# Project nr 1

## Statistical Data Analysis 2

Mateusz Kapusta

December 15, 2022

## 1 Data exploration

In the project we will use gene expresion matrix with each with measure of mRNA activity. Each row coresponds to one of the cells while columns represent diffrent genes. In train set we have 72208 cells with expresion values for 5000 genes while test set consists of 18052 cells also with vales for 5000 diffrent genes. Values of mRNA expression are depicted on the figure 1.

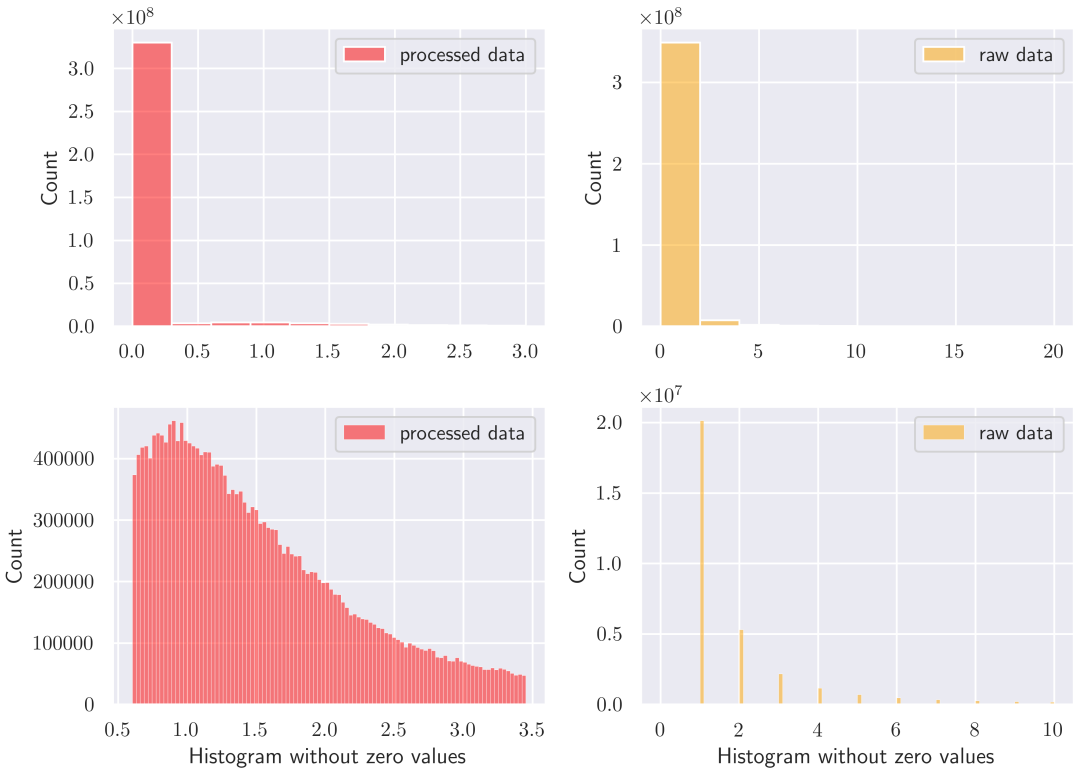



Figure 1: Histogram representing distribution of data in matrices.

We can see that most values of our data is zero, in fact nearly 91% of our values are zero. This fact have biological reason, it means that we observed our cell when it was inactive, in order to see any expresion special circumscences must occure. We have also most important information about each cell mainly about donor (sex,race etc.) but also information about batch. There are 12 unique batches, this is something we need to be carefull later. Each batch number represent unique combination of patient number and lab id, there are total 10 patients that were examiend in 4 diffrerents labs. There are also 45 unique cell types. Data is presented in two versions, raw and preprocessed one. Raw data is purly discrete while preprocessed data is continuous. Data was propably normalized to 10k counts but not transformed using function log1p. Also data wasn't normalized to unit variance what can be easly shown when computing standard deviation of preprocessed data. What should be mentioned preprocessed data has mean 3.46 and standard deviation of 1436, it's more then values of raw data for which those two parameters are equal 0.44 and 34, also range of raw data is smaller then range of preprocessed one. Data does not follow any easly described distribution, it's heavly zero inflated and does not follow popular discrete distributions like Poisson distribution.

## 2 Simple VAE example

Variational autoencoder is a type of generativ model using ANN to model propabilistic distribution of data. Lets consider Bayesian graphical model as on the figure 2.

