PolyVI: Deep Generative Models for Gene Expression, Chromatin Accessibility, and Surface Protein Expression Data

Rohan Koodli

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-113
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-113.html

May 13, 2022

Acknowledgement

I thank my mentors Tal Ashuach, Alyssa Morrow, Matt Jones, Adam Gayoso, Nir Yosef, and the entire Yosef Lab for their mentorship and guidance over the past two years.

I thank my friends at Berkeley and beyond for all the laughs and memories over the past 4 years.

Finally, I thank my family for their unwavering support throughout my academic, personal, and research journeys.

PolyVI: Deep Generative Models for Gene Expression, Chromatin Accessibility, and Surface Protein Expression Data

by Rohan V. Koodli

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Nir Yosef
Research Advisor

5/12/22

(Date)

* * * * * * *

Professor Nilah M. Ioannidis
Second Reader

5/12/22

(Date)

PolyVI: Deep Generative Models for Gene Expression, Chromatin Accessibility, and Surface Protein Expression Data

by

Rohan V. Koodli

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Abstract

PolyVI: Deep Generative Models for Gene Expression, Chromatin Accessibility, and Surface Protein Expression Data

by

Rohan V. Koodli

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Single-cell multimodal sequencing methods, which measure multiple different modalities simultaneously (such as gene expression, chromatin accessibility, and surface protein data) are an exciting new space in the field of genomics as they provide a more comprehensive picture of cellular state than technologies that assay a single modality. Here I present PolyVI, a suite of three deep generative models to analyze DOGMA-seq (gene expression, protein, chromatin), ASAP-seq (chromatin, protein), and SNARE-seq (gene expression, chromatin) datasets. PolyVI is able to map the data to a low dimensional latent space, batch correct, and de-noise the data.

Contents

Chapter 1

Introduction

Single-cell genomic sequencing has transformed molecular biology. Sequencing technologies such as single-cell RNA-sequencing [19] and single-cell ATAC-sequencing (Assay for Transposase-Accessible Chromatin sequencing) [4] are pivotal to help understand the gene expression and accessibility of genomic regions within cells. Up until recently, a majority of the methods being used to profile cellular data has been limited to a single modality [12, 3]; for example, single-cell ATAC-seq only profiles chromatin accessibility in a cell, and single-cell RNA-seq only profiles gene expression levels.

Multimodal single-cell data is particularly of interest, since providing more than one modality can provide a much clearer picture of cell state. Specifically, multiomic sequencing methods such as CITE-seq [17] profile cell surface protein and transcriptome (gene expression) data simultaneously. Recently, additional multimodal technologies have been published, such as DOGMA-seq (gene expression, chromatin accessibility, surface protein expression), ASAP-seq (chromatin accessibility, surface protein expression) [14], and SNARE-seq (gene expression, chromatin accessibility) [5]. Leveraging all the modalities in such multimodal assays is difficult, as such a method should be able to collate all the modalities to learn a joint representation for each cell, thus providing a summary of its molecular state.

We present PolyVI, a suite of three deep generative models for probabilistic modelling of ASAP-seq (AsapVI), DOGMA-seq (DogmaVI), and SNARE-seq (SnareVI). PolyVI is implemented as part of the scvi-tools variational inference methods [12] and learns joint representations of DOGMA-seq, ASAP-seq, and SNARE-seq data. PolyVI is based off existing scvi-tools methods ScVI [12] (which models gene expression data), PeakVI [3] (chromatin accessibility), and TotalVI [6] (gene expression and surface protein data). PolyVI maps paired measurements to low-dimensional latent spaces, which can then be used to stratify cells into types, subtypes, quantitative gradients (such as developmental stage), and can be visualized using a dimensionality reduction technique. In addition, PolyVI predicts likelihoods of each of the modalities, quantifies uncertainties, and corrects for noise and batch effects in the data.

Chapter 2

Results

2.1 Architecture Overview

In order to provide the batch correction, latent cell type separation, and de-noising functionalities, PolyVI uses a variational autoencoder (VAE) architecture [10]. Specifically, the VAE architecture is comprised of two main parts: an encoder and a decoder neural network. A diagram of the PolyVI architecture can be seen in Figure 1. A formal description of the architecture is provided in Methods.

The encoder is comprised of a single feedforward neural network that learns a joint, batch-corrected, Gaussian latent distribution of all modalities provided, $q_\phi(z|x_R, x_A, x_P, s)$, where z is the latent variable learned, x is the original data, and s is the batch. The encoder is able to batch correct the data and produces a latent representation of the data, which can then be used with a dimensionality reduction technique such as UMAP [13].

The decoder is comprised of several feedforward neural networks that sample from z using distinct noise models for each modality to return observations that have been de-noised. Specifically, we use a Negative Binomial noise model for scRNA-seq data (gene expression data is count-based) [12], a Bernoulli noise model for scATAC-seq (chromatin accessibility data is mostly binary) [3], and a Negative Binomial Mixture noise model for surface protein expression data (protein data is also count-based but with non-zero background noise due to non-specific binding of antibodies) [6]. DogmaVI consists of all three of the aforementioned decoders, AsapVI models scATAC-seq and protein data, and SnareVI consists of scRNA-seq and scATAC-seq.

2.2 Inference Model Analysis

In single-cell genomic datasets, different cell culture environments, different synthesis techniques, etc. cause batch effects in datasets; batch correction measures how well inference methods can filter out the discrepancies across different batches. Latent cell type separation

measures how well inference models can separate different cell types. We use local inverse Simpson's index (LISI), as described in [11], to compute both these metrics.

