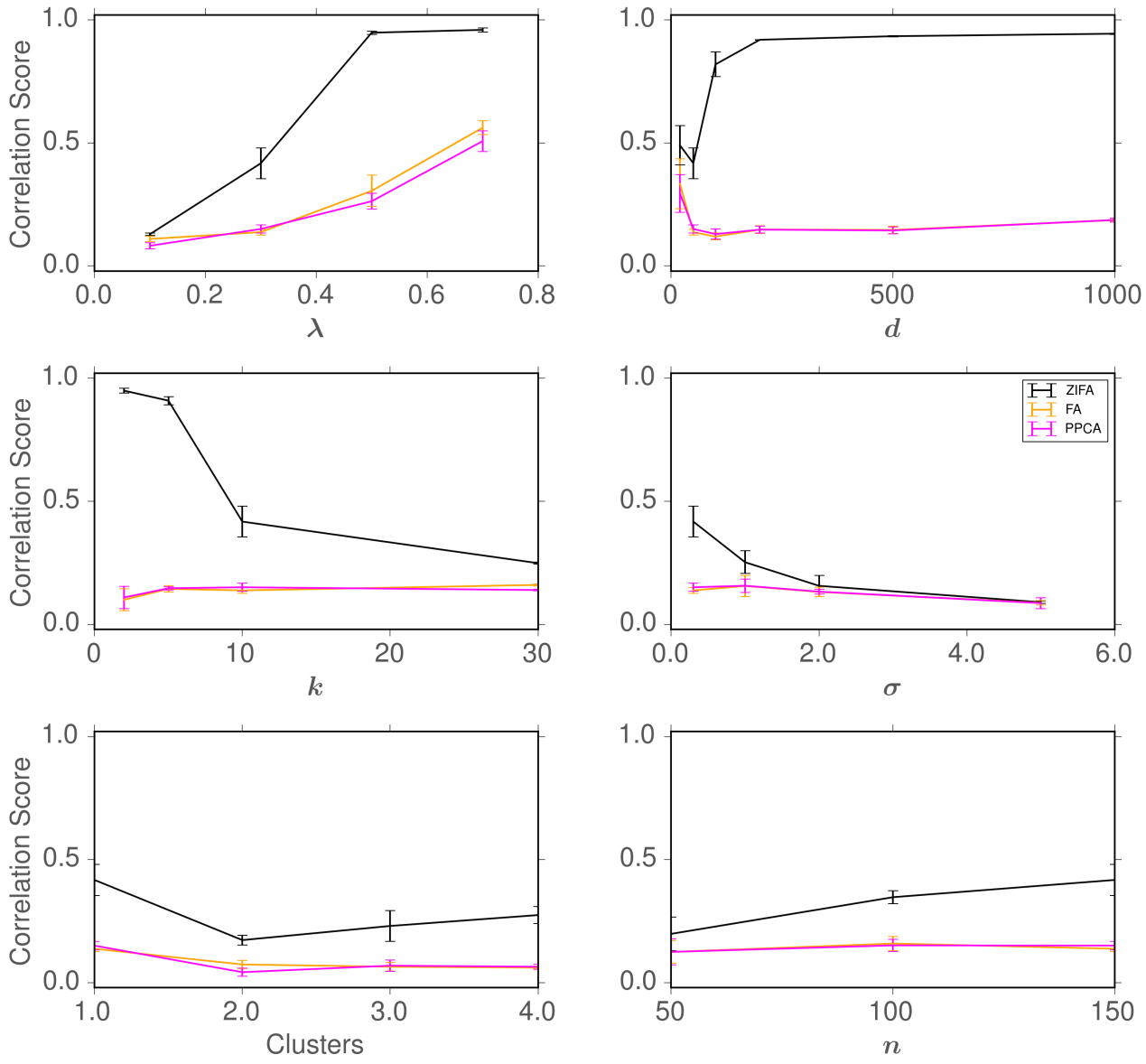


Figure 2.9: Correlation between the true correlation matrix and the estimated correlation matrix on simulated data that is zeroed out with uniform probability $1 - \lambda$.

2.4.1 Identification of modules in biological datasets

We selected the 1,000 genes with the largest variance from the eleven tissues dataset, estimated the covariance matrix using ZIFA, and used the affinity propagation algorithm on the estimated covariance matrix to identify 11 coexpressed clusters. The covariance matrix did display block-diagonal structure (Figure 2.10). In Table 2.2, we list the clusters that are enriched for DAVID biological functions (Huang et al., 2008) along with all tissues in which the mean expression of genes in the cluster is significantly higher in tissue samples than in non-tissue samples. (We used a paired t-test that compared the the mean expression of each gene in the tissue to the mean expression of each gene outside the tissue). We find several biologically plausible clusters: for example, a cluster enriched for neural function which is upregulated in neural tissues, and a cluster enriched for cell-cycle function which is upregulated in in leukemia cells, neural progenitor cells, and stem cells.

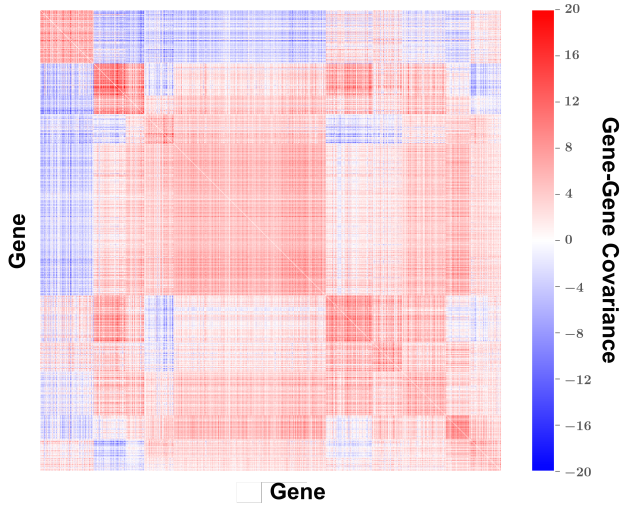


Figure 2.10: The estimated covariance matrix from the eleven tissues dataset, sorted by the cluster indices from affinity propagation, reveals diagonal blocks of coexpressed genes. Color denotes the estimated value of $C = AA^T + W$.

Cluster Functions	**Number of Genes**	**Tissues in Which Upregulated**
neuron differentiation, neuron development, cell morphogenesis involved in neuron differentiation, cell part morphogenesis,neuron projection morphogenesis	117	GW21+3, neural progenitor cell, GW16, stem cell, GW21
cell cycle phase, M phase, cell cycle, cell cycle process, mitotic cell cycle	50	myelogenous leukemia, neural progenitor cell, stem cell, foreskin keratinocytes, B lymphoblast
extracellular region, Secreted, extracellular region part, cell-substrate junction	69	foreskin keratinocytes, breast cancer, fibroblasts
intracellular organelle lumen, organelle lumen, membrane-enclosed lumen, mitochondrion, nuclear lumen	328	myelogenous leukemia, stem cell, breast cancer, B lymphoblast, foreskin keratinocytes
glycoprotein, cell adhesion, biological adhesion, glycosylation site:N-linked, regulation of cell growth	100	breast cancer, foreskin keratinocytes, stem cell, fibroblasts

Table 2.2: Coexpressed clusters in the eleven tissues dataset identified by ZIFA and affinity propagation which are enriched for biological functions. We list the five functions with the lowest p-values (all functions are significantly enriched using a false discovery rate of .05) and the tissues in which each cluster is significantly upregulated.

As a second example, we selected the 1,000 genes with the largest variance from the leukemia dataset, estimated the covariance matrix using ZIFA, and used the affinity propagation algorithm on the estimated covariance matrix to identify 9 coexpressed clusters. The leukemia dataset contains two groups of blood cells: cancerous cells that carry the BCR-ABL mutation, and non-mutated cells that do not. We display the clusters which are enriched for biological functions in Table 2.3, along with whether they show higher expression in BCR-ABL cells or normal blood cells.

Cluster Functions	**Number of Genes**	**Higher Expressed In**
membrane-enclosed lumen, intracellular organelle lumen, organelle lumen, nuclear lumen,non-membrane-bounded organelle	158	BCR-ABL mutant cells
mitochondrial part, mitochondrial inner membrane, organelle inner membrane, respiratory chain,mitochondrion	77	BCR-ABL mutant cells
nucleosome, chromatin assembly, nucleosome assembly, protein-DNA complex,DNA packaging	17	Normal blood cells

Table 2.3: Coexpressed clusters in the CML dataset identified by ZIFA and affinity propagation which are enriched for biological functions.

