MULTINOMIAL VARIATIONAL AUTOENCODERS CAN RECOVER PRINCIPAL COMPONENTS

Anonymous authors
Paper under double-blind review

1 ABSTRACT

Covariance estimation on high dimensional data is a central challenge across multiple scientific disciplines. Sparse high-dimensional count data frequently encountered in biological applications such as DNA sequencing and proteomics are often well modeled using multinomial logistic-normal models. In many cases these datasets are also compositional, presented item-wise as fractions of a normalized total, necessitated by measurement and instrument constraints. Three key challenges prove limiting in covariance estimation using these models in compositional settings: (1) the computational complexity of inverting high-dimensional covariance matrices, (2) non-exchangability introduced from the summation constraint on multinomial parameters, (3) the irreducibility of the component multinomial logistic-normal distribution that necessitates the use of parameter augmentation, or similar techniques, during inference. We show that a variational autoencoder augmented with a fast Isometric Log-ratio (ILR) transform can address these issues and accurately estimate principal components from multinomially logistic-normal distributed data. This model can be optimized on GPUs and modified to handle mini-batching, with the ability to scale across thousands of dimensions and thousands of samples.

2 INTRODUCTION

Accurately scaling probabilistic inference across high dimensional count data is a major outstanding challenge (1). This problem is particularly pronounced in biological fields where datasets can observe tens of thousands of features across hundreds to millions of samples. One major computational bottleneck with Gaussian distributed data is the inversion of a d dimensional covariance matrix that has a runtime of $O(d^3)$ (2; 3). As a result, probabilistic covariance estimation for high-dimensional data is a computationally challenging problem.

Recent theoretical developments (4) cementing the connection between Variational Autoencoders (5) and Probabilistic Principal Components Analysis (PPCA) (6) holds much promise enabling accurate, scalable estimation of large covariance matrices. Variational autoencoders (VAEs) were originally proposed as a generative model (5), but are now commonly deployed across scientific disciplines, highlighted by contributions in single cell RNA sequencing (7), microbiome modeling (8), protein modeling (9; 10; 11), natural language processing (12) and image processing (5). Following insights that connected regularized linear autoencoders and PCA (13), Lucas et al (4) showed that carefully designed VAEs can recover the weights that are solved by PPCA. A computational advantage of VAEs is that they do not require the inversion of a covariance matrix, and the resulting runtime is $O(ndkT)$ for n samples, d dimensions, k latent dimensions and T epochs. While it has been noted that VAEs may take tens of thousands of epochs to estimate the principal component (14), VAEs are easily parallelizable and can be accelerated with GPUs, presenting an attractive alternative to estimating PCA (13), and the resulting covariance matrix.

In light of these advances, the connection between VAEs and PPCA is currently limited to Gaussian distributed data. Showing that VAEs can recover PPCA on count distributed data is non-trivial due to the non-conjugacy issues between the Logistic Normal distribution and count distributions such as the Multinomial distribution. Furthermore, the parameters of the Multinomial distribution are compositional; they are constrained within the simplex and the resulting covariance matrix is singular and non-invertible (15; 16). Aitchison (17) showed that PCA can be adapted to handle compositional data through the use of the center log-ratio (CLR) transform, which maintains isometry. However, this transformation is not isomorphic, requiring that the resulting log-ratios sum to zero, and as a

result CLR transformed data will produce a singular covariance matrix and rank-deficient principal components. It has been shown that the isometric log ratio (ILR) transform ([18]) satisfies both isomorphism and isometry and can handle this singularity issue ([19]; [20]) while enabling the estimation of full-rank principal components. Here, we show that VAEs augmented with the ILR transform can also obtain principal components learned from PPCA on Multinomially distributed data.

3 Related Work

In the microbiome literature, there have been a number of methods, such as SPEIC-EASI ([21]), SPARCC ([22]), PLNnetwork ([23]), gCoda([24]), and Flashweave ([25]) that have attempted to model ecological networks through the estimation of pairwise microbe interactions, where microbes are aggregated at different taxonomical scales or 'taxa'. Of these tools, only Flashweave can scale across more than thousands of taxa, but it does this by avoiding the estimation of the covariance matrix. Methods that attempt to estimate the covariance matrix can only handle on the order of few thousand. As a consequence, being able to efficiently estimate Multinomial PPCA can be highly useful for exploratory biological analysis. Up to now, there has not been a consensus on defining Multinomial PPCA. A number of studies have proposed utilizing mixture modeling as a proxy for PCA ([26]; [27]; [28]), however these techniques either depend on the Dirchilet distribution whose covariance matrix is not flexible, or breaks permutation invariance ([29]).

Luca et al has previously showed that the following two models can obtain the same Maximum likelihood estimates of principal components $\boldsymbol{W}$

Probabilistic PCA	Linear VAE
$p(\boldsymbol{x}\|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}_d)$	$p(\boldsymbol{x}\|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}_d)$
$p(\boldsymbol{z}) = N(\boldsymbol{0}, \boldsymbol{I}_k)$	$q(\boldsymbol{z}\|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{V}(\boldsymbol{x} - \boldsymbol{\mu}), \boldsymbol{D})$

Here, $p(\boldsymbol{x}|\boldsymbol{z})$ denotes the likelihood of observations $\boldsymbol{x} \in \mathbb{R}^d$ given the latent representation $\boldsymbol{z} \in \mathbb{R}^k$, $p(\boldsymbol{z})$ denotes the prior on $\boldsymbol{z}$ and $q(\boldsymbol{z}|\boldsymbol{x})$ denotes the estimated variational posterior distribution of $\boldsymbol{z}$ given an encoder parameterized by $\boldsymbol{V}$ and diagonal variances $\boldsymbol{D}$. While PPCA is estimated through Expectation Maximization ([6]), Linear VAEs are optimized by maximizing the Evidence Lower Bound (ELBO) given by

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\big[\log p(\boldsymbol{x}|\boldsymbol{z})\big] - KL\big(q(\boldsymbol{z}|\boldsymbol{x})\big|\big|p(\boldsymbol{z})\big)$$

For Linear VAEs with a Gaussian likelihood, the variational posterior distribution $q(\boldsymbol{z}|\boldsymbol{x})$ can be shown to analytically agree with the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ learned from PPCA ([4]). However, deriving this connection for count-based likelihoods such as the Multinomial is complicated due to non-conjugacy issues (Appendix A). This is a major obstacle for many biological applications; multiple works have shown the merits of incorporating count distributions explicitly into the model ([30]; [31]; [32]; [33]). Here, we provide some directions on overcoming this issue.

