Statistical Analysis II: Project 1 report

Kornel Howil

December 14, 2022

1 Exploration

a)

The training data contains 72208 observations of 5000 variables. The test data contains 18052 observations of 5000 variables.

b)

The figures below show histograms of the training data.

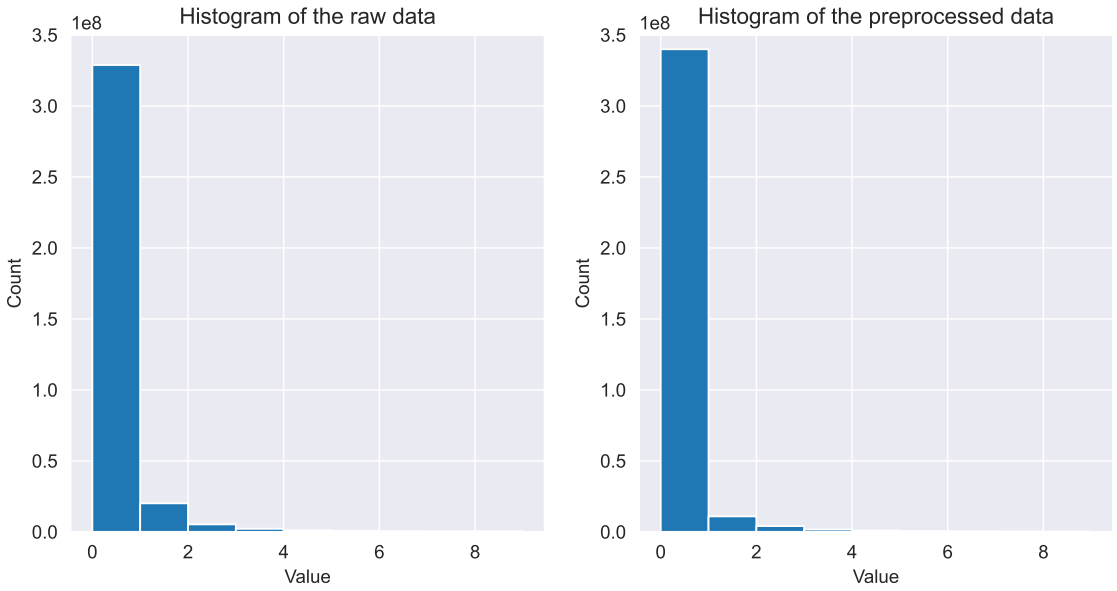


Figure 1: Histograms of the raw data (left) and the preprocessed data (right).

c)

Each observation in the preprocessed data corresponds to the same row in raw data multiplied by a constant number. According to the description of the dataset [1], this constant number is a size factor calculated using *scran* [2].

d)

The figures below show histograms of the training data after removing all zero values.