Chapter 3

Zero-Inflated Mixture Models

Having produced a model which enables dimensionality reduction of single-cell data, we now describe a model which accomplishes another common task: clustering. Given a high-dimensional $N \times P$ matrix Y, the goal is to infer a clustering assignment vector Z of length N such that points that points that lie close to each other are assigned to the same cluster. Our model is a zero-inflated mixture model (ZIMM).

3.1 Model

The generative model is similar to the one used to produce simulated data in Section 1.3.1:

$$Z_i \sim Multinomial(\phi),$$

$$X_{ij}|(Z_i = k) \sim N(\mu_{jk}, \sigma^2_{jk}),$$

$$H_{ij}|X_{ij} \sim Bernoulli(\exp(-\lambda X^2_{ij})),$$

$$Y_{ij} = \begin{cases} X_{ij} & \text{if } H_{ij} = 0 \\ 0 & \text{if } H_{ij} = 1 \end{cases}$$

The only difference between this model and the one used in Section 1.3.1 is that

except that negative values of X are not set to zero, because that would render the model intractable; thus, the model can produce negative gene expression values. We optimize the data log-likelihood using the EM algorithm as before. We provide a boxed summary of the optimization algorithm below followed by the full derivation.

Algorithm 3: Zero-inflated EM Algorithm

```
initialize model parameters φ, μ, σ², λ;
initialize log likelihood ℓ and Δℓ = ∞;
while Δℓ > Δ_0 do
    E-step: given φ, μ, σ, λ, compute E[X_ij|Z_i = k], E[X_ij²|Z_i = k], w_ik;
    M-step: compute analytic updates for φ, μ, σ² and optimize λ numerically;
    compute new ℓ;
end
```

We now describe the full details of the algorithm. In the expressions below, $i = 1, ..., N$ indexes samples, $j = 1, ..., D$ indexes genes, and $k = 1, ..., K$ indexes clusters.

3.1.1 Initialization of parameters

We initialize the cluster weights ϕ, the cluster means μ, and the cluster variances σ^2 using a GMM. We first impute zero values of Y by replacing $Y_{ij} = 0$ with a random draw from the nonzero values of the jth column of Y; then we fit a GMM to the imputed Y, and set ϕ, μ, and σ^2 to the model parameters. To initialize λ, for each gene j, we compute the mean of nonzero samples μ and the fraction of samples that are zero, p_0; we then fit a decaying squared exponential to the pairs (μ, p_0), and set λ to the fitted exponential parameter.

3.1.2 Optimization of parameters

As before, we maximize

$$Q(\Theta_n) = \sum_i \sum_{Z_i} \int_{X_i} \sum_{H_i} P(Z_i, H_i, X_i | Y_i, \Theta_{n-1}) \log P(Z_i, H_i, X_i, Y_i | \Theta_n) dX_i \tag{3.1}$$

We first derive an expression for the complete data log likelihood, $\log P(Z_i, H_i, X_i, Y_i|\Theta)$. We have

$$\log P(Z_i, H_i, X_i, Y_i|\Theta) = \log\Big(P(Z_i)p(X_i|Z_i)p(H_i|X_i)p(Y_i|H_i, X_i)\Big), \tag{3.2}$$

$$= \log\left(\phi_k \prod_j p(X_{ij}|Z_i)p(H_{ij}|X_{ij})p(Y_{ij}|X_{ij}, H_{ij})\right) \tag{3.3}$$

As before, there are two different cases to consider: $Y_{ij} \neq 0, H_{ij} = 0$, and $Y_{ij} = 0, H_{ij} = 1$. We have

$$p(Y_{ij}, X_{ij}, H_{ij}|Z_i = k) = \begin{cases} N(Y_{ij}; \mu_{jk}, \sigma^2_{jk})(1 - \exp(-\lambda Y^2_{ij})) & \text{if } Y_{ij} \neq 0, H_{ij} = 0 \\ N(X_{ij}; \mu_{jk}, \sigma^2_{jk}) \exp(-\lambda X^2_{ij}) & \text{if } Y_{ij} = 0, H_{ij} = 1 \end{cases}$$

Substituting these expressions into (3.3) yields the expression for the complete log likelihood

$$\log P(Z_i, H_i, X_i, Y_i|\Theta) = \log \phi_k + \sum_{j:Y_{ij}\neq 0} \log \frac{1}{\sqrt{2\pi\sigma^2_{jk}}} - \frac{(Y_{ij} - \mu_{jk})^2}{2\sigma^2_{jk}} + \log(1 - e^{-\lambda Y^2_{ij}})$$
$$+ \sum_{j:Y_{ij}=0} \log \frac{1}{\sqrt{2\pi\sigma^2_{jk}}} - \frac{(X_{ij} - \mu_{jk})^2}{2\sigma^2_{jk}} - \lambda X^2_{ij} \tag{3.4}$$

3.1.2.1 E-step

The only terms in $\log P(Z_i, H_i, X_i, Y_i|\Theta)$ that depend on a latent variable are contained in the second sum in (3.4). We need to compute $E[X^2_{ij}]$ and $E[X_{ij}]$ for $Y_{ij} = 0$. H_{ij} is determined by Y_{ij}, so the expectation is just over X_{ij} and Z_i. Define

$$w_{ik} = p(Z_i = k|Y_i) \tag{3.5}$$

Then we have

$$E[X_{ij}] = \sum_{Z_i=k} w_{ik} E[X_{ij}|Z_i = k], \tag{3.6}$$

$$E[X_{ij}|Z_i = k] = \frac{1}{C} N(X_{ij}|u_{jk}, \sigma^2_{jk}) \exp(-\lambda X^2_{ij}) X_{ij} dX_{ij} \tag{3.7}$$

and similarly for X^2_{ij}; C is a normalization constant so that the probability distribution integrates to one. Note that this expression is not dependent on the index i, increasing computational tractability. We derive closed forms for $E[X_{ij}|Z_i = k]$ and $E[X^2_{ij}|Z_i = k]$ in Appendix B. We give two options for computing the expectations: either using integral bounds of $(0, \infty)$ for X_{ij} or using $(-\infty, \infty)$. There are arguments to be made for both bounds: on the one hand, bounds of $(0, \infty)$ are more biologically realistic, because gene expression values cannot be negative; on the other hand, bounds of $(-\infty, \infty)$ are implied by the generative model. We note that using bounds of $(0, \infty)$ is *not* equivalent to using a truncated normal in the generative model, as that would also change the likelihood function. Given that most clusters are centered fairly far from zero, there is unlikely to be much difference between the two bounds; we test the algorithm using bounds of $(0, \infty)$.

Given $E[X_{ij}|Z_i = k]$ and $E[X^2_{ij}|Z_i = k]$, our expression for Q is

$$\sum_{i=1}^{N} \sum_{Z_i=k} w_{ik} \Bigg(\log \phi_k + \sum_{j:Y_{ij} \neq 0} \log \frac{1}{\sqrt{2\pi\sigma^2_{jk}}} - \frac{(Y_{ij} - \mu_{jk})^2}{2\sigma^2_{jk}} + \log(1 - \exp(-\lambda Y^2_{ij})) + \sum_{j:Y_{ij}=0} \log \frac{1}{\sqrt{2\pi\sigma^2_{jk}}} - \frac{E[X^2_{ij}|Z_i = k] - 2\mu_{jk} E[X_{ij}|Z_i = k] + \mu^2_{jk}}{2\sigma^2_{jk}} - \lambda E[X^2_{ij}|Z_i = k] \Bigg) \tag{3.8}$$

3.1.2.2 M-step

In the M-step, we solve $\frac{dQ}{d\Theta} = 0$, maximizing with respect to $\phi_k, \lambda, \mu_{jk}$, and σ_{jk}^2. The expression involving ϕ_k is identical to that in the standard GMM; thus, we have

$$\phi_k \propto \sum_{i=1}^{N} w_{ik} \tag{3.9}$$

We note that, if desired, a Dirichlet prior (Görür and Rasmussen, 2010) can be placed upon the weights ϕ_k: if the prior parameter is α, this corresponds to setting $\phi_k \propto \sum_{i=1}^{N} w_{ik} + \alpha$. This is sometimes useful for preventing empty clusters from being formed. The other update equations are left unchanged.