4 Methods

First, we will redefine Multinomial PPCA with the ILR transform ([18]). Then we will make the connection between Multinomial VAEs and Multinomial PPCA by leveraging insights from the CU sampler ([30]). We will then derive an algorithm to obtain the maximum a posteriori (MAP) estimate for the VAE parameters.

4.1 Probabilistic Multinomial PCA

PPCA can be extended to multinomially distributed data with the following generative model

$$p(\boldsymbol{x}|\boldsymbol{\eta}) = \text{Mult}(\phi(\boldsymbol{\Psi}\boldsymbol{\eta})) \tag{1}$$

$$p(\boldsymbol{\eta}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z}, \sigma^2 \boldsymbol{I}_d) \tag{2}$$

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_k) \tag{3}$$

Here $\boldsymbol{W} \in \mathbb{R}^{d-1 \times k}$ represents the PCA loading matrix, σ^2 is the variance. $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d-1}$ is a fixed contrast matrix where the columns sum to zero and ϕ is the softmax transform. For a single sample,

$\boldsymbol{x} \in \mathbb{N}^d$ are the observed d -dimensional counts, $\boldsymbol{\eta} \in \mathbb{R}^{d-1}$ are the latent logits, $\boldsymbol{z} \in \mathbb{R}^k$ is the latent representation. The term $\phi(\boldsymbol{\Psi}\boldsymbol{\eta})$ is distributed Logistic Normal $\phi(\boldsymbol{\Psi}\boldsymbol{\eta}) \sim \mathcal{LN}(\boldsymbol{W}\boldsymbol{z}, \sigma^2 I)$ as shown by Aitchison (34). Furthermore, $p(\boldsymbol{x}|\boldsymbol{z})$ yields a Multinomial Logistic Normal distribution, which is given by marginalizing out $\boldsymbol{\eta}$

$$\mathcal{MLN}(\boldsymbol{x}|\boldsymbol{z}) = \int_{\boldsymbol{\eta}} p(\boldsymbol{x}|\boldsymbol{\eta})p(\boldsymbol{\eta}|\boldsymbol{z})d\boldsymbol{\eta}$$

This integral is not tractable; as a result, this distribution has no analytical expectation or variance and the probability density function cannot be directly evaluated. There have been multiple attempts to estimate the posterior distribution with MCMC (35; 36; 32; 37), but the complexity of this distribution requires a large number of samples, reducing the scalability of these methods. Variational methods have been developed to estimate the logistical normal distribution, but due to conditional non-conjugacy, these methods often rely on approximations to the ELBO, further complicating estimation (38).

Recently, Silverman et al (30) proposed to use a Laplace approximation to estimate the parameters of the Multinomial Logistic Normal posterior distribution. This approach relies on a two stage optimization procedure on the factorized posterior distribution given by

$$p(\boldsymbol{\eta}, \boldsymbol{z}|\boldsymbol{x}) \propto p(\boldsymbol{\eta}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{\eta}) \tag{4}$$

If $\boldsymbol{\eta}$ can be directly estimated, then conditional independence can be used to factorize the posterior distribution. Since the both probability densities of the Multinomial distribution and Normal distribution are log-concave functions, a global maxima can be obtained (30). We will exploit this to design a new algorithm to obtain the MAP estimate.

4.2 The ILR Transform enforces identifiablity

The softmax function is a shift invariant function, which introduces an identitiability issue that has been addressed by the compositional data analysis community (15; 39). In order to remove the identifiability issues, an isomorphism between the logits $\boldsymbol{\eta}$ and the Multinomial parameters must be maintained. One commonly used solution is to use a degenerate softmax also known as the inverse additive log-ratio (ALR) transform (15) (Appendix B.2). Previous work has suggested that the isometric log ratio transform (ILR) (18; 19) is more suitable for principal components analysis (Appendix B). The ILR and inverse ILR are given as follows

$$\text{ILR}(x) = \boldsymbol{\Psi}^T \log \boldsymbol{x} \qquad \text{ILR}(x)^{-1} = \phi(\boldsymbol{\Psi}\boldsymbol{x}) \tag{5}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d-1}$ is a basis where $\boldsymbol{\Psi}^T\boldsymbol{\Psi} = \boldsymbol{I}_{d-1}$ and $\boldsymbol{\Psi}\boldsymbol{\Psi}^T = \boldsymbol{I}_d - \frac{1}{d}\mathbf{1}_{d\times d}$. A naive implementation of the ILR transform can be memory and computationally intensive for large d. However, any orthonormal basis can be used to parameterize the ILR transform and some of these bases can be represented by binary trees (40; 41; 42). A binary tree can be used to represent $\boldsymbol{\Psi}$ with $O(d \log d)$ elements where the lth column vector of $\boldsymbol{\Psi}$ is given as follows

$$\boldsymbol{\Psi}_{\cdot l} = (\underbrace{0, \ldots 0}_{\boldsymbol{k}}, \underbrace{a, \ldots a}_{\boldsymbol{r}}, \underbrace{b, \ldots, b}_{\boldsymbol{s}}, \underbrace{0, \ldots, 0}_{\boldsymbol{t}}) \tag{6}$$

$$a = \frac{\sqrt{|\boldsymbol{s}|}}{\sqrt{|\boldsymbol{r}|(|\boldsymbol{r}| + |\boldsymbol{s}|)}} \qquad b = \frac{-\sqrt{|\boldsymbol{r}|}}{\sqrt{|\boldsymbol{s}|(|\boldsymbol{r}| + |\boldsymbol{s}|)}} \tag{7}$$

where l indexes an internal node in the tree with left children $\boldsymbol{r}$, right children $\boldsymbol{s}$, nodes to the left $\boldsymbol{k}$ and nodes to the right $\boldsymbol{t}$ (43) (Figure 4). If this binary tree is balanced, this will bring down the memory requirements of $\boldsymbol{\Psi}$ from $O(d^2)$ to $O(d \log d)$ and can reduce the matrix vector multplication runtime from $O(d^2)$ to $O(d \log d)$ (See Appendix B.1). In practice, this can speed up the matrix-vector multiplication operations by an order of magnitude for datasets with more than ten thousand dimensions.