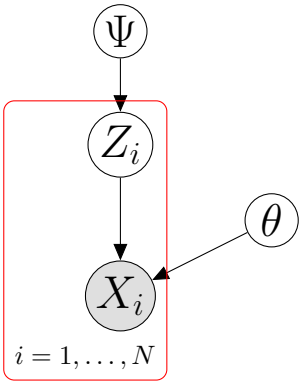



Figure 2: Bayesian net model for variational autoencoder

In our basic case we have latent space $\mathbf{Z}$ which have prior $\Psi$. We assume gaussian prior $\mathbf{\Psi} \sim \mathcal{N}(0, \mathbf{1})$, bold font stands for n-th dimensional shape. We denote $\theta$ parameters of the decoder which computes parameters of probability of $\mathbf{X}$. In case of VAE stricly speeking we are modeling $P(\mathbf{X}|\mathbf{Z}, \theta)$ using suitable neural netowork. We also want to aproximate posterior distribution $p_{\theta,\psi}(\mathbf{Z}|\mathbf{X}) \approx q_{\varphi}(\mathbf{X}|\mathbf{Z})$

using other neral network called encoder denoted by $\varphi$. We will select such models that

$$q_\varphi(\mathbf{Z}|\mathbf{X}) \sim \mathcal{N}(f_\varphi(\mathbf{X}), h_\varphi(\mathbf{X})) \tag{1}$$

where $f$, $h$ and $g$ functions modeled by neural networks. We constrain posterior probability to normal distribution with diagonal covariance matrix. On labolatory we used two reconstruction losses: MSE and binary crossentropy. Both losses have some background in propabilistic approach. Lets write loss function (-ELBO) for our case:

$$\mathcal{L}(\mathcal{D}) = -\mathbb{E}_{q_\varphi} \log P_\theta(\mathbf{X}|\mathbf{Z}) + D_{KL}(q_\phi(\mathbf{Z}|\mathbf{X})|\mathcal{N}(0, \mathbf{1})) \tag{2}$$

. We want to assume, that the expression of each gene is independent from each other so $P(X_1, \ldots, X_n) = P(X_1) \cdot \ldots \cdot P(X_n)$. Moreover we want neural network to produce $n$ sets of parameters $\mu_i$ that will be parametrizing those distribution. If we assume that our distribution is

$$p(X_i|\mu_i) = \mathcal{N}(\mu_i, \sigma) \tag{3}$$

where $\sigma$ is simple hyperparameter. We then have

$$\log P(\mathbf{X}|\mathbf{Z}) = \sum \frac{(\mu_i(\mathbf{Z}) - X_i)^2}{2\sigma^2} + C. \tag{4}$$

Hence we can see that when we assume gaussian probability model our approach will be equal to minimalizing MSE reconstruction loss if $\sigma = \frac{1}{\sqrt{2}}$. $\sigma$ is hyperparameter and hence can be chosen freely allowing us to control relative importance of reconstruction loss and KL divergence. In our decoder $\sigma = 1$ will be used as default value of standard deviation.

## 2.1 Implementation

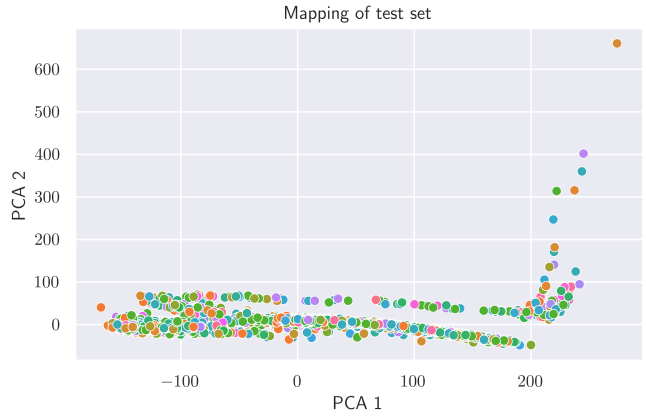
In order to implement variational autoencoder Pyro package was used. In this particular VAE pre-processed data was used as it have continuous distribution so is more sutable to model with gaussian distribution then raw thata that takes discrete values only. Each layer of neural network consist of dense layer followed by batch normalization and ReLu activation layer with dropout $p = 5\%$. This setup prevents VAE from overfitting and also allows to deal with numerical problems with variance.

Decoder and encoder consisted of 2 hidden layers with 500 and 200 hidden neurons (in different order). Encoder was also supplied with normalization layer which preprocessed data to have zero mean and unit variance. As encoder need to output standard deviation of gaussian distribution one of the two outputs was then passed through ReLU function. Decoder output was also passed through ReLU function as we deal with positive data only. VAE was trained using batch size equal to 128 with learning rate $3 \cdot 10^{-3}$ for 600 epoches. Learing rate was multiplied by factor of 0.6 after 100 epoches allowing to achive better accuracy over time. In order to find out optimal hyperparameters such as latent size small grid search was performed using 5 latent space dimensions.Values of $-ELBO$ loss on test set after 200, 400 and 600 epoches is reported together with explained variance ratio for latent space by three first components of PCA (denoted as $\sigma_{PCA_i}$) in table 1.