To optimize μ_{jk}, we compute

$$\frac{dQ}{d\mu_{jk}} = \frac{d}{d\mu_{jk}}\left(\sum_{i:Y_{ij}\neq 0} -w_{ik}\frac{(Y_{ij}-\mu_{jk})^2}{2\sigma_{jk}^2} + \sum_{i:Y_{ij}=0} w_{ik}\frac{2\mu_{jk}E[X_{ij}|Z_i=k]-\mu_{jk}^2}{2\sigma_{jk}^2}\right), \tag{3.10}$$

$$= \sum_{i:Y_{ij}\neq 0} w_{ik}\frac{Y_{ij}-\mu_{jk}}{\sigma_{jk}^2} + \sum_{i:Y_{ij}=0} w_{ik}\frac{E[X_{ij}|Z_i=k]-\mu_{jk}}{\sigma_{jk}^2} \tag{3.11}$$

which yields

$$\mu_{jk} = \frac{\sum_{i:Y_{ij}\neq 0} w_{ik}Y_{ij} + \sum_{i:Y_{ij}=0} w_{ik}E[X_{ij}|Z_i=k]}{\sum_{i=1}^{N} w_{ik}} \tag{3.12}$$

To optimize σ_{jk}^2, we define $v_{jk} = \frac{1}{\sigma_{jk}^2}$ and $\Delta_{ijk} = (E[X_{ij}^2|Z_i=k] - 2\mu_{jk}E[X_{ij}|Z_i=k] + \mu_{jk}^2)$. Then we have

$$\frac{dQ}{dv_{jk}} = \frac{d}{dv_{jk}}\left(\sum_{i:Y_{ij}\neq 0} w_{ik}\left(\log v_{jk} - v_{jk}(Y_{ij}-\mu_{jk})^2\right) + \sum_{i:Y_{ij}=0} w_{ik}\left(\log v_{jk} - v_{jk}\Delta_{ijk}\right)\right), \tag{3.13}$$

$$= \left(\sum_{i:Y_{ij}\neq 0} w_{ik}\left(\frac{1}{v_{jk}} - (Y_{ij}-\mu_{jk})^2\right) + \sum_{i:Y_{ij}=0} w_{ik}\left(\frac{1}{v_{jk}} - \Delta_{ijk}\right)\right) \tag{3.14}$$

which yields

$$\sigma_{jk}^2 = \frac{\sum_{i:Y_{ij}\neq 0} w_{ik}(Y_{ij} - \mu_{jk})^2 + \sum_{i:Y_{ij}=0} w_{ik}\Delta_{ijk}}{\sum_{i=1}^{N} w_{ik}} \tag{3.15}$$

The updates for both μ_{jk} and σ_{jk}^2 are what we would expect: they are the weighted average of the means and variances of the values computed in the E-step. We note that the second derivatives $\frac{d^2Q}{dv_{jk}^2}$ and $\frac{d^2Q}{d\mu_{jk}^2}$ are always negative, implying that setting the first derivatives to zero identifies maxima, not minima. As before, we optimize λ numerically.

3.2 Performance in simulations

In Figures 3.1-3, we compare the performance of our algorithm to GMM and K-means. Simulated data is drawn using the generative mixture model described in section 1.3.1 with zero model $f(X_{ij}) = \exp(-\lambda X_{ij}^2)$ (Figure 3.1), $f(X_{ij}) = 1 - \lambda X_{ij}$ (Figure 3.2), and $f(X_{ij}) = 1 - \lambda$ (Figure 3.3). Algorithm performance is evaluated, as before, using the adjusted Rand score of the estimated clustering relative to the true clustering. ZIMM outperformed GMM and K-means regardless of what zero model we used. In the limit of large λ (for which there were very few zeros) or large D (for which clusters were clearly separated) all algorithms performed similarly well. In the limit of large σ (for which clusters overlapped) all algorithms performed badly.

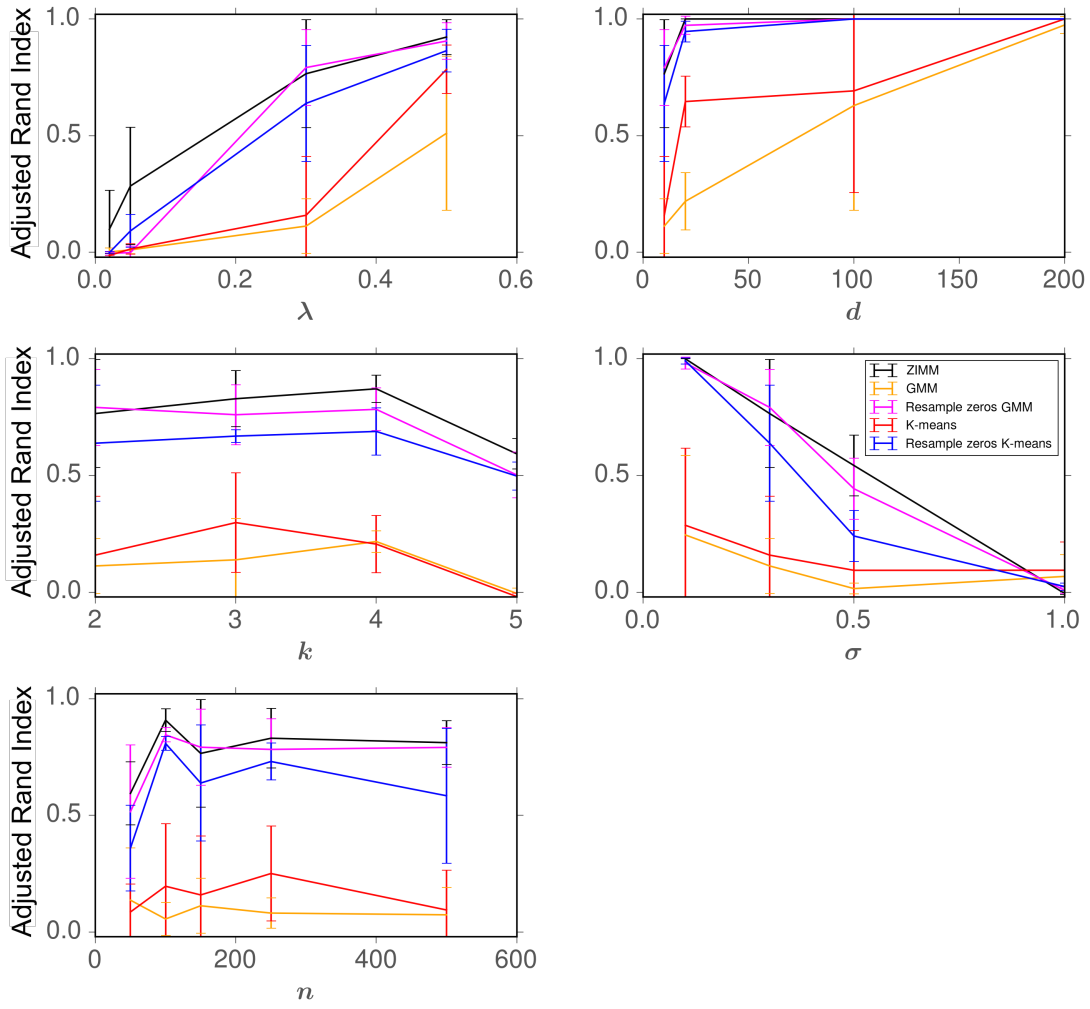


Figure 3.1: Performance of clustering algorithms on simulated data that is zeroed out with exponentially decaying probability $\exp(-\lambda X^2)$.