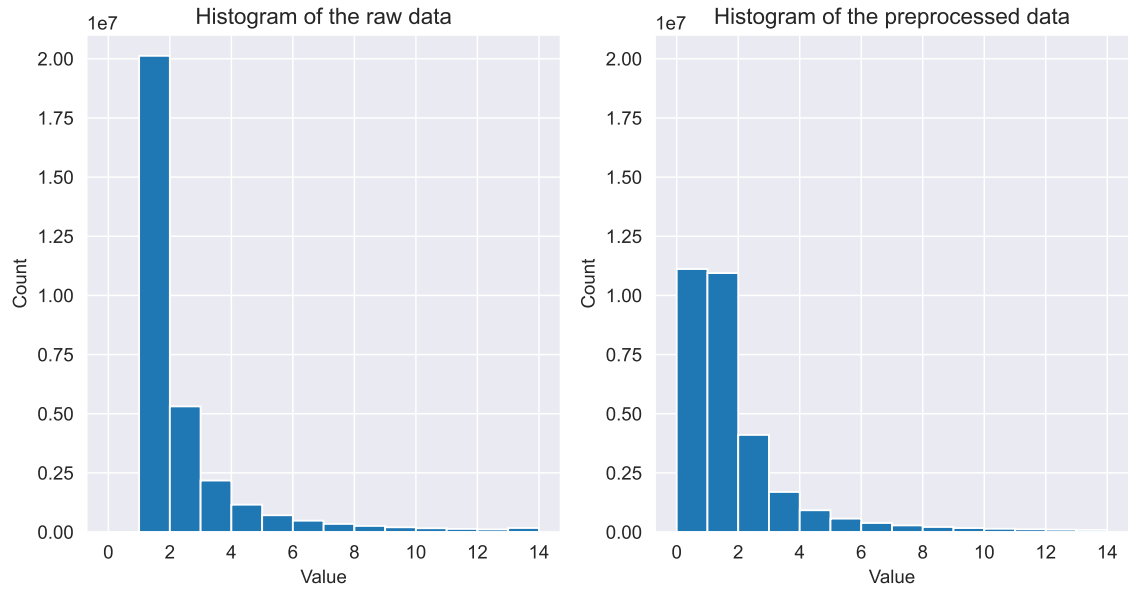


Figure 2: Histograms of the raw data (left) and the preprocessed data (right) after removing all zero values.

e)

The data comes from a distribution with an abundance of zeros. A raw dataset consists of integers hence data may come from a geometric distribution.

f)

`adata.obs` contains gene expression observation metadata i.e. data about the gene expression dataset [1]. From this metadata, we can learn that the dataset consist the data from

1. 4 labs (`andata.obs["Site"]`)
2. 8 patients (`andata.obs["DonorID"]`)
3. 45 cell types (`andata.obs["cell_type"]`)

2 Vanilla VAE training

a)

In a vanilla VAE implemented during Lab 6 and Lab 7, the encoder (Tab **??**) returns `latent_dim` means and `latent_dim` variances. Then, by using reparameterization trick, values from the latent space are sampled from normal distributions defined by the output of the encoder. For the decoder, we assumed that each value of scRNA-seq data comes from a gaussian distribution with variance equal to one and mean equal to the output of the decoder. Loss is defined as

$$\text{Loss}(Data) = -\text{ELBO} = \sum_{i=1}^{5000} \log \text{Normal}(x_i, \mu_i) - \beta \cdot \texttt{KL}(N(z), N(0, \mathbf{1})), \tag{1}$$

where $Data = (x_1, x_2, ..., x_{5000})$, $z = (\mu, \sigma)$ is a vector of size `latent_dim` sampled from distribution defined by an output of the encoder and $\log \text{Normal}(x_i, \mu_i)$ is log likelihood of a value x_i coming from normal distribution with mean μ_i (output of the decoder) and variance 1.

Layer	Type	Activation	Input	Output
1	Full		5000	2000
	BN			
	Dropout	ReLU		
2	Full		2000	1500
	BN			
	Dropout	ReLU		
3	Full		1500	1000
	BN			
	Dropout	ReLU		
4.1	Full		1000	`latent_dim`
4.2	Full	Softplus $+ \varepsilon$	1000	`latent_dim`

Table 1: The architecture of the encoder. The activation function was applied after a given layer. The output of layer 4.1 is a mean of the distribution in the latent space and the output of layer 4.2 is a variance of the distribution in the latent space. Constant value ε was set to 10^{-4}.

Layer	Type	Activation	Input	Output
1	Full		`latent_dim`	1000
	BN			
	Dropout	ReLU		
2	Full		1000	1500
	BN			
	Dropout	ReLU		
3	Full		1500	2000
	BN			
	Dropout	ReLU		
4	Full	ReLU	2000	5000

Table 2: The architecture of the decoder. The activation function was applied after a given layer.

On the Figure 3 there is a learning curve of a Vanilla VAE trained with $\beta = 1$ and `latent_dim` $= 50$. Figure 4 shows reconstruction and regularization losses of the same model.