4.3 MULTINOMIAL VARIATIONAL AUTOENCODER ARCHITECTURE

Our full Multinomial VAE model is given as follows

$$p(\boldsymbol{x}|\boldsymbol{\eta}) = \text{Mult}(\phi(\boldsymbol{\Psi}\boldsymbol{\eta})) \tag{8}$$

$$p(\boldsymbol{\eta}|\boldsymbol{z}, \boldsymbol{\theta_{dec}}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2 I_d) \tag{9}$$

$$q(\boldsymbol{z}|\boldsymbol{\theta_{enc}}, \boldsymbol{x}) = \mathcal{N}(F_L\big(\boldsymbol{\Psi}^T(\widetilde{\log}(\boldsymbol{x}) - \boldsymbol{\mu})\big), \boldsymbol{D}), \tag{10}$$

where $\boldsymbol{\theta_{dec}} = \{\boldsymbol{W}, \sigma^2\}$ denotes the decoder parameters and $\boldsymbol{\theta_{enc}} = \{F_L, \boldsymbol{D}\}$ denotes the encoder parameters and $\boldsymbol{\mu} \in \mathbb{R}^{d-1}$ is a bias parameter. Here, $q(\boldsymbol{z}|\boldsymbol{\theta_{enc}}, \boldsymbol{x})$ denotes the variational posterior distribution of $\boldsymbol{z}$ given by the encoder represented as an L layer dense neural network with appropriate activations. This encoder is directly used to evaluate $p(\boldsymbol{\eta}|\boldsymbol{z}, \boldsymbol{\theta_{dec}})$. Furthermore, flat priors are assumed for all variables except $\boldsymbol{z}$.

It is important to note potentially challenging modeling issues when designing the encoder. The ILR transform is not directly applicable to count data, since $\log(0)$ is undefined. A common approach to this problem is to introduce pseudocount beforehand, namely

$$\widetilde{\log}(\boldsymbol{x}) = \log(\boldsymbol{x} + 1)$$

The choice of pseudocount is arbitrary and can introduce biases. To circumvent this issue, we introduce a deep encoder neural network highlighted in Equation 10; we expect that the universal approximation theorem would apply here (44; 45), and the accuracy of estimating the latent representation $\boldsymbol{z}$ will improve with more complex neural networks. This is supported in our simulations benchmarks; more complex encoder architectures can better remove biases induced from the pseudocounts.

4.4 ALTERNATING MINIBATCH OPTIMIZATION PROCEDURE

Given that our objective here is to obtain the MAP estimate of the VAE model parameters, the VAE parameters $\boldsymbol{\theta} = \{F_L, \boldsymbol{W}, \boldsymbol{D}, \sigma^2\}$ can be obtained by estimating the global maximum of the posterior distribution. In the original CU sampler implementation, the parameter $(\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta_n})$ is optimized across the entire dataset and then $(\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta_n})$ is fixed in order to estimate the remaining parameters. For large studies, this can be memory demanding, since $(\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta_n}) \in \mathbb{R}^{n \times d-1}$ alone can scale to millions of parameters.

To scale this estimation procedure to large high-dimensional datasets we have devised a mini-batched alternating minimization procedure as follows

Algorithm 1 VAE Alternating Maximization Optimization

repeat
  for $\boldsymbol{X}^{(i)} \in \boldsymbol{X}$ **do**
    $\hat{\boldsymbol{H}} \leftarrow \arg\max_{\boldsymbol{\eta}} \left[\log p(\boldsymbol{X}^{(i)}|\boldsymbol{H})\right] \in \mathbb{R}^{b \times d-1}$
    $\boldsymbol{\theta} \leftarrow \arg\max_{\boldsymbol{\theta}} \left[\log q(\hat{\boldsymbol{H}}|\boldsymbol{\theta}, \boldsymbol{X}^{(i)}) + \log p(\boldsymbol{Z})\right]$
  end for

until convergence

For a mini-batch $\boldsymbol{X}^{(i)} = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(b)})$ of size b and the corresponding latent variables $\boldsymbol{Z}^{(i)} = (\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(b)})$ and $\boldsymbol{H}^{(i)} = (\boldsymbol{\eta}^{(1)}, \ldots, \boldsymbol{\eta}^{(b)})$, the quantities $\log p(\boldsymbol{X}^{(i)}|\boldsymbol{H}^{(i)})$ and $\log p(\boldsymbol{Z^{(i)}})$ are given by Equation 1. The variational posterior distribution of $\boldsymbol{\eta}$ given by $q(\boldsymbol{H}^{(i)}|\boldsymbol{\theta}, \boldsymbol{X}^{(i)}) = \prod_{j=1}^{b} q(\boldsymbol{\eta}^{(j)}|\boldsymbol{\theta}, \boldsymbol{x}^{(j)})$ can be obtained by marginalizing out $\boldsymbol{z}$ in Equations 8 and 9 as follows

$$q(\boldsymbol{\eta}^{(j)}|\boldsymbol{\theta}, \boldsymbol{x}^{(j)}) = \mathcal{N}\big(\boldsymbol{W}F_L\big(\boldsymbol{\Psi}^T(\widetilde{\log}(\boldsymbol{x}^{(j)}) - \boldsymbol{\mu})\big) + \boldsymbol{\mu},\ \boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}_d\big), \tag{11}$$

whose covariance matrix $\boldsymbol{W}\boldsymbol{D}\boldsymbol{W}^T + \sigma^2\boldsymbol{I} \in \mathbb{R}^{d-1 \times d-1}$ can be efficiently inverted using the Woodbury identity (46; 47). The prior $\log p(\boldsymbol{Z^{(i)}})$ is evaluated given, $\log p(\boldsymbol{Z^{(i)}}) = \mathcal{N}(\boldsymbol{Z^{(i)}}|0, \boldsymbol{I}_k)$ where