The LISI scores we use measure how well batches are mixed (iLISI) and how separate different cell types are (cLISI). For iLISI, we want batches to be mized evenly; so in any given neighborhood of cells, an effective integration of cells should have n batches. Thus, the optimal batch correction iLISI score is n [11] (in our dataset, $n = 2$). Similarly, cLISI measures the separation of cell types, so in any given neighborhood of cells, we want only 1 cell type to be present [11]. Thus, the optimal cLISI is 1 and the worst possible cLISI would be m where m is the number of cell types (our dataset has $m = 8$). Detailed descriptions of the two metrics can be found in 5.3 and 5.4.

We compared PolyVI to three other multiomic methods: Schema [16], which uses metric learning to identify specific features per modality; MOFA+ [1], which uses statistical inference methods to construct low dimensional representations; and Seurat_v4 [8], which uses a weighted nearest neighbor approach to rank cells. As opposed to PolyVI, the three aforementioned algorithms do not have an analog to the generative model. Schema, MOFA+, and Seurat_v4 only contain the inference aspect of the analysis, mainly to infer a low dimensional embedding that summarizes all modalities and can be used to perform batch and visualize latent spaces. The LISI scores for PolyVI, Schema, MOFA+, and Seurat_v4 are listed in Tables 1 and 2 below.

We tested these models on the Mimitou et al dataset [14], which contains DOGMA-seq bone marrow PBMCs (peripheral blood mononuclear cell). This data contains two batches, cells treated with low-loss lysis (LLL) and cells treated with digitonin (DIG). In addition, this dataset contains monocytes (CD14, CD16 cells) and lymphocytes (CD4, CD8 T cells, other T cells, B cells, NK cells), which serve as our cell types to benchmark cell type separation.

Figure 2 shows UMAP visualizations of PolyVI latent spaces. Figure 2A shows Leiden algorithm [20] clusters based on gene expression, protein expression, and chromatin accessibility. Figure 2B shows the UMAP with annotated cell types (included in the original Mimitou et al study); we can see that the Leiden clusters are similar to the pre-determined cell types given in the dataset, implying that PolyVI is able to separate cell types well. In particular, clusters 4 and 6 in Figure 2A correspond well with B cells and NK cells in Figure 2B. In addition, the Leiden clustering distinguishes between the two largest subpopulations of cells, the CD4 and CD8 T cells, even though the clusters are subdivided (Figure 2A, see clusters 0, 1 and 2, 3). Figure 2C shows that the two batches, LLL and DIG, are mixed well throughout the latent space. This shows that PolyVI is able to batch correct well, further backed up by the high iLISI score shown in Table 1.

Algorithm	iLISI
PolyVI	1.89
Schema	1.88
MOFA+	1.88
Seurat_v4	1.74

Table 1: Batch correction scores

Algorithm	cLISI
PolyVI	2.09
Schema	1.50
MOFA+	2.03
Seurat_v4	2.06

Table 2: Latent separation scores

As seen above, PolyVI performs the best out of the four algorithms on iLISI, albeit by a slim margin. PolyVI, MOFA+, and Seurat_v4 exhibit similar performance on cLISI, while Schema performs the best. While this is a good result, the main benefit of using PolyVI over these methods is that PolyVI can de-noise and impute data in addition to its batch correction functionalities, which are discussed in section 2.3.

In addition, we decided to compare PolyVI to MultiVI [2], a similar scvi-tools model that supports scRNA-seq and scATAC-seq. MultiVI has the same functionality as PolyVI (SnareVI), but can also work with data that is not fully paired (i.e., either some gene expression or accessibility data is missing). MultiVI is able to accomplish this as it has 2 separate encoders, one for chromatin accessibility and one for gene expression, as opposed to PolyVI (which has 1 encoder for all modalities). We decided to compare the two models on the Mimitou data for the inference model (since the Mimitou data has batches), and on a fully paired 10x granulocyte dataset [15] for the generative model. We benchmark the two models on *only fully paired scRNA-seq and scATAC-seq*, as PolyVI only supports fully paired data, and MultiVI currently does not support protein data.

Batch corrected latent spaces for the two models can be seen in Figure 3; both models appear to have good mixing between the DIG and LLL batches of the Mimitou data. We used the SnareVI model within PolyVI to compare against MultiVI since MultiVI currently does not support protein data. Batch correction scores for SnareVI and MultiVI are shown below in Table 3.

Algorithm	iLISI
PolyVI (SnareVI)	1.3
MultiVI	1.17

Table 3: Batch correction scores

The score for PolyVI is lower than in Table 1 since we are not using protein data. However, the model still does better than MultiVI, likely due to the fact that PolyVI uses a single, unified encoder that can accept both scRNA-seq and scATAC-seq data, whereas MultiVI has two separate encoders, one for each modality.

2.3 Generative Model Analysis

To test the PolyVI generative model, we compared a standard PolyVI model trained on the Mimitou data [14] with a PolyVI model on the same dataset, but corrupted. We 'corrupted' the data by randomly setting genomic regions to 0 with probability 0.3 and adding Gaussian noise to gene and protein counts. Figure 4 shows the comparisons between PolyVI-corrupted and PolyVI-uncorrupted.

Figures 4A-B shows chromatin de-noising (the confidence of the model in confirming an observed 0/1 and that the data point is not a false negative/positive, respectively) in the original and corrupted data. Overall, PolyVI-uncorrupted is slightly more confident in imputing non-zero values. Figures 4C-D show that regardless of data corruption, there is a strong correlation between the log standard deviation of imputed values (the uncertainty of PolyVI) and the deviation from the observed. Figures 4E-F show that the model is able to produce gene counts values close to the actual observed values, even when PolyVI is provided with corrupted data.

In Figure 4 we can see that even though 30 percent of the data has been set to zero, PolyVI is robust enough to be able to predict with strong confidence that much of the data is false negatives, and not true signal. Figure 4 thus shows that the decoders are robust enough to maintain predictive power, even in the presence of increased false negatives.