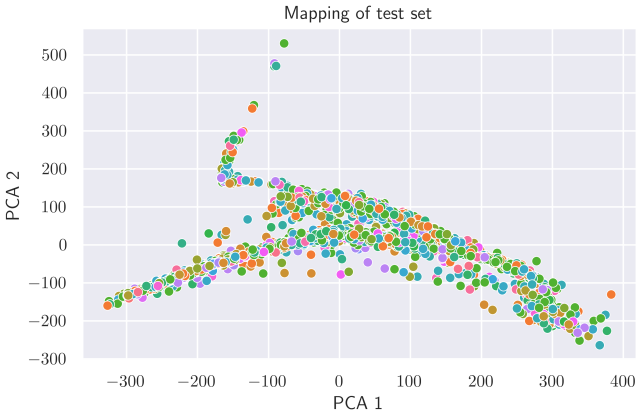
| Latent space size | $-ELBO_{200}$ $[10^9]$ | $-ELBO_{400}$ $[10^9]$ | $-ELBO_{600}$ $[10^9]$ | $\sigma_{PCA_1}$ | $\sigma_{PCA_2}$ | $\sigma_{PCA_3}$ |
|---|---|---|---|---|---|---|
| 50 | 9.532 | 8.564 | 8.243 | 0.682 | 0.160 | 0.088 |
| 70 | 9.526 | 8.490 | 8.246 | 0.621 | 0.261 | 0.072 |
| 100 | 9.508 | 8.568 | 8.231 | 0.761 | 0.132 | 0.072 |
| 120 | 9.643 | 8.634 | 8.252 | 0.809 | 0.122 | 0.048 |
| 150 | 9.629 | 8.531 | 8.160 | 0.535 | 0.380 | 0.04 |

Table 1: ELBO values for test set together with ratio of variance explained with PCA components

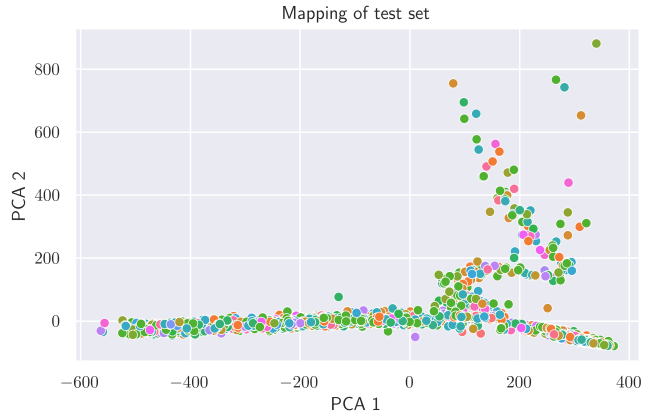
We can see, that in all cases 95% of variance is achived when we use just using 3 PCA components (except for latent size 50 when we need 4 components). We can infer from data, that latent space does not affect values of ELBO so much. What we need to adress are huge values of our loss function. This output is not proportional to MSE loss as we also have rather big constant resulting from $\log 2\pi\sigma^2$ which is also included. Morover we assumed that our $\sigma$ is just equal to 1, assumption which looks rather wrong when we come back to parameters of our data, standard deviation of test set is more then thousend. More serious approach should deal with this problem and come with more realistic value of $\sigma$. After PCA fit each test data was mapped onto first two components of PCA and then colored using cell type, relevant plots are presented on figure 3.



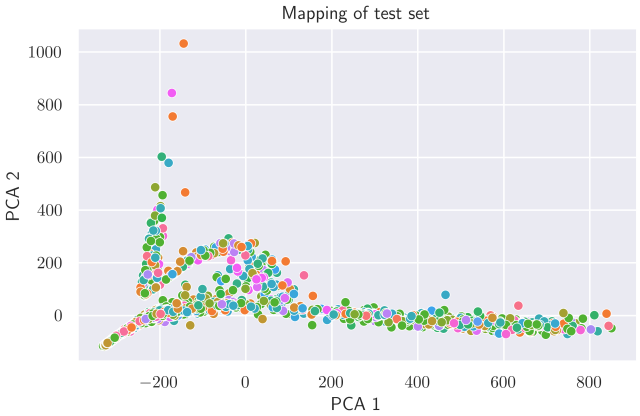

(a) Latent size 50



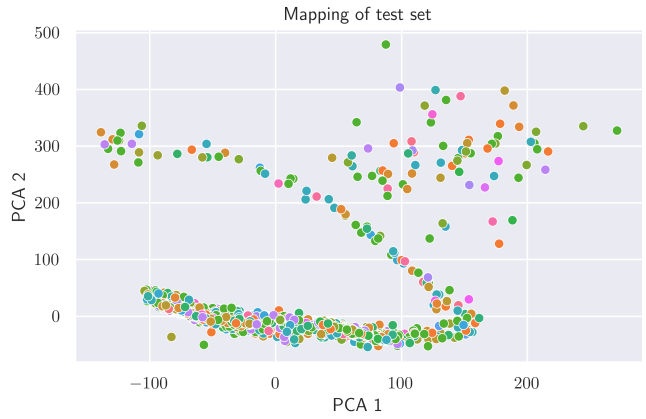

(b) Latent size 70




(c) Latent size 100




(d) Latent size 120




(e) Latent size 150

Figure 3: PCA mapping of test set for Gausian decoder.