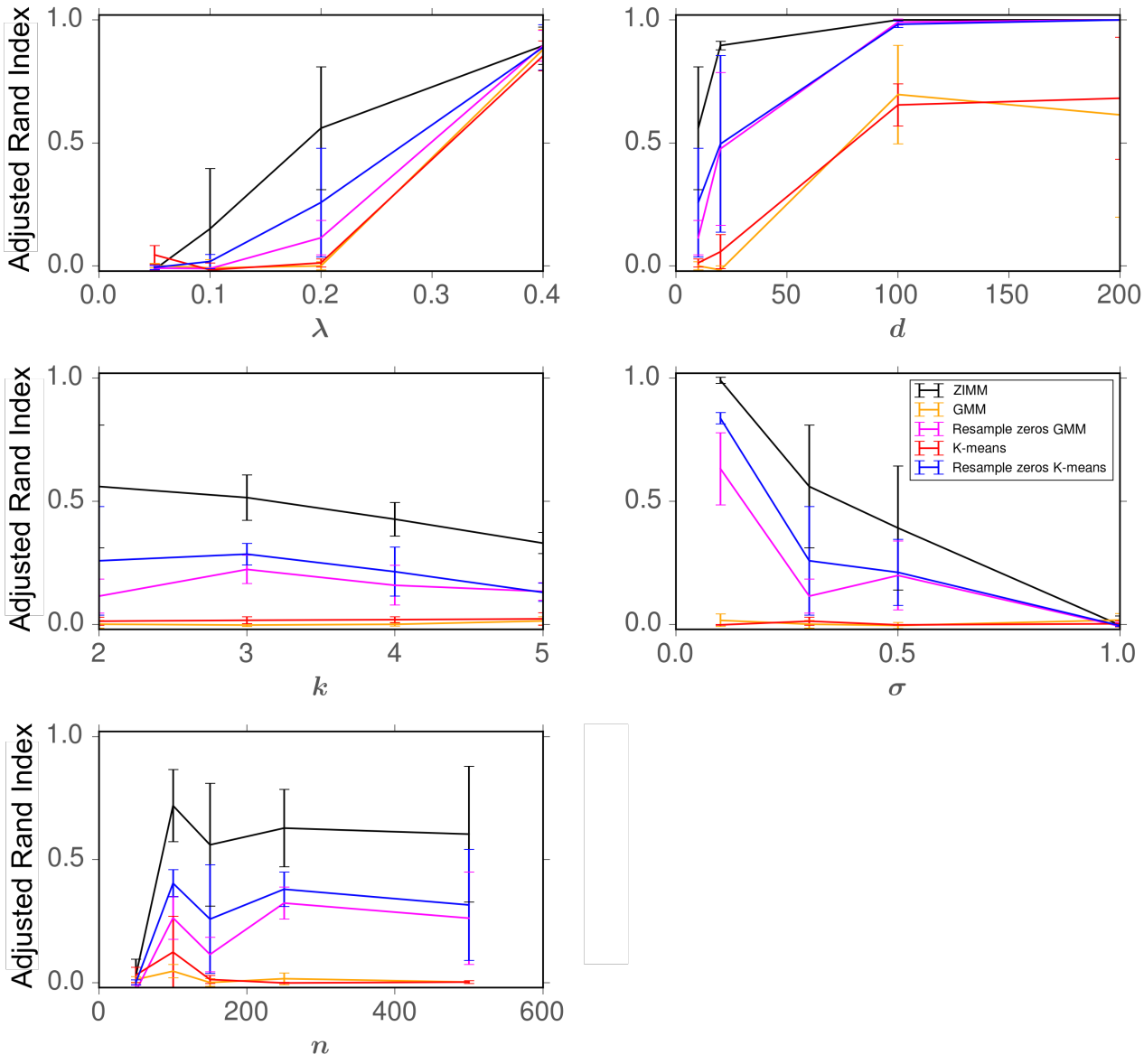


Figure 3.2: Performance of clustering algorithms on simulated data that is zeroed out with linearly decaying probability $1 - \lambda X$.

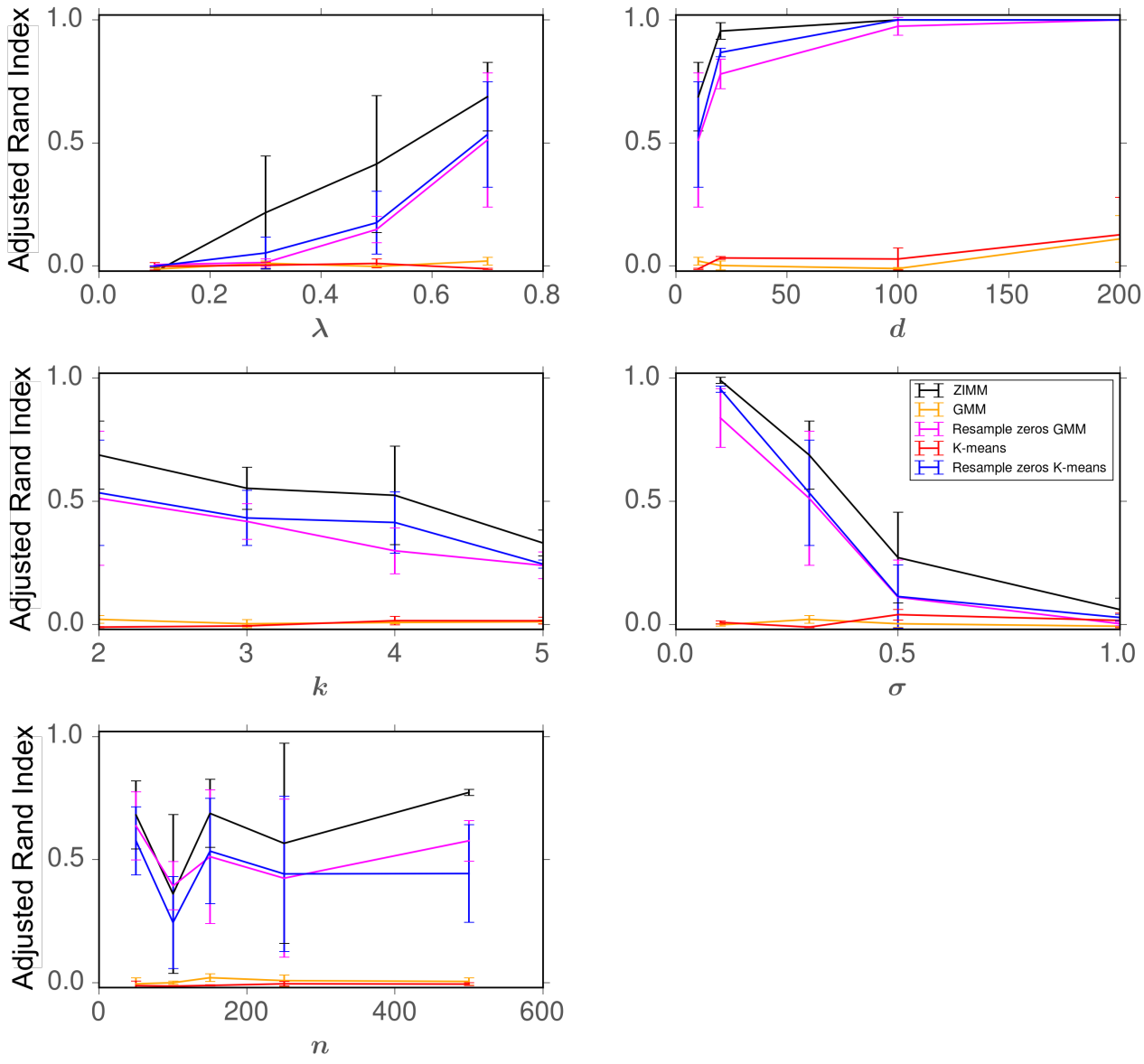


Figure 3.3: Performance of clustering algorithms on simulated data that is zeroed out with uniform probability $1 - \lambda$.

3.3 Performance on biological datasets

Having shown that ZIMM outperforms GMMs on biologically realistic simulated datasets, we show that ZIMM outperforms GMMs on five biological datasets, using a similar procedure as for ZIFA (Section 2.3). For each dataset we selected a subset of 25, 100, 250, or 1000 genes at random, filtered out genes that were zero in more than 95% of samples, and used the histogram divergence metric (using a held-out test set) described in Section 2.3.1. To increase the stability of the algorithm, we placed a Dirichlet prior (Görür and Rasmussen, 2010) on ϕ with $\alpha = 1$ (a standard choice); this prevented empty clusters from being formed, which was otherwise a problem for large numbers of genes. We used two clusters for all datasets because the true number of clusters was unknown. We averaged performance across 100 replicates. ZIMM yielded better fits than GMMs to at least 75% of genes on all five datasets (Table 3.1).

Dataset	**25 Genes**	**100 Genes**	**250 Genes**	**1000 Genes**
Differentiating T cells	$80 \pm 7.6\%$	$79 \pm 4.3\%$	$80 \pm 3.4\%$	$80 \pm 3.0\%$
11 Populations	$93 \pm 5.7\%$	$92 \pm 2.9\%$	$94 \pm 2.1\%$	$94 \pm 2.3\%$
Myoblasts	$95 \pm 4.5\%$	$94 \pm 3.3\%$	$93 \pm 2.7\%$	$94 \pm 1.9\%$
CML	$82 \pm 7.4\%$	$80 \pm 4.1\%$	$80 \pm 3.0\%$	$80 \pm 1.8\%$
Bone Marrow	$98 \pm 2.8\%$	$97 \pm 1.7\%$	$97 \pm 1.3\%$	$96 \pm 0.9\%$

Table 3.1: Comparison of ZIMM to GMMs on five biological datasets. Columns are the number of genes in the dataset (selected at random). Percentages denote the proportion of genes for which ZIMM provided a better fit than GMMs, averaged across 100 replicates.

Chapter 4

Broader Applicability and Future Work

While ZIFA and ZIMM were developed for single-cell data, they are likely to have broader applicability given the prevalence of zero-inflated datasets and the general usefulness of clustering and dimensionality reduction. Ideally the data Y – with rows samples and columns features – would have a few properties.