Next, we show a comparison of the generative models of PolyVI and MultiVI on the 10x granulocyte dataset [15] (Figure 5, has the same type of plots as Figure 4). PolyVI and MultiVI exhibit very similar performance across chromatin accessibility and gene expression. Spearman correlations between "smoothed" gene expression and accessibility data is shown in the table below. Because raw single-cell data can be affected by low sensitivity [2], we smooth the data using a k-nearest neighbors approach (see sections 5.5 and 5.6), identical to the one used in MultiVI [2].

Algorithm	Gene Expression	Chromatin Accessibility
PolyVI	0.83	0.81
MultiVI	0.83	0.82

Table 4: Spearman Correlations for PolyVI and MultiVI

The Spearman correlations are nearly identical for PolyVI and MultiVI. However, Figure 5 implies that PolyVI is marginally more confident in chromatin accessibility values as opposed to MultiVI. Figures 5A and 5B show the models' confidence in imputing a value for

a 0 or 1 in chromatin accessibility, and Figures 5E and 5F show the models' accuracy in imputed gene expression counts; both PolyVI and MultiVI do well. However, PolyVI appears to be more confident and has a lower error than MultiVI in imputing some chromatin values (bottom left corners of 5C and 5D).

Chapter 3

Summary

PolyVI is a set of new additions to the scvi-tools suite, performing variational inference on fully paired DOGMA-seq, ASAP-seq, and SNARE-seq data. When provided fully paired datasets, PolyVI can learn joint latent representations, correct for batch effects, and denoise data. When compared to similar multiomic analysis algorithms, PolyVI does as well and sometimes better than the other algorithms in batch correction and latent separation metrics. Moreover, these other algorithms are unable to replicate the decoder element of PolyVI and of the scvi-tools suite, as only PolyVI can perform denoising on this type of data.

Chapter 4

Methods

PolyVI can model any set of scRNA-seq, scATAC-seq, and cell surface protein data (except for surface protein data alone; 7 total models). The models for scRNA-seq, CITE-seq, and scATAC-seq have all been previously published [12, 6, 3], so in this paper, we introduce models for paired measurements for ASAP-seq, DOGMA-seq, and SNARE-seq (scRNA-seq and scATAC-seq).

4.1 VAE Background

PolyVI uses variational autoencoders (VAEs) to model the likelihood of the single-cell data. Our work focuses on three new models in PolyVI: DogmaVI (gene expression, chromatin, protein), AsapVI (chromatin and protein), and SnareVI (gene expression and chromatin). A diagram for the construction of the DogmaVI model can be seen in Figure 1.

All three models introduced in this paper (DogmaVI, AsapVI, SnareVI) uses VAE noise models [12, 3, 6, 2]. Regardless of the number of modalities, these models use one encoder which maps the data to a multivariate Gaussian 'latent' distribution while accounting for noise, sparsity, and batch effects. Following the VAE loss function, the model is penalized if the model's latent space deviates from the underlying multivariate Gaussian distribution (using a metric for distributional similarity known as the *Kullback-Leibler Divergence*) [10].

The generative part of the model aims to probabilistically generate data from the learned latent space, following a modality-specific distribution. For scRNA-seq data, the model decoder follows a Negative Binomial distribution; for scATAC-seq, the decoder uses a Bernoulli distribution; for surface protein data, the decoder uses a Negative Binomial Mixture model (one component for foreground signal and one for background signal). In addition to the Kullback-Leibler Divergence term, we penalize the model for producing data points that are different from the original data (known as the reconstruction loss).

Combining the KL-Divergence and reconstruction loss terms, we get the overall loss functions for VAEs, known as the Evidence Lower Bound (ELBO) loss, where $p(z)$ is the prior of the variational distribution, q_ϕ is the probability distribution of the encoder (a

Gaussian) controlled by neural network parameters ϕ, mapping the data distribution x to the latent distribution z. p_θ is the distribution of the decoder (Negative Binomial, Bernoulli, Negative Binomial Mixture), controlled by neural network parameters θ.

$$ELBO = E_{z\sim q_\phi(z|x)}[\log(p_\theta(x|z))] + D_{KL}[q_\phi(z|x)||p(z)]$$

The VAE architecture allows PolyVI to learn joint latent representations regardless of the modalities in the sequencing methods, which can be batch-corrected and normalized. In addition, the VAE can generate new data points according to the data distribution.

4.2 The Encoder Model

Across all models, the inference ('encoder') part of the model is the same: a single encoder maps the input data to a multivariate Gaussian distribution, regardless of the number of modalities. The posterior is intractable, so we use the ELBO loss described above to approximate the evidence.

$$p(z) = \mathcal{N}(0, I)$$
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

The generative ('decoder') distributions vary across the VI models.

4.3 The SnareVI Model

SNARE-seq consists of chromatin accessibility and gene expression data, so we have one decoder for each modality.

We use a Bernoulli distribution for chromatin accessibility due to the nature of the data. Chromatin accessibility measures where DNA is (or is not) accessible (i.e., wrapped around histones). Histones 'package' DNA: if the DNA is wrapped around a histone, it is inaccessible (called closed chromatin); if it is unwound, it is accessible (open chromatin). For every cell, we observe whether a genomic region (i.e., 50 base pairs to 100 base pairs) is accessible or not. Chromatin accessibility data is binary, and hence we use a Bernoulli distribution with a learned parameter p to model this data.

Let c, C denote and individual cell and the total number of cells; let j, J denote an individual region and the total number of regions. We have $X_A \in N_0^{C\times J}$, a scATAC-seq data matrix with C rows and J columns. The probability of observing a region as accessible is defined as follows, using the Bernoulli distribution:

$$z_n \sim \mathcal{N}(0, I)$$
$$\rho_{cj} = f_A(z_n, s_n)$$

$$x_A \sim \text{Bernoulli}(\ell_c \rho_{cj})$$

where ρ_{cj} is the true biological noise, f_A is a neural network, s is the batch, and ℓ is the library size.