From all of the models it was decided to choose model with laten space size 100 due to the low ELBO values and high ratio of explained variance. Learing curve of the model is presented on the figure 4.

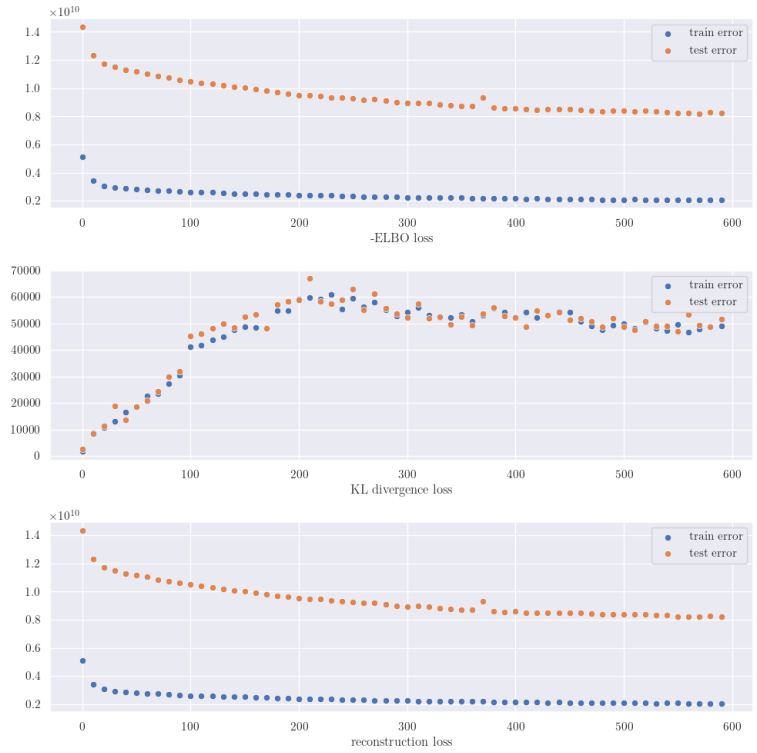



Figure 4: Learing curve for Gaussian VAE

We can see, that hyperparameters $\sigma$ which we are using lead to reconstruction loss dominating ELBO. This have rather bad efffect on our VAE, due to this behaviour KL loss part is rising making our VAE more similar to normal autoencoder. We can also clearly see, that test loss is circa 3 times higher then train error. This behaviour is very suspicious as it indicates that propably test data was preprocessed using different values then train data. PCA was trained od train data and then subsample of test data was mapped onto two first axis of PCA (2500 subsamples). Resulting plot is presented on the figure 5.

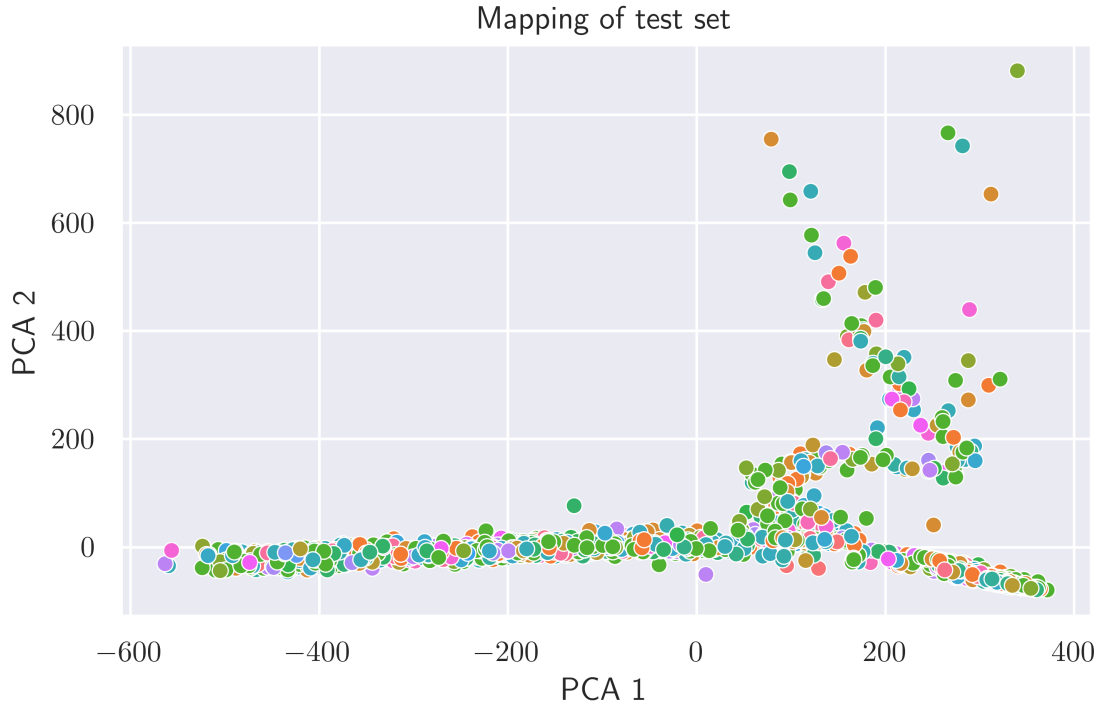



Figure 5: Mapping of test data onto first two components of PCA trained on train set.