1. Rows identically and independently distributed.

2. Multiple columns to allow dimensionality reduction.

3. Zero-inflated but not too sparse: the methods have not been tested on datasets that are almost entirely zeros and ZIFA is at present only computationally tractable for several thousand samples and features. Even if the full dataset is quite sparse, applying filters to rows or columns will yield a less sparse dataset.

4. A negative correlation between the mean value of nonzero entries in a column, μ, and the fraction of a column that is zero, p_0; ideally, this would be reasonably well fit by $p_0 = \exp(-\lambda\mu^2)$.

The final requirement is hardest to satisfy. Our simulations imply that it may suffice merely to have a negative correlation between μ and p_0, not the exact relationship

$p_0 = \exp(-\lambda\mu^2)$, because ZIFA and ZIMM improve on standard models even when, for example, the relationship between μ and p_0 is linear. Previous research into zero-inflation regression models has also explored the possibility of a relationship between the mean value μ and the probability of zero p_0 (Lambert, 1992). For example, zero-inflated Poisson regression models, which model the distribution of a discrete variable with an excess of zeros, write the probability of a zero and the nonzero mean as functions of the covariates. (Famoye and Singh, 2006) discusses a model in which the mean for the ith datapoint μ_i and probability of a zero p_{0i} are given by

$$\log(\mu_i) = \sum_{j=1}^{k} x_{ij}\beta_j \tag{4.1}$$

$$\log\left(\frac{p_{0i}}{1-p_{0i}}\right) = -\tau \sum_{i=1}^{k} x_{ij}\beta_j \tag{4.2}$$

which, for $\tau > 0$, will induce a negative correlation between μ and p_0. The basic assumption of a negative correlation has thus been the subject of previous research into zero-inflated models. These models, however, are designed for modeling a one-dimensional variable as a function of covariates, which is a very different problem than what we consider here. Additionally, we found that the logistic parameterization of p_0 (Eq. 4.2) generally did not yield tractable expressions when combined with Gaussian probability distributions.

4.1 Transforming Y

To improve the fit $p_0 = \exp(-\lambda\mu^2)$, we also explored transforming Y as follows. Let $f(Y_{ij};\Theta)$ be some function of Y parameterized by Θ with $f(0) = 0$. For example, f might be a quadratic polynomial function of Y. Then we can choose Θ such that $p_0 = \exp(-\lambda f(Y;\Theta)^2)$ is a good fit. To assess performance of this method, we generated data using the factor analysis model and zeroed it out using functions other than $exp(-\lambda X_{ij}^2)$ (Figures 4.1 and 4.2).

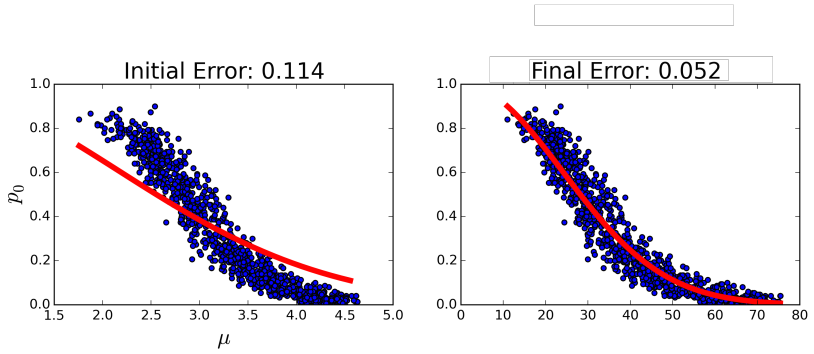


Figure 4.1: Simulated data generated with $p(H_{ij} = 1) = \exp(-.02X_{ij}^4)$ is initially poorly fit by the decaying squared exponential. Rescaling the data using a quadratic polynomial f yields a much better fit.

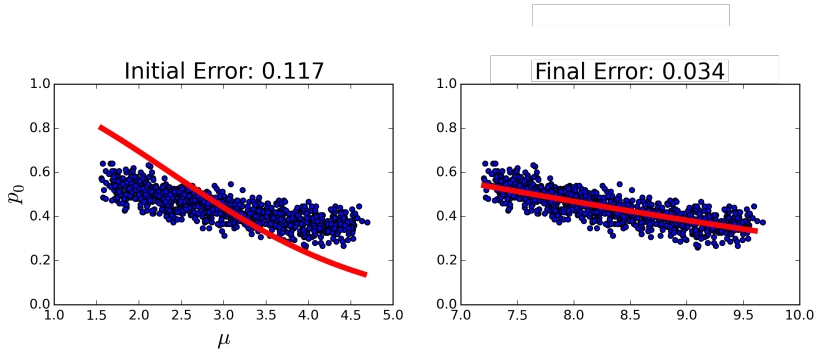


Figure 4.2: Quadratic rescaling improves the fit to $p(H_{ij} = 1) = \exp(-.5\sqrt{X_{ij}})$.

However, while this rescaling method improved the fit between p_0 and $\exp(-\lambda\mu^2)$, it often failed to improve the correlation between the true and estimated latent distances between points when used in conjunction with the full ZIFA model on simulated data, so we do not recommend its general use. The exception was if the correlation between p_0 and μ was *positive*, in which case using the rescaling method *did* dramatically improve performance. This is unsurprising, because a positive correlation between p_0 and μ indicates that the $p_0 = \exp(-\lambda\mu^2)$ model is seriously misspecified.

4.2 Application of models to other datasets

We now discuss several classes of datasets on which ZIFA and ZIMM might be useful.

4.2.1 Recommendation datasets

Let Y be a recommendation matrix such that Y_{ij} is user i's rating of item j, with $Y_{ij} = 0$ if i has not rated item j. Then we might expect there to be a positive relationship between an item's mean rating and how many people rate it because more people will use, and therefore rate, an item with high ratings. This will produce a negative correlation between μ and p_0, as desired[1].

We took all reviews for all restaurants in Las Vegas in the Yelp Challenge Dataset (Yelp, 2015) and examined restaurants in each price category individually. We found highly statistically significant negative correlations ($r = 0.2 - 0.3$) between μ and p_0 that were reasonably well-fit by $p_0 = \exp(-\lambda\mu^2)$, although this broke down for very highly rated restaurants, which were rare; we found a similarly strong negative correlation in the MovieLens dataset (GroupLens, 2015) of movie ratings (Figure 4.3).

Fitting such datasets is a problem of interest, the Netflix Challenge (Bell and Koren, 2007) offering one high-profile example, but previous latent factor models have often treated zeros as uninformative (Leskovec et al., 2014). Using our models incorporates the fact that zeros are informative: if a user has not used (and therefore rated) an item, it indicates they may like it slightly less.

[1]This effect will only occur if items are roughly comparable. For example, if items vary in price, we might imagine that very highly rated items are very highly priced, and therefore receive fewer ratings. This may be one reason we found no significant correlation between μ and p_0 in an Amazon Food Reviews dataset (Leskovec and Krevl, 2014). But one can imagine many datasets in which items are comparable: movies, books, or restaurants stratified by price.

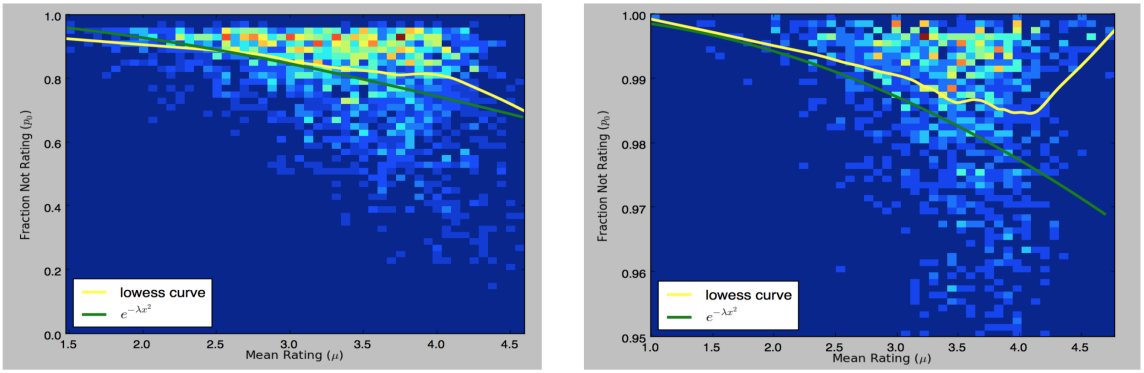


Figure 4.3: **Left**: Negative correlation between μ and p_0 in the MovieLens-1M Dataset for 1,589 reviewers rating 2,019 movies for a total of 612,632 reviews. (We filtered for reviewers with at least 200 reviews and movies with at least 100). **Right**: Negative correlation between μ and p_0 in the Yelp Challenge Dataset for 2,603 reviewers rating 1,250 restaurants for a total of 55,810 reviews. (We filtered for restaurants at price level 2, reviewers with at least 10 reviews and restaurants with at least 20). We confirmed that we observed strong negative correlations for a range of thresholds.

