PREDICTION OF PROTEIN ISOFORMS USING SEMI-SUPERVISED LEARNING

Álvaro Gutiérrez León s212714, Baris Kara s213449 and Casper Rasmussen s206220

Technical University of Denmark

ABSTRACT

RNA sequencing (RNA-seq) can be used to snapshot the transcriptome of cells, and ample data is being generated with this technique. However, the entire cellular expressome includes the various isoforms of proteins being produced by the cell, which RNA-seq does not elucidate. Hence, this project investigated whether protein isoform expression profiles could be predicted from gene expression profiles using a semi-supervised machine learning approach. First, a Variational Autoencoder (VAE) was trained on large amounts of high-dimensional RNA-seq data. This resulted in a model which was able to provide informative low-dimensional encodings of gene expression profiles. A smaller gene expression dataset, accompanied by a protein isoform expression dataset, was encoded with the trained VAE model. The encoded variables were then fed into a feed-forward neural network in order to predict protein isoform expression. The results showed that encoded gene expression data predicts protein isoform expression very well.

***Index Terms*—** ARCHS4, ASHA, ELBO, GTEx_annot, GTEx_gene, GTEx_iso, MSE, PCA, RNA-seq, UMAP, VAE

1. INTRODUCTION

RNA sequencing is a powerful technique that can give insights into the function of cells, since any given cell displays a specific RNA expression fingerprint depending on which tissue it is from and what state it is currently in. However, downstream regulation of cellular function such as RNA interference, alternative mRNA splicing and post-translational modifications of proteins are not captured by RNA-seq, which sets a limit for the granularity of the insight into cellular function.

Protein isoforms are variations of a protein arising from a single gene through the process of alternative mRNA splicing. Predicting isoform expression can be challenging because the process of alternative splicing is complex, but it is of interest as it can lead to change of function in the protein. In addition, isoforms of a protein can vary between different tissues, environmental conditions, stages of development and cell cycle. In recent years, the development of novel machine learning techniques, especially within the deep learning subfield, have offered new opportunities to make predictions on high-dimensional data such as gene expresson data and protein isoform expression data.

In general, processing high-dimensional data can be challenging due to several factors. One of these factors is known as the curse of dimensionality. This refers to the fact that as the number of dimensions increases, the volume of the space the data inhibits increases exponentially and the data becomes more sparse. This can make it difficult for a machine learning model to accurately learn the relationships between the features and the target variable. Additionally, the computational complexity of training and evaluating machine learning models increases with the number of dimensions, which can make it slow to process high-dimensional data and may require more powerful hardware. High-dimensional data can also be prone to overfitting, where the model performs well on the training data but poorly on unseen data due to learning relationships between features that are specific to the training data. Finally, high-dimensional data can be difficult to visualize and understand, making it challenging to interpret the results of a machine learning model.

To address these issues, various techniques for dimensionality reduction can be used to condense the data into a lower-dimensional space and capture the most important information in the data while minimizing information loss. For linearly separable datasets, Principal Component Analaysis (PCA) is one of the most commonly used dimensionality reduction approaches. However, in the case of non-linear relationships between variables, alternative approaches such as VAE may be a better option.

VAEs were introduced by Kingma and Welling [1] and are trained to reconstruct data by learning a compact latent representation of the data and then generating new data samples which should be similar to the original data, thus reducing the dimensionality with minimal information loss. In the context of RNA-seq, VAEs can be used as unsupervised models to learn important features embedded in the input expression data. This captures the underlying structure of the transcriptome in the latent space of the network, which can then be

extracted and used in tasks such as predicting protein isoform expressions in a supervised learning approach (such as regression). As such, it should be possible to capture the biological differences between cells in various tissues and various states.

This approach captures the underlying data structure in the latent space through unsupervised learning, while also making use of labeled data to make more accurate predictions. This has led to the development of semi-supervised learning models, which combine the power of unsupervised learning (such as VAEs) with the prediction capabilities of supervised learning. Overall, the combination of unsupervised and supervised learning in a semi-supervised model represents a powerful and effective approach for making predictions on protein isoform expressions from RNA-seq data.