For gene expression, we use a Negative Binomial model. Unlike chromatin accessibility data, gene expression data is measured in counts: for every cell, we measure the count of each unique gene.

The number of counts of a gene is directly proportional to the rate of transcription: the higher the rate of the DNA to RNA conversion, the higher the gene count. Therefore, we could use a Poisson distribution for gene counts. However, single cell data is often overdispersed. As a result, we use a Negative Binomial model, which is a Poisson distribution whose parameter is sampled from a Gamma distribution [18, 6]. The Negative Binomial has an additional variance parameter that can by learned by the VAE neural network.

$$z_n \sim \mathcal{N}(0, I)$$
$$\rho_{cg} = f_R(z_n, s_n)$$
$$x_R \sim \text{NegativeBinomial}(\ell_c \rho_{cg}, \theta_g)$$

ρ is the learned mean parameter, f_R is a neural network, ℓ is the latent RNA library size, and θ is the learned variance parameter.

4.4 The AsapVI Model

ASAP-seq data combines chromatin accessibility with surface protein data.

AsapVI uses the same Bernoulli model as SnareVI for chromatin accessibility:

$$z_n \sim \mathcal{N}(0, I)$$
$$\rho_{cj} = f_A(z_n, s_n)$$
$$x_A \sim \text{Bernoulli}(\ell_c \rho_{cj})$$

Surface protein data is similar to gene expression data, as we measure the count of each unique protein on the surface of each cell. However, protein expression data follows a different distribution compared to gene expression due to a discrepancy in the sequencing method. Sometimes, a protein that is not present on the surface of a cell gets counted due to the non-specific binding nature of tagged antibodies; this rarely happens for gene counts [6]. Because of this, protein counts have background noise; a key challenge in modelling protein data is distinguishing true negatives from false negatives.

To model this, we use a Negative Binomial Mixture model, with one component for true foreground signal, and one for background noise. Let p, P denote an individual protein and the total number of proteins. and $X_P \in N_0^{C \times P}$, a protein observation matrix with C rows

and P columns. Given z_n (latent representation, a multivariate Gaussian distribution), β_n (protein mean), and s_n (batch), we have:

$$z_n \sim \mathcal{N}(0, I)$$
$$\pi_n = h_\pi(z_n, s_n)$$
$$v_{np} | z_n, s_n \sim \text{Bernoulli}(\pi_{np})$$
$$x_P \sim \text{NegativeBinomial}(v_{np}\beta_{np} + (1 - v_{np})\beta_{np}, \phi)$$

π_{np} is the output of the decoder neural network h, which measures the probability of a protein count coming from either background or foreground. v_{np} is a Bernoulli random variable based on π_{np}, and ϕ is the variance parameter learned by the neural network.

4.5 The DogmaVI Model

DOGMA-seq measures gene expression, chromatin accessibility, and surface protein data simultaneously.

The DogmaVI model is a combination of SnareVI and AsapVI:

$$z_n \sim \mathcal{N}(0, I)$$
$$\rho_{cg} = f_R(z_n, s_n)$$
$$\rho_{cj} = f_A(z_n, s_n)$$
$$x_R \sim \text{NegativeBinomial}(\ell_c \rho_{cg}, \theta_g)$$
$$x_A \sim \text{Bernoulli}(\ell_c \rho_{cj})$$
$$\pi_n = h_\pi(z_n, s_n)$$
$$v_{np} | z_n, s_n \sim \text{Bernoulli}(\pi_{np})$$
$$x_P \sim \text{NegativeBinomial}(v_{np}\beta_{np} + (1 - v_{np})\beta_{np}, \phi)$$

Chapter 5

Benchmarking and Evaluation

5.1 Datasets and Preprocessing

For benchmarking and analysis, we use the ASAP-seq and DOGMA-seq datasets provided in the Mimitou et al [14], downloaded from GEO (Accession number GSE156478). The raw scATAC-seq fragment files were downloaded from the GEO site, and were preprocessed using ArchR [7]. The raw gene and protein count data was used directly in our analysis. In addition, we used the 10x granulocyte dataset [15]. No preprocessing was done here, as peaks were called on the ATAC-seq data.

We used AnnData [9] for our data handling and scanpy [21] for quality control. We align the RNA and ATAC data horizontally in the `adata.X` field, where the rows (cells) are the same. The protein data is placed in the `adata.obsm['protein_expression']` field. Then, depending on the model, we train a VAE on the AnnData-formatted object.

5.2 Using other models

We benchmarked PolyVI against three other models: Schema [16], Seurat_v4 [8], and MOFA+ [1] to compare their batch correction and latent separation scores. For all three models, we followed the documentation provided on their website to train a model on the DOGMA-seq data.

In addition, we compared PolyVI to MultiVI to compare if the 2-encoder approach of MultiVI yielded different results to the 1-encoder approach of PolyVI. We trained models on the Mimitou data to conduct our inference analysis (section 2.2), and trained models on the 10x granulocyte data to conduct our generative analysis (section 2.3).

5.3 Batch Correction Metric

The batch correction metric uses LISI (local inverse Simpson's index) [11]. To test batch correction, we use the 'iLISI' metric, which computes how well 'mixed' two populations are.

First, we define a set number of neighboring samples to observe (similar to k-nearest neighbors). We then compute the inverse Simpson's index is computed as

$$\frac{1}{\sum_{b=1}^{B} P(b)}$$

where $P(b)$ is the batch probability in the context of the neighbors (a Gaussian based distribution of neighborhoods in the latent space) [11]. Effectively, this measures the number of batches in a given neighborhood.

5.4 Latent Separation Metric

The latent cell type separation metric, cLISI uses the exact same formula as iLISI, but instead of specifying batches, we specify the cell types. $\frac{1}{\sum_{b=1}^{B} P(c)}$ is computed, and $P(c)$ is the cell type probability in the local neighborhood.