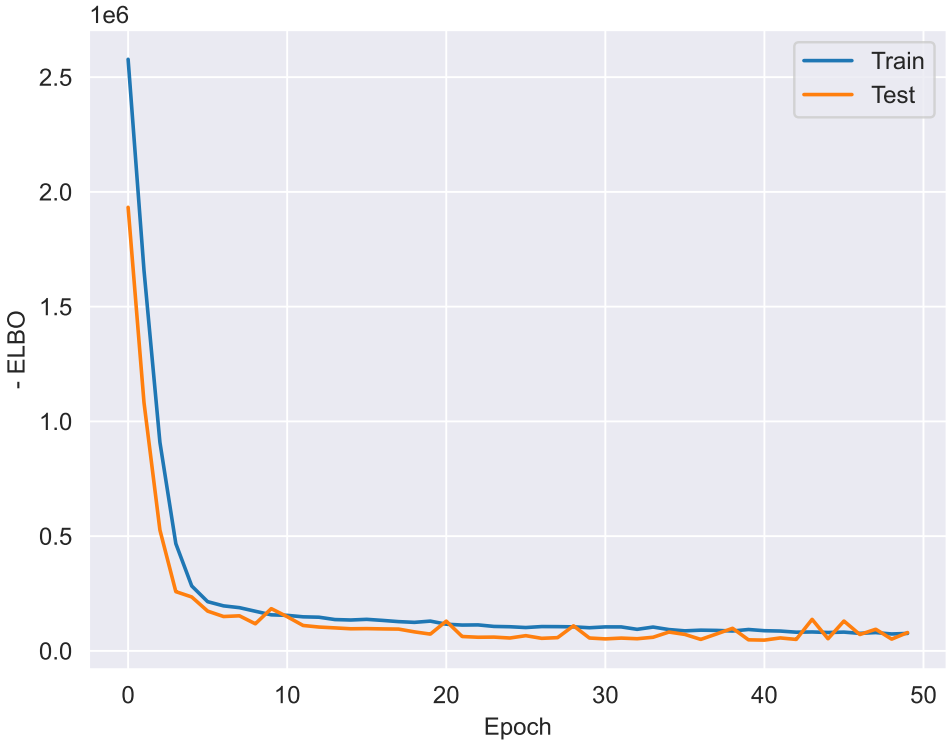


Figure 3: Learning curve of Vanilla VAE trained with $\beta = 1$ and `latent_dim` $= 50$.