2. MATERIALS

The materials used in this project included three datasets: *ARCHS4_gene_expression* (ARCHS4), *gtex_gene_expression* (GTEx_gene) and *gtex_isoform_expression* (GTEx_iso), as well as tissue annotations for the GTEx samples, *gtex_annot* (GTEx_annot) [2, 3]. ARCHS4 was a gene expression dataset with 168393 samples and 34558 genes, and it was used to train the VAE. The GTEx_gene dataset contained the same genes as ARCHS4 but 17382 samples which are different from the ARCHS4 samples, and it was run through the trained VAE model. The latent space generated from GTEx_gene after the VAE run was fed into a feed forward neural network, which was used to predict the values in the third dataset, GTEx_iso. This dataset contained expression data for 199325 isoforms in the same 17382 samples present in GTEx_gene. A subset of 965 of the 199325 isoforms were used for predictions. Finally, GTEx_annot provided tissue labels for the GTEx samples, and this was used to evaluate the latent space encodings and in tissue-specific regression analysis.

Sequencing data consists of counts which are normalized for gene length and sequencing depth. An investigation of the data revealed that it followed a delta log-normal distribution [4]. To normalize the expression levels and make the data easier to interpret, ARCHS4 and GTEx_gene were log2-transformed with a pseudo-count of one and then quantile-normalized in order to get all non-zero values to be normally distributed. The transformed data was modelled using a hurdle model, a two-component mixture model (Figure 1). The first component is given by a Bernoulli distribution (parameterized by p) which gives the probabilities of $x = 0$ and $x \neq 0$. The second component is a normal distribution (parameterized by μ and σ) which gives the probabilities of the non-zero values.

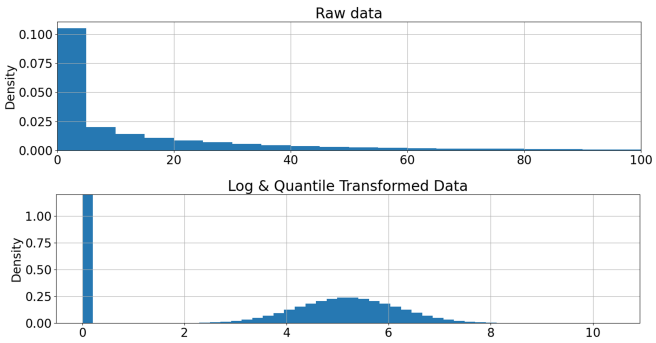


Fig. 1. Transformation of raw gene expression data into hurdle distribution.

Subsets of the datasets described above as well as jupyter notebooks containing the code used to obtain the results of this project are available at `https://github.com/sharknaro/02456-deep-learning-VAEs`.

3. METHODS

3.1. Variational Autoencoder

A VAE is a type of probabilistic generative model designed to learn a lower-dimensional representation (also known as a latent space) of the input data. It consists of an encoder neural network which learns how to map the input data to the latent space, $\mathbf{z} \sim p(z|x)$, and a decoder neural network which maps the latent space back to the original input space, $\hat{\mathbf{x}} \sim p(x|z)$. The entire model is considered under a Bayesian framework, so that:

$$p(z|x) = \frac{p(x|z) \cdot p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int_z p(x|z)p(z)dz}$$

Marginalization over z makes this intractable, and instead the posterior $p(z|x)$ is estimated by using amortized variational inference as $q_\theta(z|x)$. The encoder neural network learns the parameters θ from the data. It is common practice to choose a diagonal Gaussian distribution, $\mathcal{N}(0, I)$, as the conjugate prior $p(z)$ to ensure a regularized latent space. This enables us to express the likelihood $p(x)$ as an expectation over the estimated posterior:

$$p(x) = \mathbb{E}_{q(z|x)}\left[\frac{p(x|z)p(z)}{q(z|x)}\right]$$

It is now possible to sample from the estimated posterior, $\mathbf{z} \sim q_\theta(z|x)$, and use the decoder neural network to find the parameters ϕ of the likelihood $p_\phi(x|z)$. To evaluate the likelihood of observing x, the Evidence Lower Bound (ELBO) is used. The ELBO makes it possible to define a lower bound on the log-likelihood of the data through Jensen's Inequality, which states that for a concave function $f[\mathbb{E}[x]] \geq \mathbb{E}[f[x]]$, resulting in:

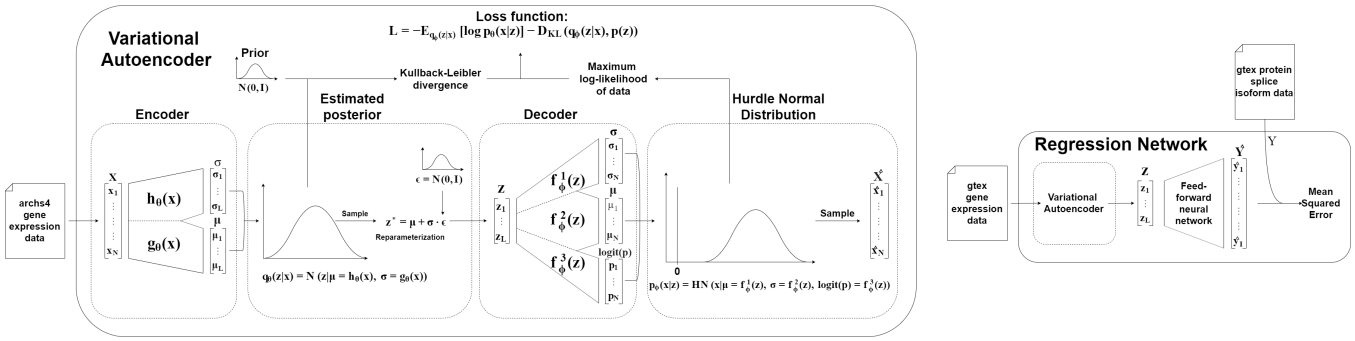


Fig. 2. Overview of the entire model. Left: The Variational Autoencoder can be understood through a Bayesian framework, $p(z \mid x) = \frac{p(x)z)p(z)}{p(x)}$, where the encoder estimates the posterior $q_\theta(z \mid x)$. A prior distribution $p(z)$ which follows a standard normal distribution $N(0, I)$ is assumed in order to regularize our latent space. The decoder parameterizes the observation model which returns the likelihood $p_\phi(x \mid z)$, that is, the likelihood of the data. Right: The regression network takes gene expression data encoded into latent dimensionality and predicts protein isoform expression through a feed-forward Neural Network.

$$log\, p(x) = log\, \mathbb{E}_{q_\theta(z|x)}\left[\frac{p_\phi(x|z)\cdot p(z)}{q_\theta(z|x)}\right] \geq \mathbb{E}_{q_\theta(z|x)} log\left[\frac{p_\phi(x|z)\cdot p(z)}{q_\theta(z|x)}\right]$$

The lower bound can be re-expressed using the Kullback-Leibler divergence:

$$\mathcal{L}(x) = \overbrace{\mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}\left[\log p_\phi(\mathbf{x}|\mathbf{z})\right]}^{\text{(a) Reconstruction Error}} - \overbrace{\mathcal{D}_{\mathrm{KL}}\left(q_\theta(\mathbf{z}|\mathbf{x}) \mid p(\mathbf{z})\right)}^{\text{(b) Regularization}}$$

Thus, the loss function presents a trade-off between reconstruction quality and latent space regularity (that is, how much the latent space diverges from a standard normal distribution).

The sampling process in the model $\mathbf{z} \sim q_\theta(z|x)$ requires the use of the reparameterization trick, as it is not possible to backpropagate through a stochastic process. Instead, the stochastic part is pushed into a separate branch of the model which samples from a standard normal distribution $\epsilon \sim \mathcal{N}(0, I)$ to calculate z as $z = \mu_\theta + \sigma_\theta \cdot \epsilon$.