$\boldsymbol{Z}^{(i)} = F_L\big(\boldsymbol{\Psi}^T(\widetilde{\log}(\boldsymbol{X}^{(i)}) - \boldsymbol{\mu})\big)$. The optimal $\hat{\boldsymbol{H}}^{(i)}$ that maximizes $\log p(\boldsymbol{X}^{(i)}|\boldsymbol{H}^{(i)})$ is obtained through gradient descent optimization. Only once $\hat{\boldsymbol{H}}^{(i)}$ is obtained, the remaining VAE parameters $\boldsymbol{\theta}$ can be estimated via gradient descent optimization. Similar to the EM algorithm proposed in (48), this procedure alternates between estimating $\boldsymbol{H}^{(i)}$ and $\boldsymbol{\theta}$ and repeats this procedure until convergence. For a single layer linear encoder, this optimization procedure will eventually reach the global maxima with respect to $\boldsymbol{H}^{(i)}$ and the multivariate normal mean and covariance in $q(\boldsymbol{H}^{(i)}|\boldsymbol{\theta}, \boldsymbol{X}^{(i)})$, since the multinomial and the normal distributions are both log concave functions. As a result, $\boldsymbol{W}$ and $\boldsymbol{D}$ can be estimated up to rotation and scale (Appendix B.4). This is difficult to claim for complex encoder architectures due to non-linear activations; simulations can be constructed to evaluate how well the decoder weights can be recovered as discussed in Section 5.1. Since $\hat{\boldsymbol{H}}^{(i)}$ needs to be accurately estimated before optimizing $F_L, \boldsymbol{W}, \boldsymbol{\sigma}, \boldsymbol{D}$, multiple gradient descent steps are required for a given batch.

In practice, obtaining the optimal $\hat{\boldsymbol{H}}^{(i)}$ for a given minibatch i may require hundreds of gradient descent steps from a randomly initialization. Since $\boldsymbol{H}^{(i)}$ is used to approximate the Multinomial parameters describing the observed counts $\boldsymbol{x}$, we can initialize $\boldsymbol{H}^{(i)}$ with $\hat{\boldsymbol{H}}_0^{(i)} = \boldsymbol{\Psi}^T\widetilde{\log}(\boldsymbol{X}^{(i)})$, where $\hat{\boldsymbol{H}}_0^{(i)}$ is the initialization of $\boldsymbol{H}^{(i)}$ prior to applying gradient descent updates. In practice, this can significantly reduce the necessary number of gradient descent updates per minibatch.

5 Results

5.1 Multinomial PPCA and VAEs agree in Simulation Benchmarks

To determine if the proposed multinomial variational autoencoder can recover principal components, we extended the benchmarks proposed in (14). Here, we benchmarked our proposed analytical VAE to the stochastic VAEs (5) with a multinomial likelihood. The multi-layer encoders were parameterized with interwoven softplus activations with final linear layer yielding the latent representation $\boldsymbol{z}$. We evaluated the performance of these methods with respect to the choice of encoder depth as well as the choice of log-ratio transform, including the ILR transform, ALR transform and the standard Log transform that is most commonly used. To evaluate these methods, we simulated two microbiome datasets with 200 species and 1000 species to test how well these methods can scale with respect to input dimension and sparsity. We use axis-alignment and subspace distance proposed by (14) in addition to Procrustes analysis (49) to measure the difference between the ground truth eigenvectors and the VAE decoder weights $\boldsymbol{W}$. Since PPCA and VAEs will only be able to correctly identify $\boldsymbol{W}$ up to scale and rotation (see Appendix B.4), we also added Pearson correlation to measure the agreement between thethe estimated row Euclidean distances of $\boldsymbol{W}$ and the simulated ground-truth row Euclidean distances as shown in Figure 5.

It is important to note that due to the high-level of sparsity, much of the information of the underlying latent parameters is lost. As a result, we do not expect the principal components to be perfectly recovered. This is not an issue for dense dataset as shown in Figure 5, the decoder weights $\boldsymbol{W}$ have over 0.9 correlation with the ground truth decoder weights with both the stochastic and analytical VAEs that utilize the ILR transform.

From the results shown in Figure 1a-f and Figure 5a,c, it is clear that the ILR transform yields superior accuracy compared to both the ALR transform and the Log transform in both the stochastic VAEs as well as the analytical VAEs across all metrics. This observation corroborates previous findings when performing compositional PCA (39; 20). Furthermore, from the benchmarks shown in Figure 1g-i and and Figure 5b, it appears that increasing the complexity of encoder with the ILR transform can improve estimation of the principal components, which is pronounced in the axis-alignment metrics; the stochastic VAE with 5 layers had the best performance in the 200 dimension dataset, whereas the analytical VAE with 3 layers had the best performance in the 1000 dimensional dataset. As shown in Figure 1b,e both the stochastic VAE and the analytical VAE augmented with the ILR transform converge to the same subspace distance and similar procrustes error, suggesting that they learn similar row correlations and have the same magnitude of residual error after correcting for rotation and scaling.