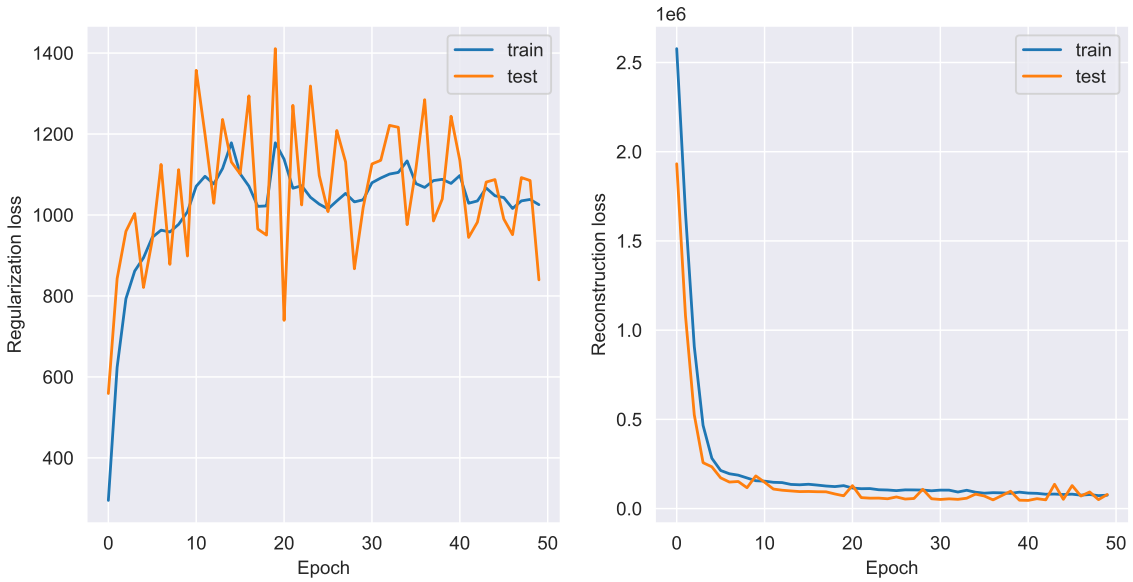


Figure 4: Regularization and reconstruction losses of Vanilla VAE trained with $\beta = 1$ and `latent_dim` $= 50$.

b)

PCA was fitted on a test set encoded using a trained model using $\beta = 1$ and `latent_dim` $= 50$. More than 95% of the variance is explained by at least 4 PCA components. Figure 5 shows how many PCA components is needed to explain a given amount of variance.

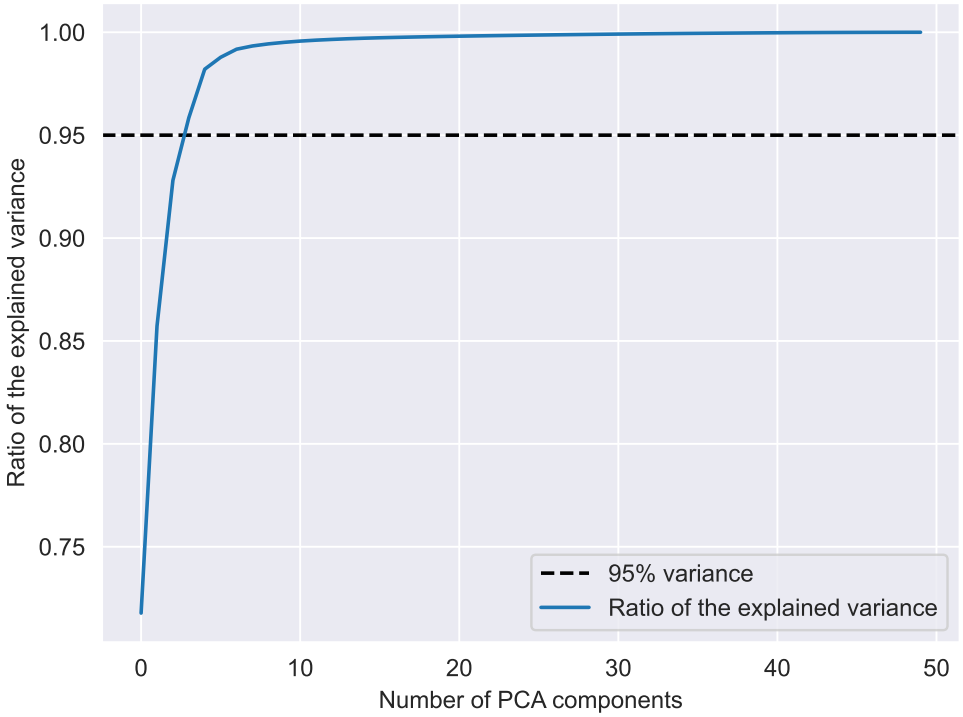


Figure 5: Ratio of explained variance in function of PCA components. PCA was fitted on a test set using Vanilla VAE trained with $\beta = 1$ and `latent_dim` $= 50$.

To check how `latent_dim` affects -ELBO, a model was trained 3 times using different values of `latent_dim`. Results of the final performance on a test set are shown in the table **??**.