5.5 Raw Chromatin Accessibility Smoothing

We wanted a simple method to remove the effects of false negative observations in scATAC-seq data, so we use Latent Semantic Analysis to compute a 30-dimensional latent space, where for each cell, we averaged the accessibility profiles of the 50 nearest neighbors. This approach was taken from MultiVI (Methods, [2]).

5.6 Raw Gene Expression Smoothing

For gene expression, we use PCA to compute a 30-dimensional latent space and averaged the profiles over the 50 nearest neighbors, similar to 5.5. This approach was taken from MultiVI (Methods, [2]).

Chapter 6

Figures

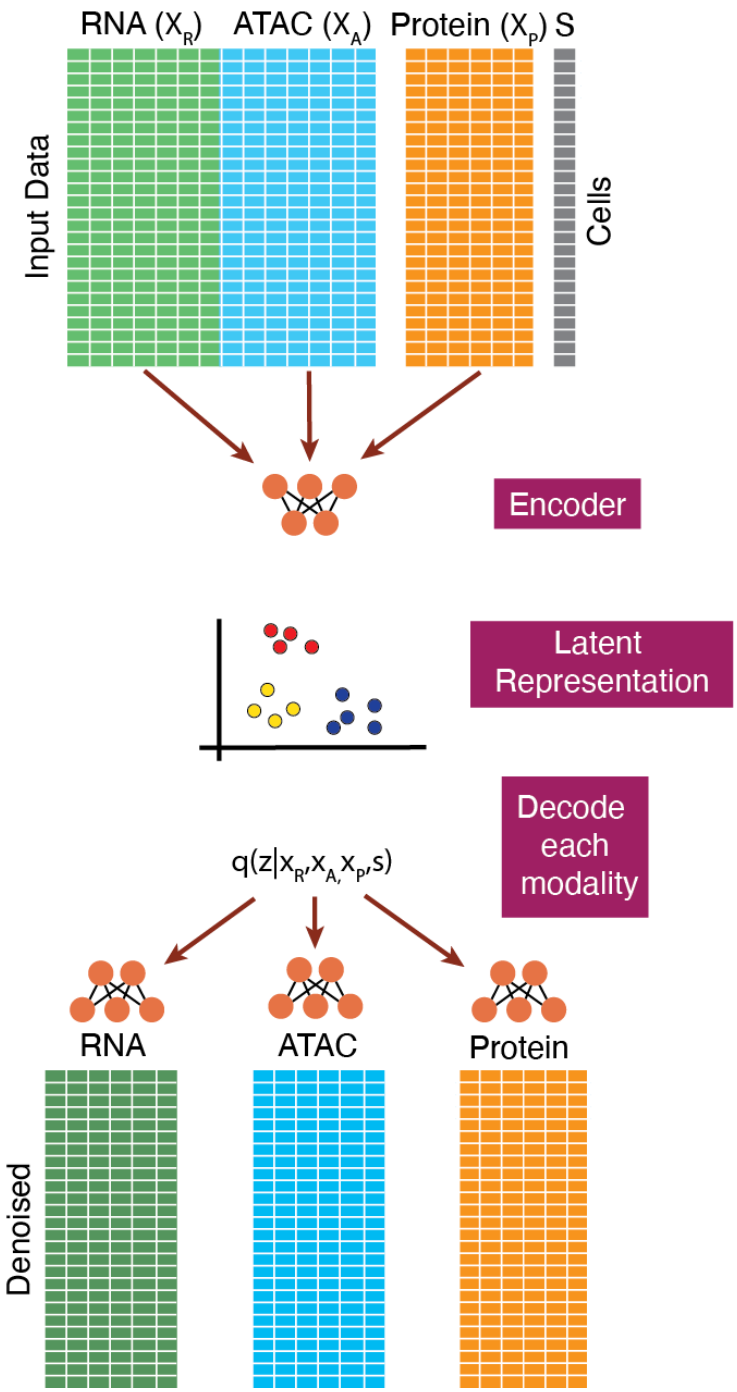

Figure 1: PolyVI (DogmaVI) model overview. Given a unified dataset of gene expression, chromatin, and protein data, a single encoder maps the data to a latent space. To decode and de-noise the data, three decoders are used, one for each modality.