There are some discrepancies between our proposed analytical VAE and the stochastic VAE. Our proposed analytical optimization using the ILR transform reaches convergence in fewer epochs than

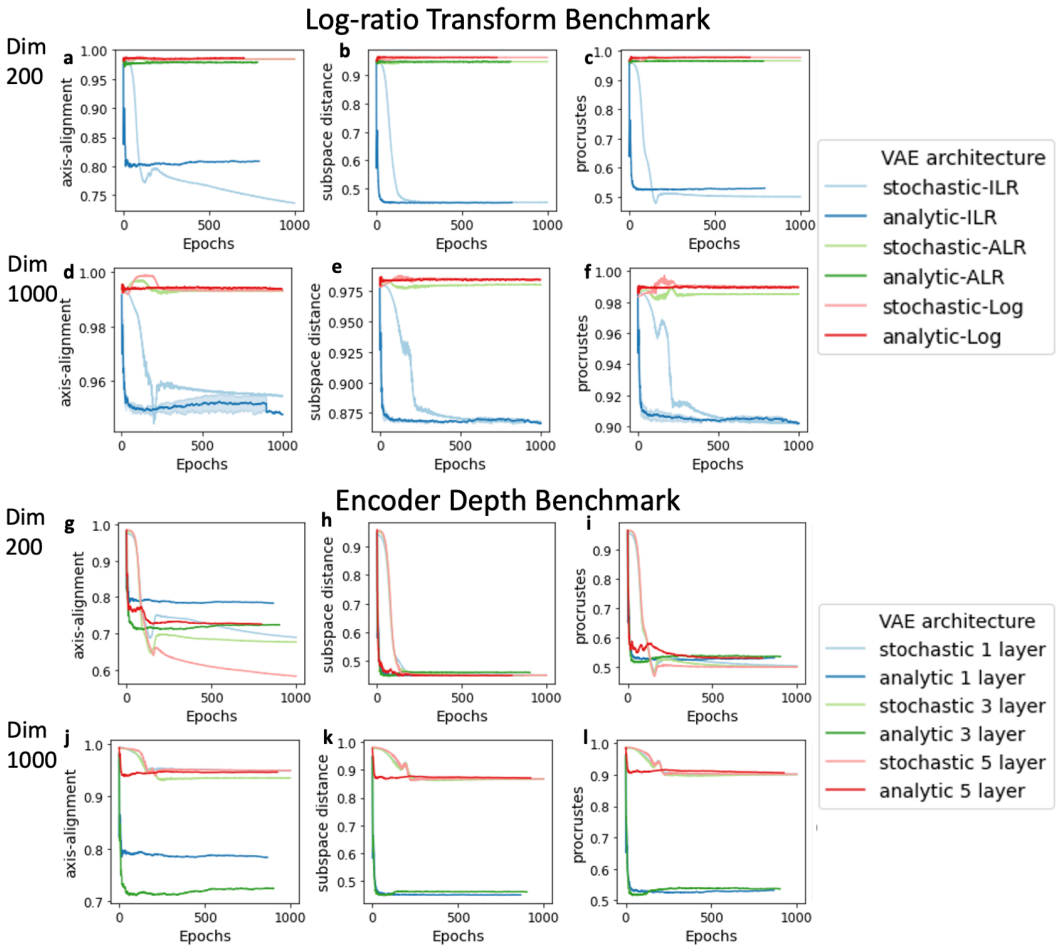

Figure 1: Comparison of different log-ratio transforms (a-f) and encoder depths (g-l). Two datasets with 200 and 1000 input dimensions with the same sequencing depth are simulated for benchmarks (a-c, g-i) and (d-f, j-l) respectively. Log-ratio transform benchmarks investigate different log-ratio transforms using a 5 layer encoder. The Encoder Depth benchmark investigates the impact of the number of encoder layer combined with the ILR transform. The alignment metric measures the angular differences between $\boldsymbol{W}$ and the ground truth. The subspace distance is a measure of agreement between the ground truth correlations and the correlations estimated from $\boldsymbol{W}$. Procrustes measures the residual error after obtaining the best rotation and scaling factors to match the decoder weights to the ground truth. See Appendix D.3 for details.

the stochastic VAEs proposed by Kingma (5) across all metrics in the sparse simulation benchmark. However, the analytical VAEs appears to have an advantage that is particularly pronounced in the larger dataset, whereas the stochastic VAEs appear to have a stronger performance on smaller, dense datasets. It is difficult to explain the reason for performance gap between the our proposed analytical VAE and the stochastic VAE due the analytical intractability of the Multinomial VAE ELBO. The challenge of accurately estimating an optimal $\boldsymbol{H}$ for each minibatch, could be one limiting factor behind the performance of our proposed analytical VAE.

5.2 Pretraining Multinomial VAEs on a very large compendium of mouse microbiome observations

To showcase this on real microbiome datasets, 11k mouse fecal pellet samples and 3.8k blank control samples with a total of 8.8k microbial taxa across 45 studies were retrieved from qiita (50) using redbiom (51). Since technical variation can vary across experimental conditions and induce study specific "batch effects", we investigate the merits of estimating and removing study specific batch

effects using the blank controls. In total, we trained 4 models to evaluate the differences between the stochastic VAE and the analytical VAE, in addition to the differences due to batch correction (Appendix C). We trained these models for 10k epochs and the models with the smallest validation error are reported here. The details behind the full training procedure are given in Appendix D.2. From visualizing learned decoder weights shown in Figure 2, it is clear that both the stochastic

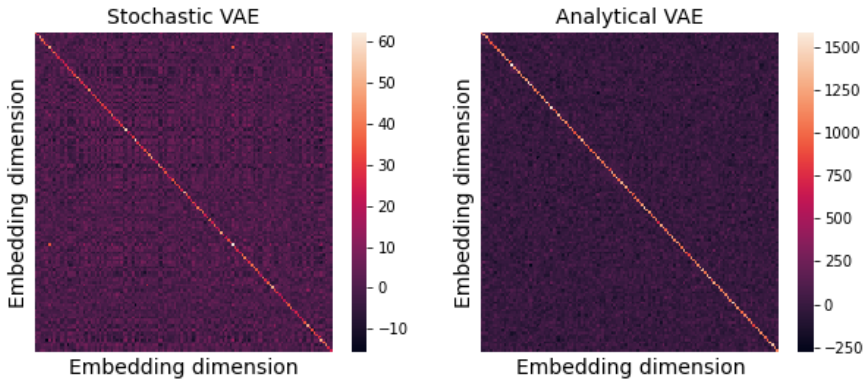

Figure 2: Comparison of the decoder inner product $\boldsymbol{W}^T\boldsymbol{W}$ between the pretrained stochastic VAE and the pretrained analytical VAE. The eigenvalues are proportional to the diagonal of $\boldsymbol{W}^T\boldsymbol{W}$.