# 3 Negative Binomial VAE

In order to model our data more accurate we want to model not preprocessed data but raw one. One of the distributions which can be used to model discrete data is poisson distribution. We can clearly see, that mean is not equall to variance of our distribution which is true for poisson distributed data. Instead of Poisson distribution one can use zero inflated one or negative binomial distribution. We will use second option which is parametrised by 2 parameters, number of total counts $r$ and probability $p$. We model bouth values using neural network using setup similar to encoder, last hidden layer is connected to 2 output layers, one for parameter $t$ and second for parameter $p$. Log probability of our distribution is equal to

$$\log P(x = k|r_i, p_i) = \log \left( \binom{k+r-1}{k} (1-p)^k p^r \right) \tag{5}$$

.

## 3.1 Implementation

This type of VAE was implemented using very similar setup as in previous subtask with main differences in decoder. Instead of big decoder with two hidden layers with 200 and 500 neurons setup was swiched to one with one hidden layer with 200 neurons. Hidden layer was connected to 2 outputs layers modeling $r$ and logits of $p$ parameter of Negative Binomial distribution. $r$ parameter was forced to be positve using ReLU function. Encoder was adopted from Gaussian VAE, only diffrence was GELU activation function which was used instead of ReLU (there was also normalizing layer!). Also training process was similar as in the case of Gaussian VAE, only value of learning rate was 3 times smaller at the start. As declared VAE was trained using raw data instead of preprocessed one. In order to find optimal size of latent space search similar to one with gaussian decoder was performed and results can be found in table below.

| Latent space size | $-ELBO_{200}$ | $-ELBO_{400}$ | $-ELBO_{600}$ | $\sigma_{PCA_1}$ | $\sigma_{PCA_2}$ | $\sigma_{PCA_3}$ |
|---|---|---|---|---|---|---|
| 70 | 1425 | 1423 | 1421 | 0.802 | 0.078 | 0.027 |
| 100 | 1438 | 1434 | 1432 | 0.768 | 0.093 | 0.034 |
| 120 | 1429 | 1424 | 1423 | 0.807 | 0.065 | 0.027 |

Table 2: ELBO and PCA variance ration values for Negative Binomial VAE.

After considorations it was decided to use latent size 100. Learning curve for training is presented at figure 6.

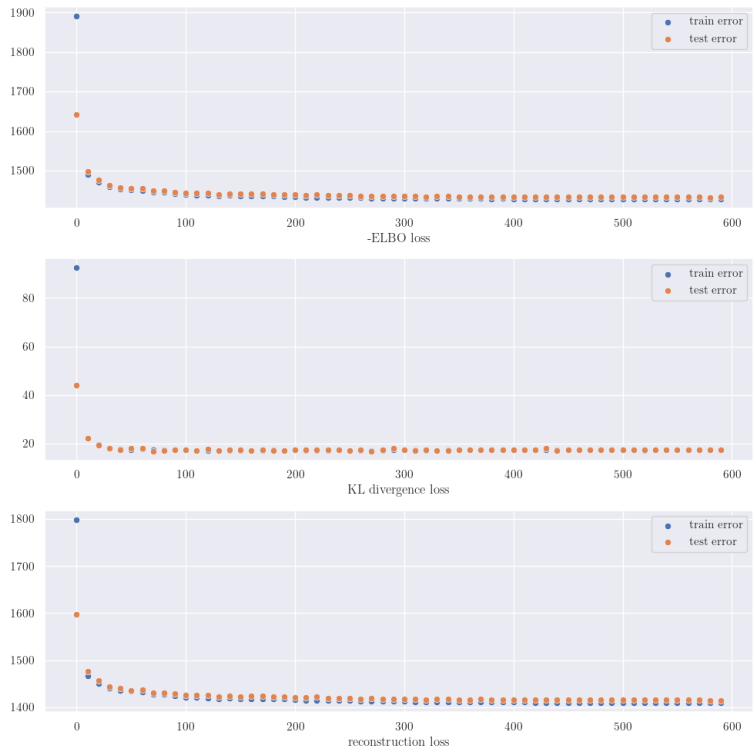



Figure 6: Learing curve for Nagative Binomial VAE.