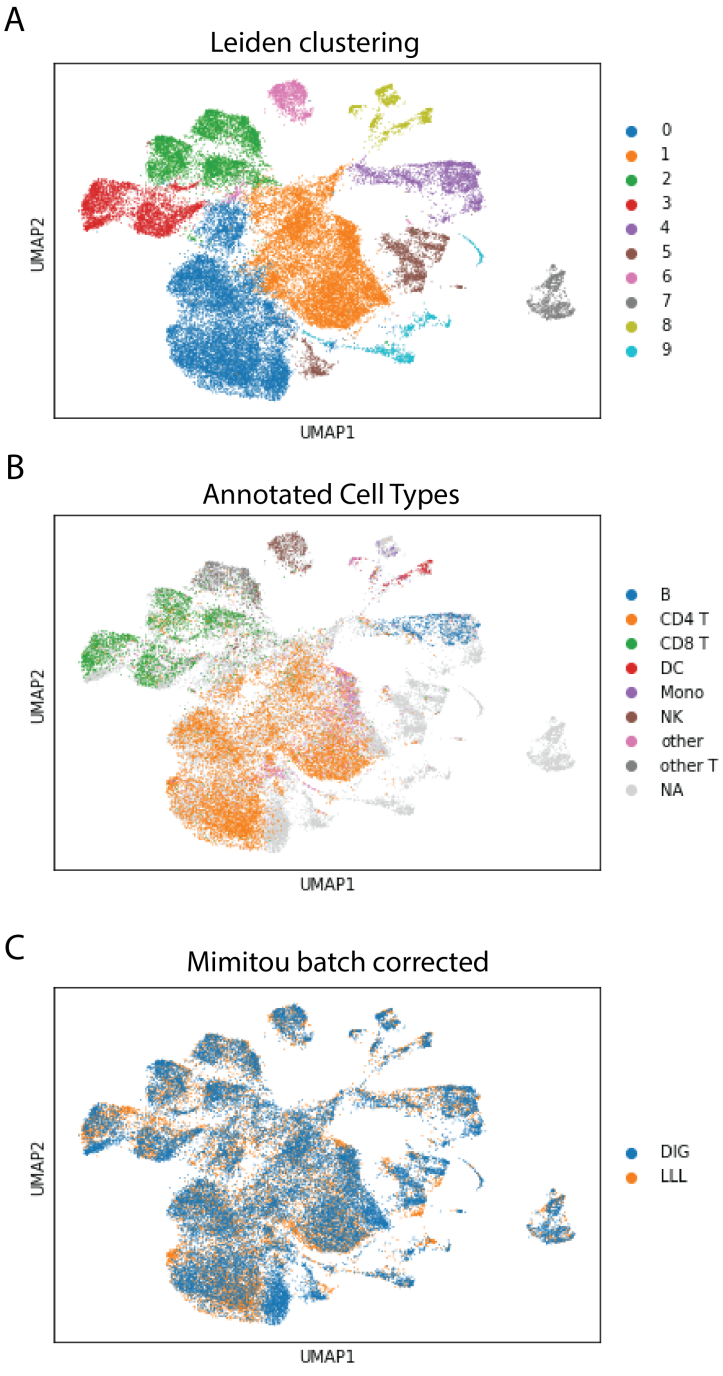

Figure 2: UMAPs of PolyVI latent spaces. A) Latent space with Leiden algorithm clustering. B) Latent space with annotated cell type (given in the dataset) clusters. C) Latent space colored by batch (blue for DIG, orange for LLL).

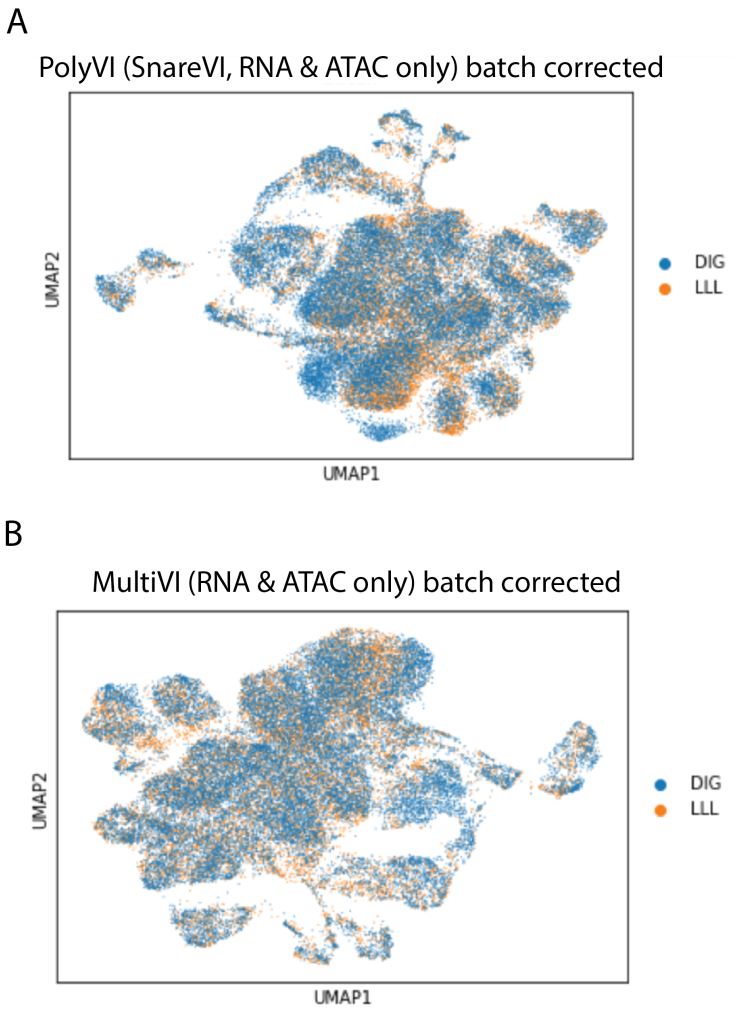

Figure 3: UMAPs of PolyVI and MultiVI batch corrected latent spaces. A) PolyVI (SnareVI, only gene expression and chromatin accessibility data) batch corrected latent space. B) MultiVI batch corrected latent space.