`latent_size`	**-ELBO**
50	$\sim 7.6 \cdot 10^4$
10	$\sim 7.0 \cdot 10^4$
5	$\sim 7.5 \cdot 10^4$

Table 3: The architecture of the decoder. The activation function was applied after a given layer.

c)

Figures 6, 7 and 7 show the test set encoded using three models with different `latent_dim`, projected on the top two PCA components.

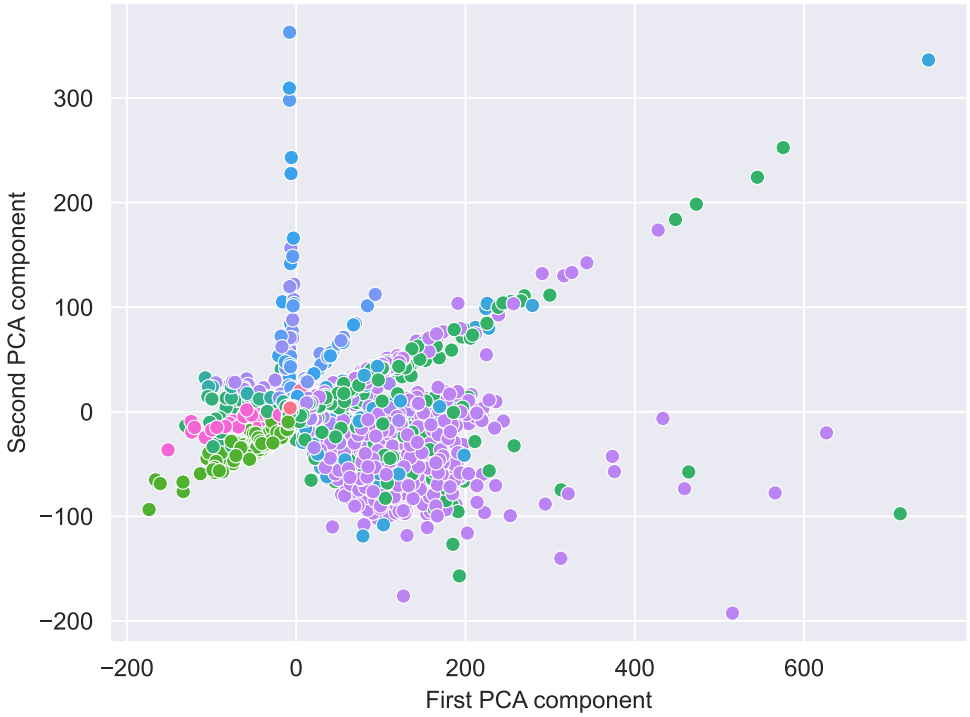


Figure 6: Encoded test set projected on the top two PCA components. Model trained with $\beta = 1$ and `latent_dim` $= 50$. Colours represent different cell types.