VAE and the analytical VAE are able to learn orthogonal embeddings, since $\boldsymbol{W}^T\boldsymbol{W}$ approximates a diagonal matrix, as suggested by (13; 4; 52). Unlike Linear VAEs (13) the connection between the eigenvalues and the decoder weights is not as clear due to the asymmetry between the encoder and the decoder; scale identifiability of the decoder weights complicates the process of estimating the eigenvalues. This is apparent in Figure 2; heatmaps of the $\boldsymbol{W}$ inner products are on vastly different scales. Some strategies on how to enforce scale identifiability and recover the eigenvalues are discussed in the Appendix B.4.

One of the advantages of employing a pretraining strategy with VAEs is that the pretrained models can enable one-shot learning (53). By training on a large unlabeled dataset, a lower dimensional representation across the observed data can be constructed, which requires fewer labeled samples for training downstream classifiers. We will showcase this property in two classification benchmarks below.

5.3 Classification and correlation benchmarks

To construct benchmarks on real biological datasets we analysed 16S sequencing data obtained from mouse fecal pellets in a study conducted by Tripathi et al investigating hypoxia (54) and a study conducted by Shalapour et al investigating cancer effects on mice (55).

Due to the connections between Multinomial VAEs and compositional PCA (17; 19), we expect that, if our VAE is performing well, we will see that the row Euclidean distances of $\boldsymbol{\Psi}\boldsymbol{W} \in \mathbb{R}^{d\times k}$ correspond to Lovell's un-normalized proportionality metric (56; 20) given by

$$\left\| (\boldsymbol{\Psi}\boldsymbol{W})_i - (\boldsymbol{\Psi}\boldsymbol{W})_j \right\|_2 \propto \text{Var}\left(\log \frac{x_i}{x_j} \right), \tag{12}$$

where $\boldsymbol{x}$ refers to the observed microbe proportions and i and j refer to two different microbial taxa. Only a relative relationship between the proportionality metric and the VAE decoder row distances can be stated due to scale identifiability issues inherent in PPCA. Potential strategies to correct for this scale identifiability are further discussed in the Appendix B.4. Proportionality has been shown to be a more reliable metric for compositional data compared to standard correlation metrics such as Pearson and Spearman (57; 58; 59; 60) supporting the notion that our proportional comparison of our VAE decoder to Lovell's un-normalized proportionality metric will provide a sound perspective and a framework for interpreting the weights of the VAE.

To determine how well the estimated VAE embeddings agree with the proportionality metric on microbiome data, the pairwise decoder distances are compared to the pairwise proportionality metrics of the top 300 most abundant microbes (Figure 7 and Table 1). To test how the VAE representations can be utilized for classification, K-nearest neighbors (KNN) classification on the learned VAE encodings was performed on the two mouse pellet datasets, classifying mice that were induced with hypoxia (54) and mice based on their phenotype (55). These classification models are compared against the baseline model where KNN is applied to the raw counts.

Dataset	Method	Batch-corrected	F1 score	AUC	Lovell
Shalapour et al	analytical-VAE	False	0.700 ± 0.021	$\mathbf{0.910} \pm 0.013$	$\mathbf{0.53}$
		True	0.702 ± 0.025	0.861 ± 0.015	0.49
	stochastic-VAE	False	$\mathbf{0.777} \pm 0.024$	0.908 ± 0.014	0.44
		True	0.135 ± 0.025	0.555 ± 0.011	0.29
	raw	NA	0.433 ± 0.039	0.686 ± 0.018	NA
Tripathi et al	analytical-VAE	False	0.867 ± 0.005	0.873 ± 0.004	0.60
		True	0.859 ± 0.004	0.865 ± 0.004	$\mathbf{0.61}$
	stochastic-VAE	False	$\mathbf{0.884} \pm 0.004$	$\mathbf{0.886} \pm 0.003$	0.55
		True	0.872 ± 0.004	0.876 ± 0.004	0.23
	raw	NA	0.700 ± 0.007	0.729 ± 0.006	NA

Table 1: Classification and Correlation benchmarks. KNN classifications with k=5 was used for all representations. 100 rounds of 10 fold cross validation was applied to estimate average F1 score, AUC and standard errors. Raw indicates that no transformation was applied to the raw counts before performing KNN classification. The agreement between the VAE decoder weights and Lovell's proportionality metric are measured according to Pearson's r.

From Table 1, it doesn't appear that the batch effects improves the performance of these methods. One possibility is that the blank controls are not optimal technical replicates, which could explain the drop in performance of the stochastic VAEs. Another possibility as hinted (7), is that our models have enough latent dimensions can implicitly learn the batch effects, which could explain why there isn't a major difference between the analytical VAE and batch-corrected analytical VAE.

In the classification benchmarks presented in Table 1, the accuracy of our analytical VAE is competitive with the stochastic VAE in both datasets. Across both datasets, both VAE models substantially outperform the baseline. This performance improvement is particularly pronouced in the HCC dataset (55) where there are 52 possible phenotypes.

The difference between the analytical VAE and the stochastic VAE is apparent when measuring the agreement to Lovell's proportional metric. We don't expect the VAEs to exactly match Lovell's proportionality since the classification benchmarks datasets are a subset of the full pretraining dataset. However, our proposed analytical VAE embeddings are in better agreement compared to the stochastic VAE, suggesting our analytical VAE better agrees with compositional PCA and can learn more biologically relevant embeddings. This is also in agreement with the simulation findings, showing that our proposed analytical VAEs better agree with Multinomial PPCA in sparse data.

Conclusion

Prior to this work, there was not a clear consensus on how to define Multinomial PPCA. Here, we have provided another definition that is consistent with standard PCA using the ILR transform. Furthermore, we have provided a proof-of-concept that Multinomial VAEs that can learn the principal components obtained from Multinomial PPCA. These methods can be used in one shot learning and transfer learning settings, requiring fewer labeled samples for training and/or allowing for the use of pre-trained models to extract features from smaller, noisier or batch-convolved data-sets. Due to the connection to compositional PCA, the learned VAE decoder embeddings are interpretable and can be directly linked to Lovell's proportionality metric. All in all, Multinomial VAEs are an attractive, scalable solution for estimating large covariance matrices from count data that is prevalent across many scientific disciplines.