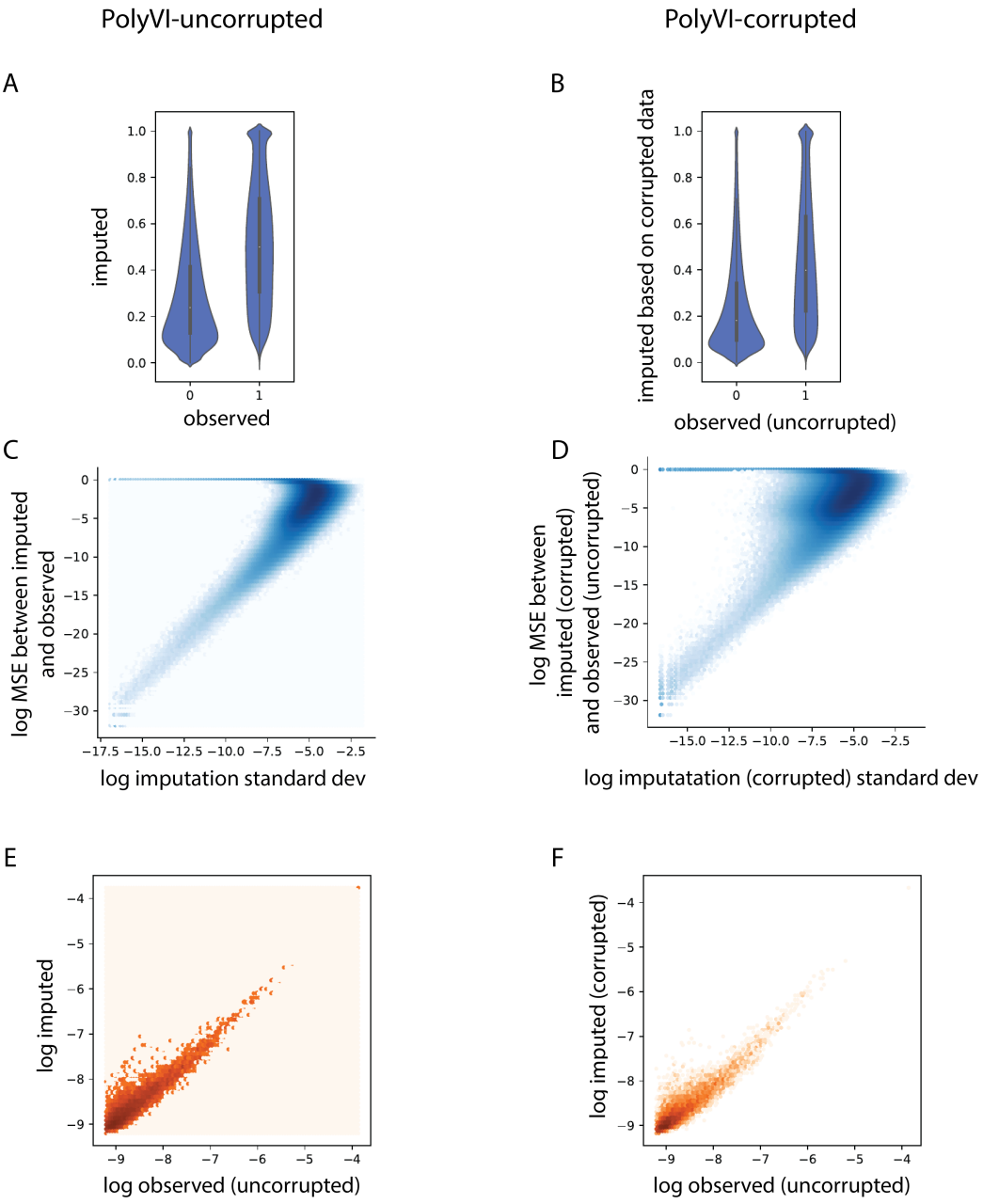

Figure 4: Comparison of PolyVI performance on uncorrupted and corrupted Mimitou data. A, B) PolyVI imputed estimates for chromatin accessibility versus actual observed values for uncorrupted and corrupted, respectively. Panel B compares imputed values of PolyVI (based on corrupted data) versus the ground-truth uncorrupted chromatin accessibility data. C, D) Imputation error (y axis) versus imputation standard deviation for chromatin data for uncorrupted and corrupted, respectively. This measures the model's uncertainty in prediction (the SD) versus the squared error between the imputed and observed values. Panel D's y axis plots the MSE between PolyVI's imputed (based on corrupted data) chromatin data and the uncorrupted ground-truth data. E, F) Logarithmic imputed gene counts versus logarithmic observed gene counts for uncorrupted and corrupted, respectively. Panel F compares imputed values of PolyVI (based on corrupted data) versus the ground-truth uncorrupted RNA count data.