Few things can be noticed as we can clearly see, that both training and test data give similar error which solves problem that we encountered in previous section. KL loss is very stable and is decreasing during training process contrary to Gaussian VAE. Also values of $-ELBO$ are much smaller which tells us that our model is better description of reality. We can also compare PCA fits of latent spaces for both decoders which can we see on 7.

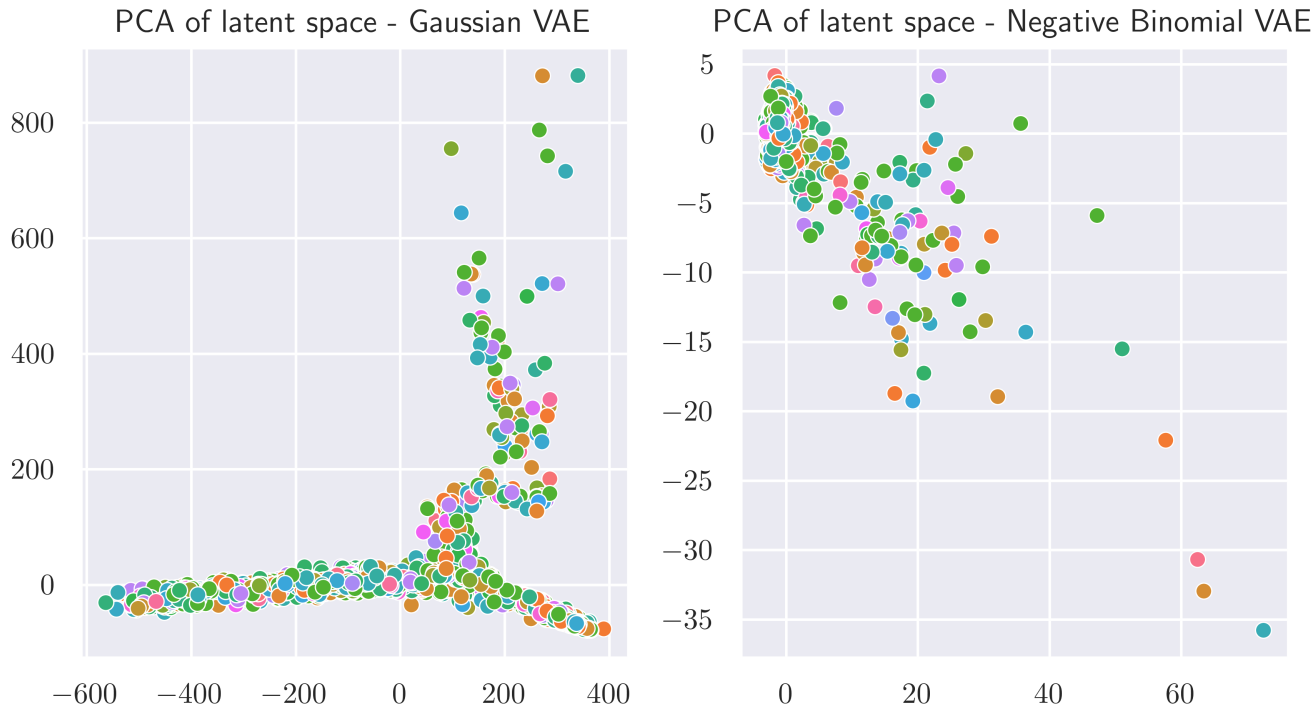



Figure 7: Comparsion of PCA fits.

PCA fit on our Binomial VAE is rather good, we need 5 or 6 components to recover 95% of variance. There are many diffrences between those two plots, second one seem to be much more concentrated. Lets also color our data by batch information. Relevant plot is presented on figure 8