As an illustration of the utility of ZIFA on recommendation datasets, we applied ZIFA to the Yelp Challenge Dataset and computed the matrix $C = AA^T + W$ of correlations between restaurant ratings on restaurants at price level 2 (filtering for reviewers who had provided at least 20 reviews and restaurants with 100 reviews to yield a matrix of tractable size). We used affinity propagation (Frey and Dueck, 2007) to cluster the correlation matrix and found that the clustering correlated with properties of the restaurants (categorical f-test; Table 4.1). For example, we found a cluster with a higher proportion of restaurants classified as "Japanese" or "Sushi", a cluster with a higher proportion of restaurants classified as "pub" or "bar", and a cluster with a higher proportion of restaurants classified as "traditional American".

Restaurants in Cluster	**Sushi**	**Japanese**	**Traditional American**	**Pub**	**Bar**
145	17%	**18**%	3%	1%	9%
42	0%	12%	**36**%	0%	2%
34	3%	0%	9%	9%	**27**%

Table 4.1: Clusters of restaurants in the Yelp Challenge Dataset selected via ZIFA and affinity propagation (Frey and Dueck, 2007) on the estimated covariance matrix.

4.2.2 Usage frequency datasets

Let Y_{ij} denote the frequency with which person i uses product j: then we might expect there to be a positive correlation between the number of people who use a product and the frequency with which they use it, producing a negative correlation between μ and p_0. For example, we studied a dataset on drug usage (NSDUH, 2012), where Y_{ij} denoted the number of days per year that person i used drug j, and found that drugs which were used by a large fraction of the drug-using population (alcohol, marijuana) were also used more frequently by people who used them (Figure 4.4), producing a negative relationship between μ and p_0. We applied ZIMM with two clusters and identified one cluster of people who use drugs less frequently and one cluster that used drugs more frequently. Using ZIFA to estimate the drug-drug correlation matrix (Figure 4.5) revealed that all drugs showed positive correlations in their usage patterns.

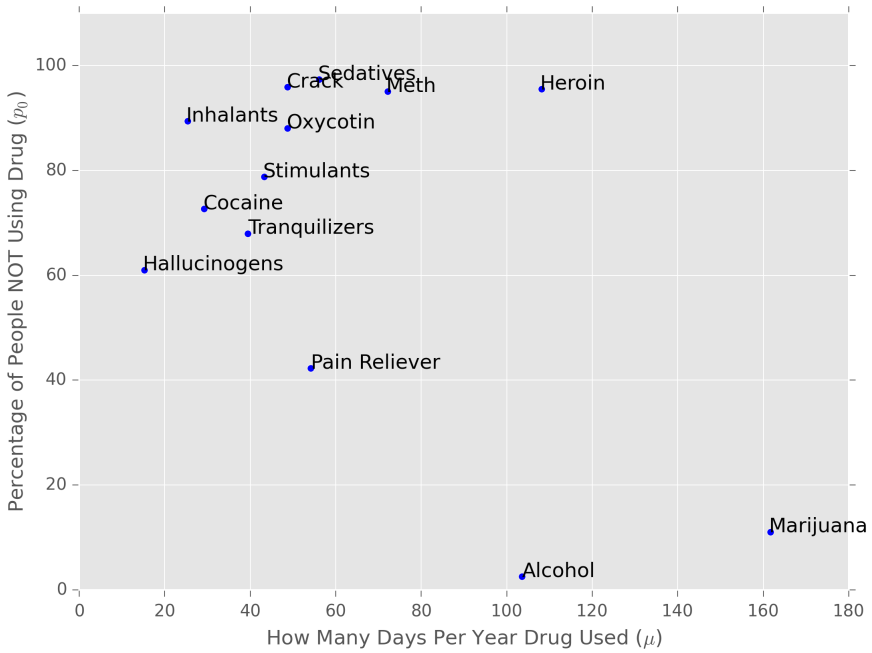


Figure 4.4: Negative correlation between how often a drug is used by people who use it (μ) and the fraction of the population never using the drug (p_0) among people who used at least three drugs.

One might imagine that other usage frequency datasets would show a similar negative

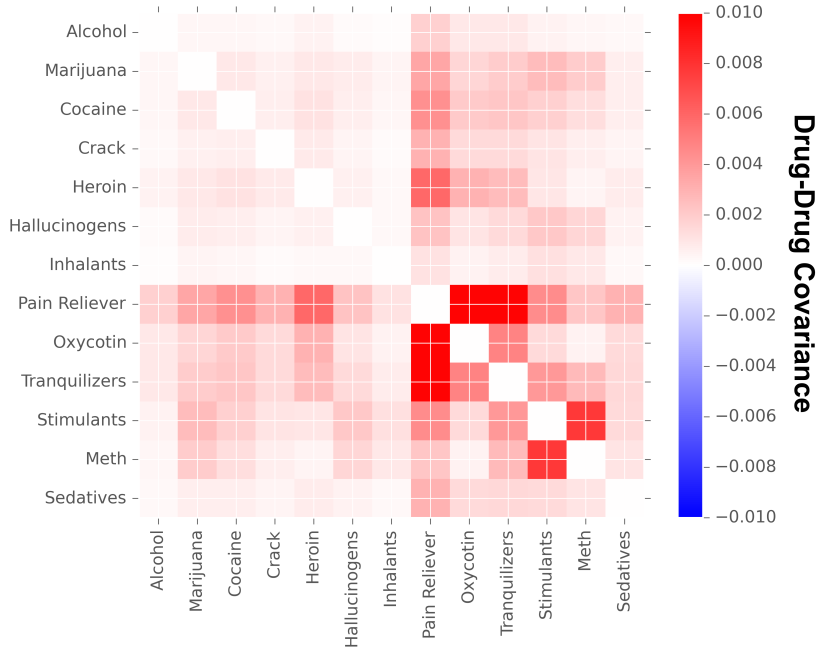


Figure 4.5: Estimation of the drug-drug covariance matrix $AA^T + W$ using ZIFA.

correlation between μ and p_0: for example, perhaps Y_{ij} is the frequency with which person i interacts with website j, and people interact more frequently with websites that are more popular. Similarly, if Y_{ij} is the quantity of item j that person i has purchased, with $Y_{ij} = 0$ if i has not purchased item j, we might imagine there is a "diamonds and eggs" effect, where items like diamonds are purchased both a) by fewer customers and b) in smaller quantities when they are purchased. (Note that in all these cases, zeros are real values, not censored ones).

4.2.3 Network datasets

Let Y_{ij} be a weighted measure of the strength of the connection between nodes i and j in a network. Then Y_{ij} is likely to be zero-inflated in real-world networks because most pairs of nodes are not connected. The requirement of a negative correlation between μ and p_0 corresponds to a positive correlation between a node's degree and the mean weight of edges to that node. This correlation is in fact observed in real networks frequently

enough that theoretical models have been proposed to explain it (Barrat et al., 2004b; Ramasco and Gonçalves, 2007). Examples include the airport traffic network (Barrat et al., 2004a) where nodes represent airports and edge weights represent the number of available seats on flights between two airports. (Barrat et al. (2004a) also discusses the scientific coauthorship network, where the correlation between degree and mean edge weight is not observed, so the correlation is not a ubiquitous property.) ZIFA and ZIMM thus might be applied to network datasets – for example, to identify clusters of interconnected nodes.

4.3 Discussion and future work

In this work, we have presented two new algorithms for single-cell gene expression data: a zero-inflated factor analysis model (ZIFA), which performs dimensionality reduction, and a zero-inflated mixture model (ZIMM) which performs clustering. Both these models extend conventional statistical models (factor analysis and the Gaussian Mixture Model, respectively) by assuming that the true data X is zeroed out with probability $\exp(-\lambda X^2)$. We show that these models outperform conventional models on both simulated and biological datasets. We discuss several other classes of datasets on which our zero-inflated models might be useful.

Further work is possible in several areas:

- One limitation of ZIFA is that it is at present far slower than PCA or other dimensionality-reduction algorithms. It is still computationally tractable (runtime several minutes on a laptop) for up to several thousand features and samples, and it can be parallelized easily because computations for each sample are independent. Nonetheless, it is possible that further mathematical or computational optimizations could decrease the runtime of the algorithm considerably. This would be especially useful for scaling to very sparse non-biological datasets.