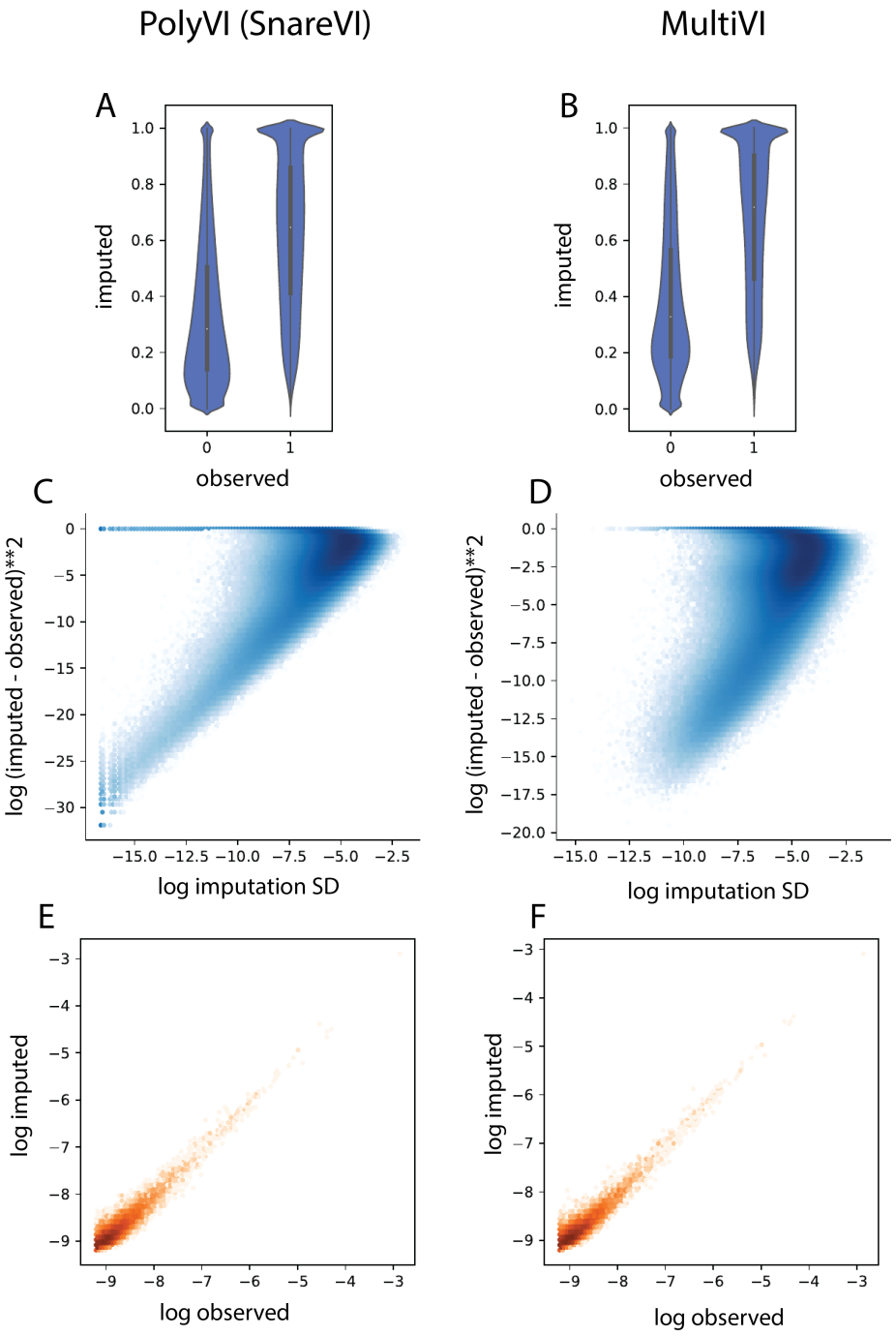

Figure 5: Comparison of PolyVI (SnareVI) and MultiVI on the fully-paired 10x granulocyte dataset. A, B) PolyVI versus MultiVI imputed estimates for chromatin accessibility versus actual observed values. C, D) Imputation error (y axis) versus imputation standard deviation for chromatin data. This measures the models' uncertainty in prediction (the SD) versus the squared error between the imputed and observed values. E, F) Logarithmic imputed gene counts versus logarithmic observed gene counts.

Chapter 7

Declarations

7.1 Acknowledgements

I thank my mentors Tal Ashuach and Professor Nir Yosef for providing guidance and support as I worked through this project. I also thank Justin Hong and Adam Gayoso for scientific discussions and assistance in developing the models.

7.2 Author Contributions

Tal Ashuach (TA) and Nir Yosef (NY) formulated the problem. RVK built the models; RVK tested and evaluated the models. RVK wrote the manuscript. TA and NY edited the manuscript.

Bibliography

[1] Ricard Argelaguet et al. "MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data". In: *Genome Biology* 21.1 (2020). DOI: 10.1186/s13059-020-02015-1.

[2] Tal Ashuach et al. "MultiVI: Deep generative model for the integration of multi-modal data". In: (2021). DOI: 10.1101/2021.08.20.457057.

[3] Tal Ashuach et al. "PeakVI: A deep generative model for Single Cell Chromatin Accessibility Analysis". In: (2021). DOI: 10.1101/2021.04.29.442020.

[4] Jason D Buenrostro et al. "ATAC-seq: a method for assaying chromatin accessibility genome-wide". In: *Current protocols in molecular biology* 109.1 (2015), pp. 21–29.

[5] Song Chen, Blue B. Lake, and Kun Zhang. "High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell". In: *Nature Biotechnology* 37.12 (2019), pp. 1452–1457. DOI: 10.1038/s41587-019-0290-0.

[6] Adam Gayoso et al. "Joint probabilistic modeling of single-cell multi-omic data with totalVI". In: *Nature Methods* 18.3 (2021), pp. 272–282. DOI: 10.1038/s41592-020-01050-x.

[7] Jeffrey M. Granja et al. "ARCHR is a scalable software package for integrative single-cell chromatin accessibility analysis". In: *Nature Genetics* 53.3 (2021), pp. 403–411. DOI: 10.1038/s41588-021-00790-6.

[8] Yuhan Hao et al. "Integrated analysis of multimodal single-cell data". In: *Cell* (2021). DOI: 10.1016/j.cell.2021.04.048. URL: https://doi.org/10.1016/j.cell.2021.04.048.

[9] Isaac Ivirshup et al. "Anndata". In: *GitHub* (). URL: https://github.com/ivirshup/anndata-paper.

[10] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (2013). DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[11] Ilya Korsunsky et al. "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nature Methods* 16.12 (2019), pp. 1289–1296. DOI: 10.1038/s41592-019-0619-0.

[12] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". In: *Nature Methods* 15.12 (2018), pp. 1053–1058. DOI: 10.1038/s41592-018-0229-2.

[13] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: (). DOI: 10.48550/ARXIV.1802.03426.

[14] Eleni P. Mimitou et al. "Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells". In: *Nature Biotechnology* 39.10 (2021), pp. 1246–1258. DOI: 10.1038/s41587-021-00927-2.

[15] "PBMC from a healthy donor - granulocytes removed through cell sorting (10k)". In: *Single Cell Immune Profiling Dataset by Cell Ranger 3.1.0, 10x Genomics* (2019).

[16] Rohit Singh et al. "Schema: Metric learning enables interpretable synthesis of heterogeneous single-cell modalities". In: *Genome Biology* 22.1 (2021). DOI: 10.1186/s13059-021-02313-2.

[17] Marlon Stoeckius et al. "Simultaneous epitope and transcriptome measurement in single cells". In: *Nature Methods* 14.9 (2017), pp. 865–868. DOI: 10.1038/nmeth.4380.

[18] Valentine Svensson. "Droplet scRNA-seq is not zero-inflated". In: *Nature Biotechnology* 38.2 (2020), pp. 147–150.

[19] Fuchou Tang et al. "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature methods* 6.5 (2009), pp. 377–382.

[20] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific reports* 9.1 (2019), pp. 1–12.

[21] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. "Scanpy: Large-scale single-cell gene expression data analysis". In: *Genome Biology* 19.1 (2018). DOI: 10.1186/s13059-017-1382-0.