For the VAE in this project, a hurdle normal distribution was chosen as the observation model. In other words, the decoder outputs the parameters for a hurdle normal distribution, which is the distribution that the data follows after transformations. Thus, it is possible to observe what the log-likelihood of observing the input x is under the predicted hurdle normal distribution. The objective is to optimize this log-likelihood while also minimizing the Kullback-Leibler divergence between the posterior and the prior.

The quality of the VAE reconstructions was evaluated in a number of ways. First, heatmaps of the input and the reconstruction (sampled from the output hurdle normal distribution) were used as a qualitative way to investigate whether patterns in gene expression were captured. Second, the parameters of the data distribution (μ, σ and p) were compared to the same parameters estimated directly from the data. Third, cosine similarity was used to see whether the 34588-dimensional input and reconstructions had the same directionality, which can be interpreted as whether the correct on/off state for gene expression was predicted, but not the expression levels in the on-state. Fourth and finally, Uniform Manifold Approximation and Projection (UMAP) was used to examine the generated latent space more directly.

3.2. Artificial Neural Network

The architecture of the regressor feed forward neural network included the input layer, two hidden layers of size 128 and 256 respectively, and an output layer. Two different networks were constructed: one with the first 64 principal components of a PCA-encoded GTEx_gene as input layer, and another with the latent space from the VAE model run on the GTEx_gene dataset. The output layer produced the final output vector giving the predicted expression levels for the protein isoforms. (Figure 2, right).

The parameters in the layers were optimized during the training process to improve the prediction accuracy of the network. As a measure of loss, the model used mean squared error (MSE), which is calculated by taking the sum of the squares of the differences between the predicted and desired output (the label isoform expression vector), and then dividing this sum by the number of samples in the dataset. The MSE loss function is useful for regression tasks, as it allows the network to optimize the parameters in a way that minimizes the difference between the predicted and desired output.

4. RESULTS

4.1. Training of the Variational Autoencoder

Grid search was used to optimize the hyperparameters and consequently the representational power of our latent space encodings. Due to the limited timeframe of the project only a cursory grid search was conducted using Ray Tune with the AsyncHyperBand (ASHA) scheduler. The primary focus was on the number of latent space variables, the number of hidden layers, and the number of neurons in those layers. Weight decay, learning rate, activation function and batch sizes could also be optimized. It appeared that increasing the complexity of both the encoder and decoder up to the GPU memory limitation led to an increased performance. For the final model, 3 hidden layers were used in both encoder and decoder with sizes 4192, 16384 and 4192. Increasing number of neurons in the first layer of the encoder or the last layer of the decoder made the memory requirements too large, as the encoder input has approximately 35000 neurons, and the decoder output approximately 105000 neurons (one set of hurdle model parameters per dimension of the data). The highest stable learning rate appeared to be 1e-4 with batch size 256. LeakyReLU with a negative slope of 0.01 was the activation function of choice. The best latent space representation obtained was 64 dimensions.

During training, the Evidence Lower Bound (ELBO), Kullback-Leibler divergence and the log likelihood of the data were monitored as shown in Figure 3 (top). The final ELBO training and validation loss were -8720 and -9271, respectively. To visualise the reconstruction of our 34558-dimensional data, a heatmap was generated of both the initial input $\mathbf{x}$ and the reconstruction $\hat{\mathbf{x}}$ as shown in Figure 3 (middle).

The ability of the VAE to capture the data distribution of the input data through the observation model was also monitored with the mean of the parameters of the observation model, as well as with its comparison to the same parameters estimated directly from the input dataset. As can be observed in Figure 3 (bottom), the model obtains a fairly good estimate of the $logit(\mathbf{p})$-parameter and the weighted mean μ of the data, but undershoots the standard deviation σ to some extent. The reconstructed data was also evaluated on its cosine similarity to the input, and it settled on a value close to 0.88.