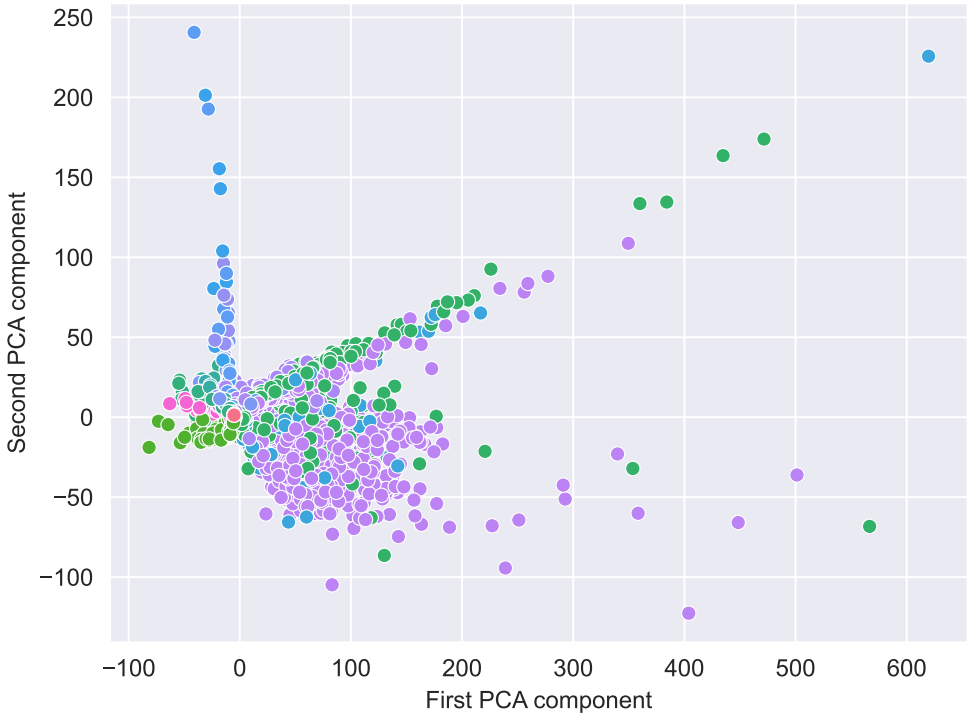


Figure 7: Encoded test set projected on the top two PCA components. Model trained with $\beta = 1$ and `latent_dim` $= 10$. Colours represent different cell types.

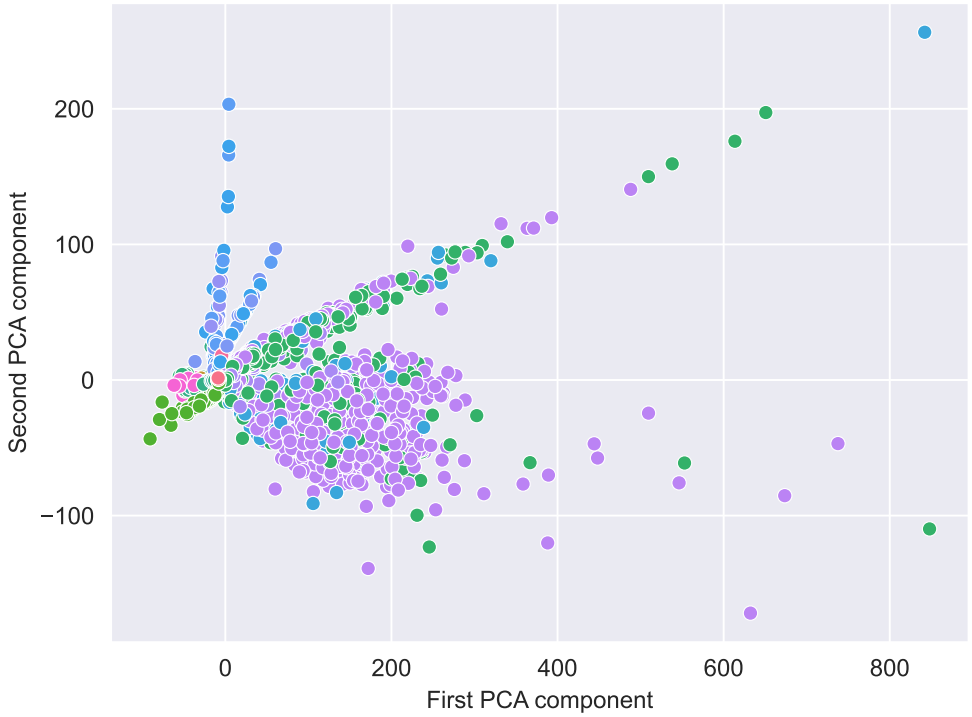


Figure 8: Encoded test set projected on the top two PCA components. Model trained with $\beta = 1$ and `latent_dim` $= 5$. Colours represent different cell types.

d)

For all models I have used a raw dataset. I am not sure why but when I trained my network on a preprocessed data, a test loss was much higher than a train loss. I think that this may be due to the specific normalization technique which was used on this data.