- ZIMM has less obvious applicability to biological datasets than ZIFA because many biological datasets are very high-dimensional and dimensionality reduction should

be performed as a preprocessing step. It is possible to identify datasets (perhaps with targeted gene sets) on which ZIMM would be more useful?

- A natural combination of ZIFA and ZIMM would be a zero-inflated extension of mixtures of PPCAs (Tipping and Bishop, 1999a), although this seems likely to be more difficult to optimize because of the more complex model.

- The algorithms could be used for further exploratory analysis of new biological or non-biological datasets. For example, one might expect that our algorithms, by better modeling biological data, would enable better detection of subpopulations of interest. (We have already explored this avenue on several datasets, and have found that our dimensionality reduction algorithm does not more clearly separate clusters of cells other researchers have identified as distinct. This is in itself somewhat interesting, because if modeling expression data more faithfully fails to more clearly separate groups, it may indicate that the groups are not clearly differentiated in the expression data.)

To our knowledge, this is the first time that zero-inflated clustering or dimensionality reduction algorithms have been proposed. We are hopeful that our methods will prove useful both to the single-cell community and to a broader statistical audience. To this end, we provide optimized Python implementations of both methods at github.com/epierson9/ZIFA and github.com/epierson9/ZIMM; we have also described ZIFA in a paper which is available online (Yau and Pierson, 2015).

Appendix A

Fast Matrix Inversions for Dimensionality Reduction

In the E-step of zero-inflated dimensionality reduction, our goal is to compute the mean and covariance for the distribution $p(Z_i, X_{i0}|Y_i)$. These are given by

$$(\Sigma_c^{-1} + 2\lambda I_x)^{-1}\Sigma_c^{-1}\mu_c \tag{A.1}$$

$$(\Sigma_c^{-1} + 2\lambda I_x)^{-1} \tag{A.2}$$

These computations are potentially expensive because they involve inverting $D \times D$ matrices, and matrix inversion has cubic complexity. We show that it is possible to evaluate these expressions without computing the inverse of any $D \times D$ matrix, which greatly decreases the runtime of our algorithm. We first simplify $(\Sigma_c^{-1} + 2\lambda I_x)^{-1}$ using the Kailath variant of the Woodbury identity (Petersen et al., 2008):

$$(A + BC)^{-1} = A^{-1} - A^{-1}(B^{-1} + CA^{-1}B)^{-1}CA^{-1} \tag{A.3}$$

with $A = \Sigma_c^{-1}$, $B = 2\lambda I$, $C = I_x$. This yields

$$(\Sigma_c^{-1} + 2\lambda I_x)^{-1} = \Sigma_c - 2\lambda\Sigma_c(I + 2\lambda I_x\Sigma_c)^{-1}I_x\Sigma_c \tag{A.4}$$

Thus, we can write the mean and covariance of $p(Z_i, X_{i0}|Y_i)$ as

$$(I - 2\lambda\Sigma_c(I + 2\lambda I_x\Sigma_c)^{-1}I_x)\mu_c \tag{A.5}$$

$$\Sigma_c - 2\lambda\Sigma_c(I + 2\lambda I_x\Sigma_c)^{-1}I_x\Sigma_c \tag{A.6}$$

Both these expressions contain the inverse of $(I + 2\lambda I_x\Sigma_c)$. We use the block matrix inversion formula (Petersen et al., 2008) to compute this, which yields

$$(I + 2\lambda I_x\Sigma_c)^{-1} = \begin{pmatrix} I & 0 \\ -(I + 2\lambda\Sigma_{xx})^{-1}2\lambda\Sigma_{xz} & (I + 2\lambda\Sigma_{xx})^{-1} \end{pmatrix} \tag{A.7}$$

where Σ_{xx}, Σ_{xz} correspond to the sub matrices of Σ_c corresponding to the X and Z indices. Thus, we need to compute the inverse of $I + 2\lambda\Sigma_{xx}$. Given the formula for Σ_c (2.14), we have

$$(I + 2\lambda\Sigma_{xx}) = I + 2\lambda(\Sigma_{00}^{(p)} - \Sigma_{0+}^{(p)}\Sigma_{++}^{(p)\ -1}\Sigma_{+0}^{(p)}) \tag{A.8}$$

$$= 2\lambda(\Sigma'_{00} - \Sigma_{0+}^{(p)}\Sigma_{++}^{(p)\ -1}\Sigma_{+0}^{(p)}) \tag{A.9}$$

where the 0 and + subscript denote the indices of X_i which correspond to zero and nonzero entries in Y_i, respectively, and $\Sigma'_{00} = \Sigma_{00}^{(p)} + \frac{1}{2\lambda}I$. We can invert this expression by applying another form of the Woodbury formula:

$$(D + CBC^T)^{-1} = D^{-1} - D^{-1}C(B^{-1} + C^TD^{-1}C)^{-1}C^TD^{-1} \tag{A.10}$$

with $D = \Sigma'_{00}, B = -\Sigma_{++}^{-1}, C = \Sigma_{0+}$. This yields

$$(\Sigma'_{00} - \Sigma^{(p)}_{0+}\Sigma^{(p)\ -1}_{++}\Sigma^{(p)}_{+0})^{-1} = \Sigma'^{\ -1}_{00} - \Sigma'^{\ -1}_{00}\Sigma^{(p)}_{0+}(-\Sigma^{(p)}_{++} + \Sigma^{(p)}_{+0}\Sigma'^{\ -1}_{00}\Sigma^{(p)}_{0+})^{-1}\Sigma^{(p)}_{+0}\Sigma'^{\ -1}_{00} \tag{A.11}$$

$$= \Sigma'^{\ -1}_{00} - \Sigma'^{\ -1}_{00}\Sigma^{(p)}_{0+}(-\Sigma^{(p)}_{++} + A_+(A_0^T\Sigma'^{\ -1}_{00}A_0)A_+^T)^{-1}\Sigma^{(p)}_{+0}\Sigma'^{\ -1}_{00} \tag{A.12}$$

where A_0 and A_+ denote the rows of A corresponding to zero and nonzero entries of Y_i, respectively. We can compute the inverse $(-\Sigma^{(p)}_{++} + A_+(A_0^T\Sigma'^{\ -1}_{00}A_0)A_+^T)^{-1}$ by applying the Woodbury formula for what I promise is the final time with $D = -\Sigma^{(p)}_{++}$, $B = A_0^T\Sigma'^{\ -1}_{00}A_0$, $C = A_+$. Note that B will be $k \times k$, making it easy to invert. We can invert Σ'_{00} and all $d \times d$ matrices with the form $AA^T + D$ (where A is skinny and D is diagonal) by noting that

$$(AA^T + D)^{-1} = D^{-1} - D^{-1}A(I + A^TD^{-1}A)^{-1}A^TD^{-1} \tag{A.13}$$

which will be much faster because the inverse $(I + A^TD^{-1}A)$ is $k \times k$ rather than $d \times d$. These simplifications allow us to avoid computing any large inverses, making the most expensive step in the algorithm the multiplications of $d \times d$ matrices. As a final computational speedup, we note that in many cases these multiplications can be avoided because the $d \times d$ matrices are products of two $d \times k$ matrices, producing large decreases in runtime if an optimal multiplication order is used.