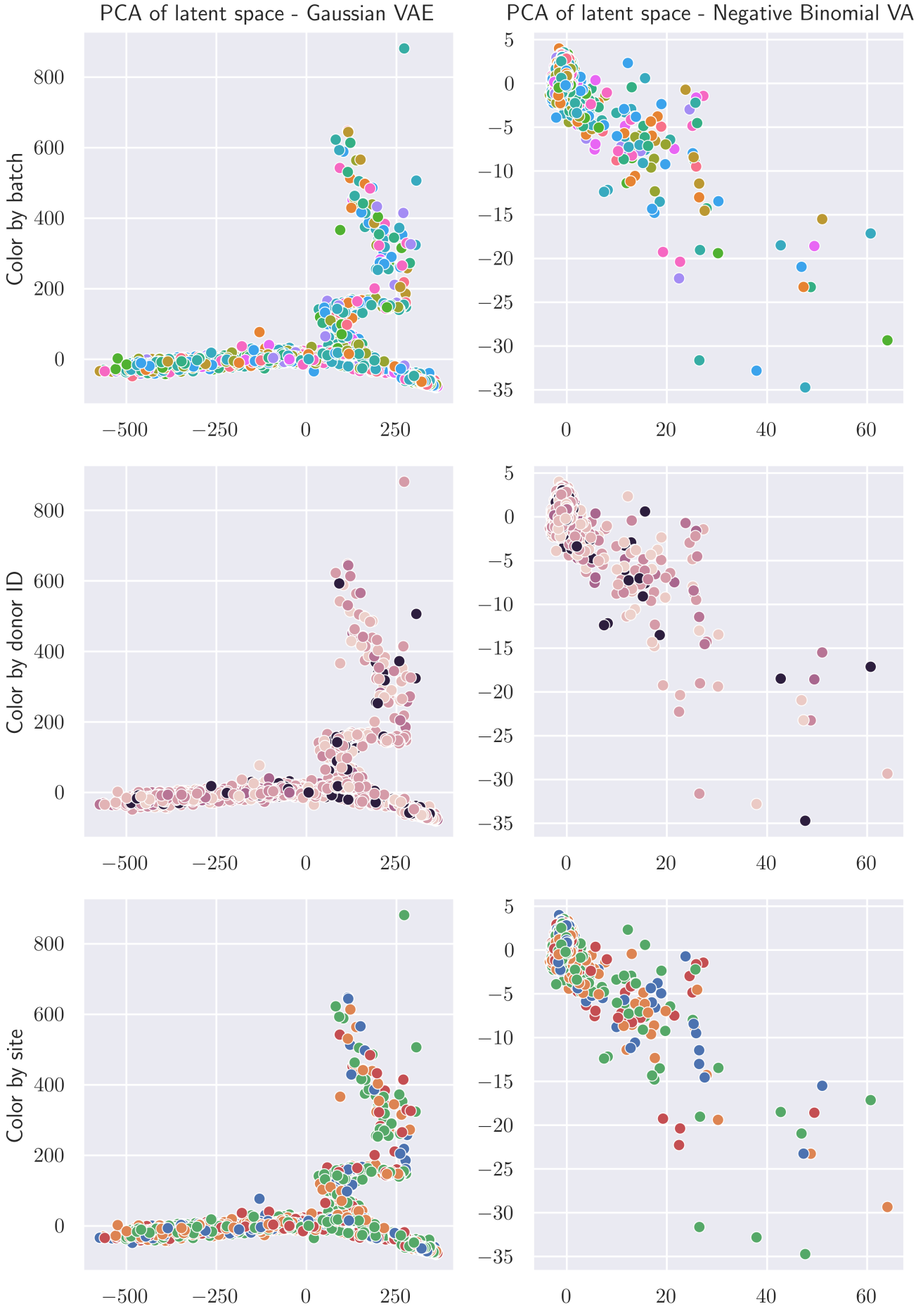



Figure 8: Comparsion of PCA fits with batch information.

# 4 Batch adjusted VAE

Let's consider model presented on figure 9.

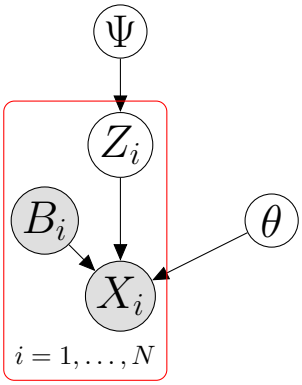



Figure 9: Bayesian net model for batch adjusted variational autoencoder

Our VAE have additional dependence on varible $B_i$ which represents batch observation. In practice we contancate tensor of one-hot encoded batch labs with latent space vector and pass it through decoder in order to obtain parameters of our distribution. Here we will use also negative binomial distribution as it's working well with our raw data.

## 4.1 Implementataion

In order to include batch id in our data we one-hot encode vector of lab asignments ant then contancat it with latent vector before passing it through decoder. Setup from previus task was adopted with all parameters except that decoder at the entry layers has increased width by 12. Model was also trained using raw data for 600 epoches. Size of latent space was also found using small grid search, values for different batch sizes are presented in table 3.

| Latent space size | $-ELBO_{200}$ | $-ELBO_{400}$ | $-ELBO_{600}$ | $\sigma_{PCA_1}$ | $\sigma_{PCA_2}$ | $\sigma_{PCA_3}$ |
|---|---|---|---|---|---|---|
| 70 | 1413 | 1409 | 1409 | 0.898 | 0.047 | 0.022 |
| 100 | 1407 | 1404 | 1403 | 0.865 | 0.065 | 0.014 |
| 120 | 1415 | 1411 | 1410 | 0.859 | 0.082 | 0.017 |

Table 3: ELBO and PCA variance ration values for batch adjusted VAE.

It can be easly seen, that latent size 100 is best and in each case 3 pca components capture 95% of variance. PCA plot of Learing curve is shown on the figure 10.