3 Custom decoder

a)

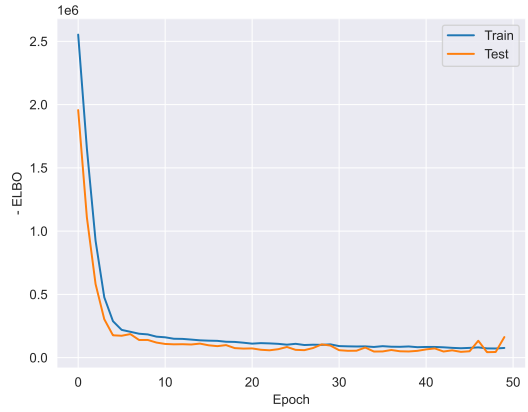
Since the raw dataset (which was used in training) cointains only integer values, I used Poisson distribution to calculate log likelihood of data knowing output of the decoder. New loss was defined as

$$\text{Loss}(Data) = -\text{ELBO} = \sum_{i=1}^{5000} \log \text{Poisson}(x_i, \lambda_i) - \beta \cdot \texttt{KL}(N(z), N(0, \mathbf{1})), \tag{2}$$

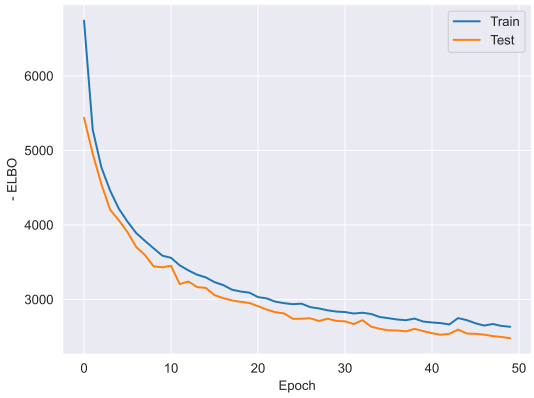
where λ_i (output of the decoder) is a rate defining Poission distribution.

b)

VAE with custom decoder was trained on the raw dataset using $\beta = 1$ and `latent_dim` $= 10$. Size of the latent space was chosen to be the same as in best Vanilla VAE model. Figure 9 shows learning curves for both best Vanilla VAE model and VAE trained with custom decoder.



(a) Vanilla VAE



(b) VAE with custom decoder

Figure 9: Learning curves for both Vanilla VAE (left) and VAE with custom decoder (right). Both models were trained using $\beta = 1$ and `latent_dim` $= 10$.

Vanilla VAE has a steeper learning curve in the first few iterations but later learning is rapidly slowing down. On the other hand VAE with a custom decoder is learning in more stable way. The most important part of comparison of this to models is value of -ELBO. Figure 9 clearly shows that -ELBO of VAE with a custom decoder is over 10 times smaller than -ELBO of the model with basic gaussian decoder. The new decoder is clearky better.

c)

Figure 10 shows encoded test set projected on the top two PCA components of the model with a gaussian decoder and the model with a custom decoder.

(a) Vanilla VAE

(b) VAE with custom decoder

Figure 10: Encoded test set projected on the top two PCA components of the Vanilla VAE (left) and the VAE with custom decoder (right). Colours represents different cell types.

It is clearly visible that these two plots differ from each other much more that plots for Vanilla VAE with different size of the latent space (Figures 5, 6, 7). Range of the values on the top two PCA components is smaller for the VAE with custom decoder.

4 Adjusting VAE for batch effect

a)

References

[1] Description of the dataset, *https://openproblems.bio/neurips_docs/data/dataset/*

[2] L. Lun, A.T., Bach, K. & Marioni, J.C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol 17, 75 (2016), *https://doi.org/10.1186/s13059-016-0947-7*