Appendix B

Closed Forms for Mixture Model Integrals

Here we derive expressions for the integrals introduced in the E-step in the zero-inflated mixture model. We compute expressions both for integral bounds of $(0, \infty)$ and $(-\infty, \infty)$. First we compute the normalization constant C. For integral bounds of $(0, \infty)$ we have:

$$C = \int_0^\infty \exp\left(- \frac{(x - \mu_{jk})^2}{2\sigma_{jk}^2} \right) \exp(-\lambda x^2) dx \tag{B.1}$$

$$= \int_{-b}^\infty D \exp(-ay^2) dy, y = x - b \tag{B.2}$$

$$= D \frac{\sqrt{\pi}(\text{erf}(b\sqrt{a}) + 1)}{2\sqrt{a}} \tag{B.3}$$

with $a = \frac{1+2\sigma_{jk}^2\lambda}{2\sigma_{jk}^2}$, $b = \frac{\mu_{jk}}{2\sigma_{jk}^2 a}$, $D = \exp(-\frac{\mu_{jk}^2}{2\sigma_{jk}^2} + ab^2)$. For integral bounds of $(-\infty, \infty)$ we have

$$C = \int_{-\infty}^\infty D \exp(-ay^2) dy = D\sqrt{\frac{\pi}{a}} \tag{B.4}$$

with a and D defined as before. Now we compute $E[X_{ij}|Z_i = k]$. For integral bounds of $(0, \infty)$ we have:

$$E[X_{ij}|Z_i = k] = \frac{1}{C}\int_0^\infty \exp\left(-\frac{(x-\mu_{jk})^2}{2\sigma_{jk}^2}\right)\exp(-\lambda x^2)x dx \tag{B.5}$$

$$= \frac{1}{C}\int_{-b}^\infty D\exp(-ay^2)(y+b)dy \tag{B.6}$$

$$= \frac{D}{C}\frac{\exp(-ab^2)}{2a} + b \tag{B.7}$$

For integral bounds of $(-\infty, \infty)$ we have

$$E[X_{ij}|Z_i = k] = \frac{1}{C}\int_{-\infty}^\infty D\exp(-ay^2)(y+b)dy = \frac{Db}{C}\sqrt{\frac{\pi}{a}} \tag{B.8}$$

Finally, we compute $E[X_{ij}^2|Z_i = k]$. For integral bounds of $(0, \infty)$ we have:

$$E[X_{ij}^2|Z_i = k] = \frac{1}{C}\int_0^\infty \exp\left(-\frac{(x-\mu_{jk})^2}{2\sigma_{jk}^2}\right)\exp(-\lambda x^2)x^2 dx \tag{B.9}$$

$$= \frac{1}{C}\int_{-b}^\infty D\exp(-ay^2)(y+b)^2 dy \tag{B.10}$$

$$= \frac{1}{C}\int_{-b}^\infty D\exp(-ay^2)(y^2+2yb+b^2)dy \tag{B.11}$$

$$= \frac{1}{C}\int_{-b}^\infty D\exp(-ay^2)y^2 dy + 2b\frac{D}{C}\frac{\exp(-ab^2)}{2a} + b^2 \tag{B.12}$$

$$= \frac{D}{C}\left(\frac{\sqrt{\pi}}{4a^{3/2}} - \frac{\sqrt{\pi}\,\mathrm{erf}(-b\sqrt{a})}{4a^{3/2}} - \frac{b\exp(-ab^2)}{2a}\right) + 2b\frac{D}{C}\frac{\exp(-ab^2)}{2a} + b^2 \tag{B.13}$$

For integral bounds of $(-\infty, \infty)$ we have

$$E[X_{ij}^2|Z_i = k] = \frac{1}{C}\int_{-\infty}^\infty D\exp(-ay^2)(y^2+2yb+b^2)dy = \frac{D}{C}\left(\frac{1}{2}\sqrt{\frac{\pi}{a^3}} + \sqrt{\frac{\pi}{a}}b^2\right) \tag{B.14}$$

We verified, by rerunning all simulations (Section 3.2), that ZIMM outperformed GMMs and K-means regardless of whether we truncated the integrals.

Bibliography

Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004a.

Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. Weighted evolving networks: coupling topology and weight dynamics. *Physical Review Letters*, 92(22):228701, 2004b.

Robert M Bell and Yehuda Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 2015.

Scott Shaobing Chen and Ponani S Gopalakrishnan. Clustering via the Bayesian information criterion with applications in speech recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 645–648. IEEE, 1998.

David B Dahl. Model-based clustering for expression data via a Dirichlet process mixture model. *Bayesian inference for gene expression and proteomics*, pages 201–218, 2006.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

Morris Eaton. *Multivariate Statistics: a Vector Space Approach*. Wiley New York, 1983.

Felix Famoye and Karan P Singh. Zero-inflated generalized Poisson regression model with an application to domestic violence data. *Journal of Data Science*, 4(1):117–130, 2006.

Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

Dilan Görür and Carl Edward Rasmussen. Dirichlet process Gaussian Mixture Models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4): 653–664, 2010.

GroupLens. MovieLens Dataset. `http://grouplens.org/datasets/movielens/`, 2015.

John A Hartigan and Manchek A Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, pages 100–108, 1979.

Daniel Hebenstreit. Methods, challenges and potentials of single cell RNA-seq. *Biology*, 1(3):658–667, 2012.

Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4 (1):44–57, 2008.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, 1985.

Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson. Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Research*, 21(7):1160–1167, 2011.

Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742, 2014.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Diane Lambert. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford Large Network Dataset Collection. `http://snap.stanford.edu/data`, June 2014.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (GTEx) project. *Nature Genetics*, 45(6):580–585, 2013.

Adam Mead. scRNA-seq data from leukemia patients. *Unpublished data*, 2015.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.

NSDUH. National Survey on Drug Use and Health. 2012. URL `http://doi.org/10.3886/ICPSR34933.v2`.

Fatih Ozsolak and Patrice M Milos. RNA sequencing: advances, challenges and opportunities. *Nature Reviews Genetics*, 12(2):87–98, 2011.

Anoop P Patel, Itay Tirosh, John J Trombetta, Alex K Shalek, Shawn M Gillespie, Hiroaki Wakimoto, Daniel P Cahill, Brian V Nahed, William T Curry, Robert L Martuza,

et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, 2014.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The Matrix Cookbook. *Technical University of Denmark*, 450:7–15, 2008.

Emma Pierson, GTEx Consortium, Daphne Koller, Alexis Battle, and Sara Mostafavi. Sharing and specificity of co-expression networks across 35 human tissues. *PLoS Computational Biology*, 2015.

Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, 2014.

Ashley A Powell, AmirAli H Talasaz, Haiyu Zhang, Marc A Coram, Anupama Reddy, Glenn Deng, Melinda L Telli, Ranjana H Advani, Robert W Carlson, Joseph A Mollick, et al. Single cell profiling of circulating tumor cells: transcriptional heterogeneity and diversity from breast cancer cell lines. *PloS One*, 7(5):e33788, 2012.

José J Ramasco and Bruno Gonçalves. Transport on weighted networks: When the correlations are independent of the degree. *Physical Review E*, 76(6):066106, 2007.

Daniel Ramsköld, Shujun Luo, Yu-Chieh Wang, Robin Li, Qiaolin Deng, Omid R Faridani, Gregory A Daniels, Irina Khrebtukova, Jeanne F Loring, Louise C Laurent, et al. Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nature Biotechnology*, 30(8):777–782, 2012.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

Douglas Reynolds. Gaussian Mixture Models. In *Encyclopedia of Biometrics*, pages 659–663. Springer, 2009.

Carl Scheffler. A Derivation of the EM Updates for Finding the Maximum Likelihood Parameter Estimates of the Student's T Distribution. *Technical Note*, 2008.

Jonathan Schug, Winfried-Paul Schuller, Claudia Kappen, J Michael Salbaum, Maja Bucan, and Christian J Stoeckert. Promoter features related to tissue specificity as measured by Shannon entropy. *Genome Biology*, 6(4):R33, 2005.

Alex K Shalek, Rahul Satija, Joe Shuga, John J Trombetta, Dave Gennert, Diana Lu, Peilin Chen, Rona S Gertner, Jellert T Gaublomme, Nir Yosef, et al. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*, 2014.

Ehud Shapiro, Tamir Biezuner, and Sten Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics*, 14(9): 618–630, 2013.

Fuchou Tang, Catalin Barbacioru, Ellen Nordman, Bin Li, Nanlan Xu, Vladimir I Bashkirov, Kaiqin Lao, and M Azim Surani. RNA-Seq analysis to capture the transcriptome landscape of a single cell. *Nature Protocols*, 5(3):516–535, 2010.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Michael Tipping and Christopher Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999a.

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

Julia Tischler and M Azim Surani. Investigating transcriptional states at single-cell-resolution. *Current Opinion in Biotechnology*, 24(1):69–78, 2013.

Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L

Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 2014.

Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature*, 509(7500):371–375, 2014.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Laura J van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin van der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

Adam K White, Michael VanInsberghe, I Petriv, Mani Hamidi, Darek Sikorski, Marco A Marra, James Piret, Samuel Aparicio, and Carl L Hansen. High-throughput microfluidic single-cell RT-qPCR. *Proceedings of the National Academy of Sciences*, 108(34): 13999–14004, 2011.

Frank Wood. The Indian GPA Problem, 2014.

Christopher Yau and Emma Pierson. Dimensionality reduction for zero-inflated single cell gene expression analysis. *bioRxiv*, page 019141, 2015.

Yelp. Yelp Dataset Challenge. `http://www.yelp.com/dataset_challenge`, 2015.

J-H Zhao, LH Philip, and Qibao Jiang. ML Estimation for Factor Analysis: EM or non-EM? *Statistics and Computing*, 18(2):109–123, 2008.