CODE AVAILABILITY

All of our software and analyses can be found at https://zenodo.org/record/4062719.X3apeZpME5k

ACKNOWLEDGEMENTS

We want to acknowledge Pytorch (61), Pytorch-Lightning (62), the Biom format (63), Matplotlib (64), Scipy (65) and Numpy (66) for providing the software foundation that this work was built upon.

REFERENCES

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[2] Steven C Althoen and Renate Mclaughlin. Gauss-jordan reduction: A brief history. *The American mathematical monthly*, 94(2):130–142, 1987.

[3] Aravindh Krishnamoorthy and Deepak Menon. Matrix inversion using cholesky decomposition. In *2013 signal processing: Algorithms, architectures, arrangements, and applications (SPA)*, pages 70–72. IEEE, 2013.

[4] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse. (NeurIPS), 2019.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] 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, 1999.

[7] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

[8] Min Oh and Liqing Zhang. Deepmicro: deep representation learning for disease prediction based on microbiome data. *Scientific reports*, 10(1):1–9, 2020.

[9] Sam Sinai, Eric Kelsic, George M Church, and Martin A Nowak. Variational auto-encoding of protein sequences. *arXiv preprint arXiv:1712.03346*, 2017.

[10] Xiaojie Guo, Sivani Tadepalli, Liang Zhao, and Amarda Shehu. Generating tertiary protein structures via an interpretative variational autoencoder. *arXiv preprint arXiv:2004.07119*, 2020.

[11] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *BioRxiv*, 2020.

[12] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. *arXiv preprint arXiv:1702.08139*, 2017.

[13] Daniel Kunin, Jonathan M Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. *arXiv preprint arXiv:1901.08168*, 2019.

[14] Xuchan Bao, James Lucas, Sushant Sachdeva, and Roger Grosse. Regularized linear autoencoders recover the principal components, eventually. *arXiv preprint arXiv:2007.06731*, 2020.

[15] John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

[16] Peter Filzmoser and Karel Hron. Correlation analysis for compositional data. *Mathematical Geosciences*, 41(8):905, 2009.

[17] John Aitchison. Principal component analysis of compositional data. *Biometrika*, 70(1):57–65, 1983.

[18] Juan José Egozcue, Vera Pawlowsky-Glahn, Glòria Mateu-Figueras, and Carles Barcelo-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

[19] Peter Filzmoser, Karel Hron, and Clemens Reimann. Principal component analysis for compositional data with outliers. *Environmetrics: The Official Journal of the International Environmetrics Society*, 20(6):621–632, 2009.

[20] Juan José Egozcue, Vera Pawlowsky-Glahn, and Gregory B Gloor. Linear association in compositional data analysis. *Austrian Journal of Statistics*, 47(1):3–31, 2018.

[21] Zachary D Kurtz, Christian L Müller, Emily R Miraldi, Dan R Littman, Martin J Blaser, and Richard A Bonneau. Sparse and compositionally robust inference of microbial ecological networks. *PLoS Comput Biol*, 11(5):e1004226, 2015.

[22] Jonathan Friedman and Eric J Alm. Inferring correlation networks from genomic survey data. *PLoS Comput Biol*, 8(9):e1002687, 2012.

[23] Julien Chiquet, Mahendra Mariadassou, and Stéphane Robin. Variational inference for sparse network reconstruction from count data. pages 1–30, 2018.

[24] Huaying Fang, Chengcheng Huang, Hongyu Zhao, and Minghua Deng. gcoda: conditional dependence network inference for compositional data. *Journal of Computational Biology*, 24(7):699–708, 2017.

[25] Janko Tackmann, João Frederico Matias Rodrigues, and Christian von Mering. Rapid inference of direct interactions in large-scale ecological networks from heterogeneous microbial sequencing data. *Cell systems*, 9(3):286–296, 2019.

[26] Wray L Buntine and Aleks Jakulin. Applying discrete pca in data analysis. *arXiv preprint arXiv:1207.4125*, 2012.

[27] Wray Buntine. Variational extensions to em and multinomial pca. In *European Conference on Machine Learning*, pages 23–34. Springer, 2002.

[28] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[29] Scott W. Linderman, Matthew J. Johnson, and Ryan P. Adams. Dependent multinomial models made easy: Stick breaking with the Pólya-gamma augmentation. *Advances in Neural Information Processing Systems*, 2015-Janua:3456–3464, 2015.

[30] Justin D. Silverman, Kimberly Roche, Zachary C. Holmes, Lawrence A. David, and Sayan Mukherjee. Bayesian Multinomial Logistic Normal Models through Marginally Latent Matrix-T Processes. pages 1–39, 2019.

[31] Travis E Gibson and Georg K Gerber. Robust and scalable models of microbiome dynamics. *arXiv preprint arXiv:1805.04591*, 2018.

[32] Tarmo Äijö, Christian L Müller, and Richard Bonneau. Temporal probabilistic modeling of bacterial compositions derived from 16s rrna sequencing. *Bioinformatics*, 34(3):372–380, 2018.

[33] Tyler A Joseph, Amey P Pasarkar, and Itsik Pe'er. Efficient and accurate inference of mixed microbial population trajectories from longitudinal count data. *Cell Systems*, 10(6):463–469, 2020.

[34] J Atchison and Sheng M Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

[35] Justin D Silverman, Heather K Durand, Rachael J Bloom, Sayan Mukherjee, and Lawrence A David. Dynamic linear models guide design and analysis of microbiota studies within artificial human guts. *Microbiome*, 6(1):1–20, 2018.

[36] Neal S Grantham, Yawen Guan, Brian J Reich, Elizabeth T Borer, and Kevin Gross. Mimix: A bayesian mixed-effects model for microbiome data from designed experiments. *Journal of the American Statistical Association*, 115(530):599–609, 2020.

[37] Dean Billheimer, Peter Guttorp, and William F Fagan. Statistical interpretation of species composition. *Journal of the American statistical Association*, 96(456):1205–1214, 2001.

[38] David M. Blei and John D. Lafferty. Correlated topic models. *Advances in Neural Information Processing Systems*, pages 147–154, 2005.