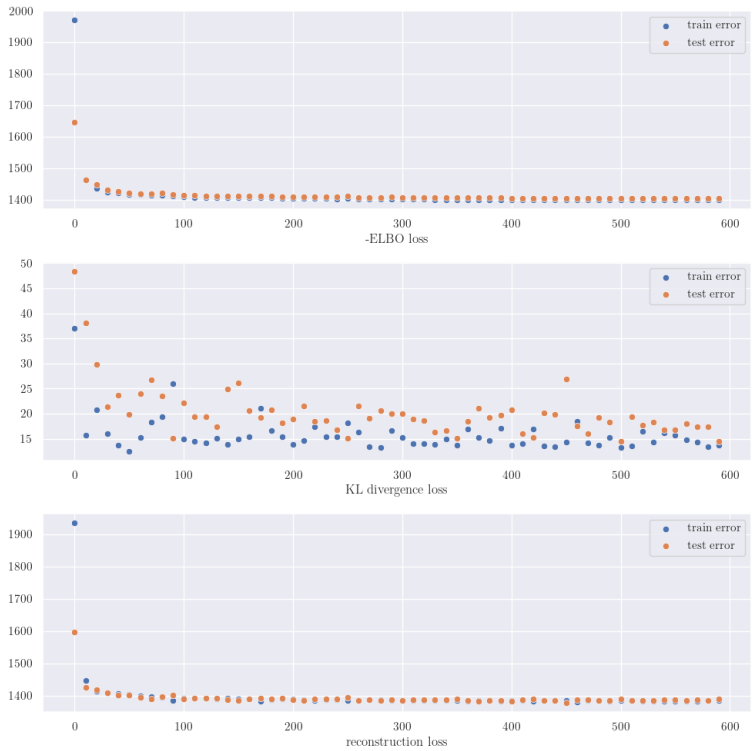



Figure 10: Learning curve for batch adjusted VAE.

We can see, that values of $-ELBO$ are much smaller then compared to normal case, we can see that semi-supervised approach is working and batch adjustment improves our results. On the figure 11 we can see PCA decomposition of our latent space.

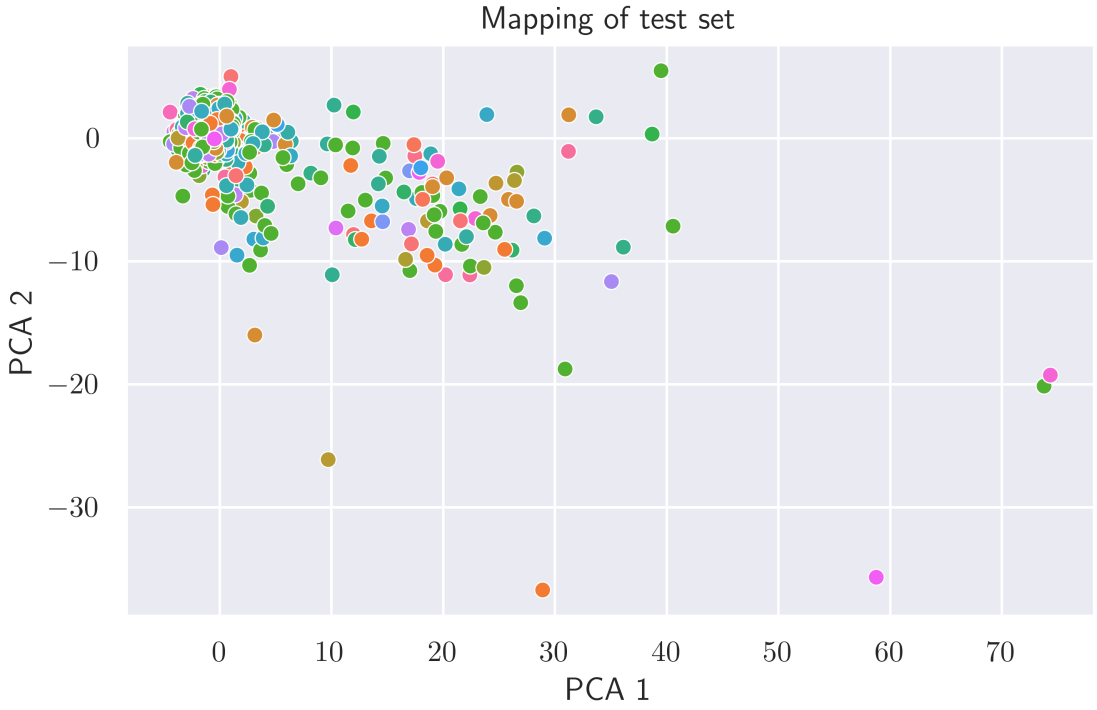



Figure 11: PCA maping of test data onto PCA for batch adjusted VAE.

# 5 Conclusions

During project three types of VAE were examiend and it was found, two of them using Negative Binomial distribution and one using Normal distribution. Each of them was trained and then evaluated, unfortunatly none of them was able to capture any type of clustering wchich can be related to biological activity. Negative Binomial distribution performed better then Normal distribution which after combination with batch information was able to yield minimal loss but still high enough making VAE unsuitable to search for any clustering.