[39] Vera Pawlowsky-Glahn, Juan José Egozcue, and Raimon Tolosana-Delgado. *Modeling and analysis of compositional data*. John Wiley & Sons, 2015.

[40] James T Morton, Jon Sanders, Robert A Quinn, Daniel McDonald, Antonio Gonzalez, Yoshiki Vázquez-Baeza, Jose A Navas-Molina, Se Jin Song, Jessica L Metcalf, Embriette R Hyde, et al. Balance trees reveal microbial niche differentiation. *MSystems*, 2(1), 2017.

[41] Justin D Silverman, Alex D Washburne, Sayan Mukherjee, and Lawrence A David. A phylogenetic transform enhances analysis of compositional microbiota data. *Elife*, 6:e21887, 2017.

[42] Alex D Washburne, Justin D Silverman, Jonathan W Leff, Dominic J Bennett, John L Darcy, Sayan Mukherjee, Noah Fierer, and Lawrence A David. Phylogenetic factorization of compositional data yields lineage-level associations in microbiome datasets. *PeerJ*, 5:e2969, 2017.

[43] Juan José Egozcue and Vera Pawlowsky-Glahn. Groups of parts and their balances in compositional data analysis. *Mathematical Geology*, 37(7):795–828, 2005.

[44] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[45] Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on Learning Theory*, pages 2306–2327, 2020.

[46] Max A Woodbury. Inverting modified matrices. *Memorandum report*, 42(106):336, 1950.

[47] William H Press, Saul A Teukolsky, Brian P Flannery, and William T Vetterling. *Numerical recipes in Fortran 77: volume 1, volume 1 of Fortran numerical recipes: the art of scientific computing*. Cambridge university press, 1992.

[48] Jonathan Chung-Kuan Huang and Tomasz Malisiewicz. Fitting a hierarchical logistic normal distribution. *Unpublished Manuscript*, pages 2–4.

[49] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[50] Antonio Gonzalez, Jose A Navas-Molina, Tomasz Kosciolek, Daniel McDonald, Yoshiki Vázquez-Baeza, Gail Ackermann, Jeff DeReus, Stefan Janssen, Austin D Swafford, Stephanie B Orchanian, et al. Qiita: rapid, web-enabled microbiome meta-analysis. *Nature methods*, 15(10):796–798, 2018.

[51] Daniel McDonald, Benjamin Kaehler, Antonio Gonzalez, Jeff DeReus, Gail Ackermann, Clarisse Marotz, Gavin Huttley, and Rob Knight. redbiom: a rapid sample discovery and feature characterization system. *mSystems*, 4(4), 2019.

[52] Xuchan Bao, James Lucas, and Roger Grosse. Regularized linear autoencoders recover the principal components , eventually. 2018.

[53] Danilo Jimenez Rezende, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-shot generalization in deep generative models. *arXiv preprint arXiv:1603.05106*, 2016.

[54] Anupriya Tripathi, Alexey V Melnik, Jin Xue, Orit Poulsen, Michael J Meehan, Gregory Humphrey, Lingjing Jiang, Gail Ackermann, Daniel McDonald, Dan Zhou, et al. Intermittent hypoxia and hypercapnia, a hallmark of obstructive sleep apnea, alters the gut microbiome and metabolome. *MSystems*, 3(3), 2018.

[55] Shabnam Shalapour, Xue-Jia Lin, Ingmar N Bastian, John Brain, Alastair D Burt, Alexander A Aksenov, Alison F Vrbanac, Weihua Li, Andres Perkins, Takaji Matsutani, et al. Inflammation-induced iga+ cells dismantle anti-liver cancer immunity. *Nature*, 551(7680):340–345, 2017.

[56] David Lovell, Vera Pawlowsky-Glahn, Juan José Egozcue, Samuel Marguerat, and Jürg Bähler. Proportionality: a valid alternative to correlation for relative data. *PLoS Comput Biol*, 11(3):e1004075, 2015.

[57] Michael A Skinnider, Jordan W Squair, and Leonard J Foster. Evaluating measures of association for single-cell transcriptomics. *Nature methods*, 16(5):381–386, 2019.

[58] Ionas Erb and Cedric Notredame. How should we measure proportionality on relative gene expression data? *Theory in Biosciences*, 135(1-2):21–36, 2016.

[59] Thomas P Quinn, Mark F Richardson, David Lovell, and Tamsyn M Crowley. propr: an r-package for identifying proportionally abundant features using compositional data analysis. *Scientific reports*, 7(1):1–9, 2017.

[60] Gregory B Gloor, Jean M Macklaim, Vera Pawlowsky-Glahn, and Juan J Egozcue. Microbiome datasets are compositional: and this is not optional. *Frontiers in microbiology*, 8:2224, 2017.

[61] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[62] WA Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

[63] Daniel McDonald, Jose C Clemente, Justin Kuczynski, Jai Ram Rideout, Jesse Stombaugh, Doug Wendel, Andreas Wilke, Susan Huse, John Hufnagle, Folker Meyer, et al. The biological observation matrix (biom) format or: how i learned to stop worrying and love the ome-ome. *Gigascience*, 1(1):2047–217X, 2012.

[64] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[65] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[66] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *arXiv preprint arXiv:2006.10256*, 2020.

[67] John Aitchison, Carles Barceló-Vidal, José Antonio Martín-Fernández, and Vera Pawlowsky-Glahn. Logratio analysis and compositional distance. *Mathematical Geology*, 32(3):271–275, 2000.

[68] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

[69] Nicole M Davis, Diana M Proctor, Susan P Holmes, David A Relman, and Benjamin J Callahan. Simple statistical identification and removal of contaminant sequences in marker-gene and metagenomics data. *Microbiome*, 6(1):226, 2018.

[70] James T Morton, Clarisse Marotz, Alex Washburne, Justin Silverman, Livia S Zaramela, Anna Edlund, Karsten Zengler, and Rob Knight. Establishing microbial composition measurement standards with reference frames. *Nature communications*, 10(1):1–11, 2019.

[71] Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. Vsearch: a versatile open source tool for metagenomics. *PeerJ*, 4:e2584, 2016.