4.2. Latent space evaluation

GTEx_gene was input to the trained VAE model and a latent 64-dimensional representation was obtained. UMAP dimensionality reduction was applied to both the original data and the VAE encodings (Figure 4). The results suggest that

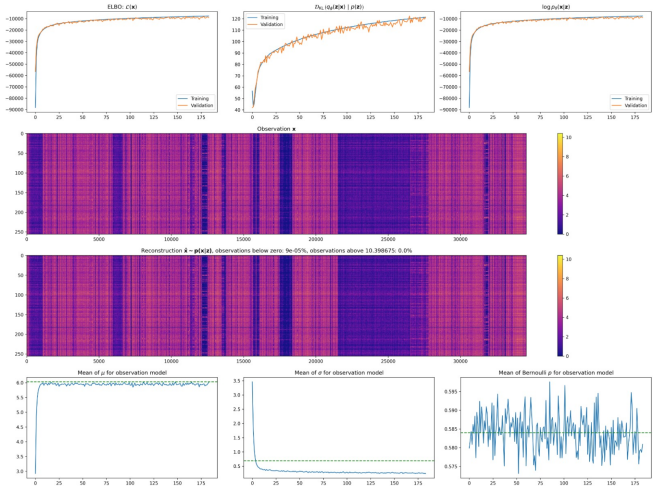
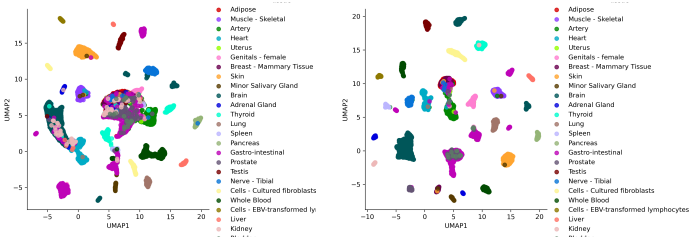


Fig. 3. Loss curves, heatmap visualization of re- construction and tracking of the mean of observation model parameters over all dimensions.



(a) GTEx_gene raw data (b) VAE encodings

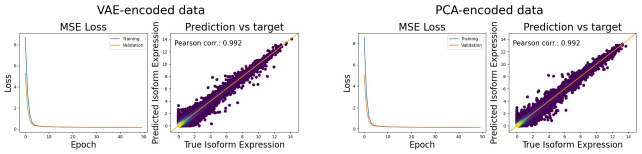
Fig. 4. UMAP dimensionality reduction.

VAE encoding achieved a slightly better separation of tissue classes. As such, VAE encodings appear to not only preserve most of the original information but also order it in a way that makes distinct clusters more separable.

4.3. Protein splice isoform prediction

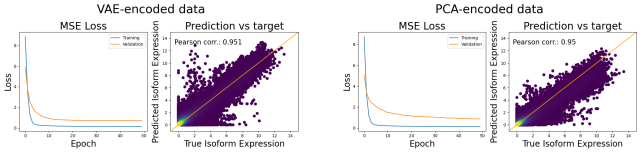
As a naïve baseline for making predictions about isoform expression based on gene expression, a Lasso regression (alpha=1) was performed, resulting in a poor fit with $R^2 = 0.091$. The gene expression encodings produced in the earlier step were at first split randomly into training and test sets, and the training set was used as input in a regressor model comprised of a simple feed forward artificial neural network. GTEx_iso was used to obtain the targets for the regressor. Additionally, as a baseline model, a PCA of the GTEx_gene expression data was conducted, and its components were also used as input to the regressor.

The test loss was measured with the MSE of the predicted isoform expression vector versus the target vector, and



(a) VAE-encoded data (b) PCA-encoded data

Fig. 5. Training curves and prediction plots when holding out 20% random samples as test set.



(a) VAE-encoded data (b) PCA-encoded data

Fig. 6. Training curves and prediction plots when holding out all brain-related tissue as test set.

it rapidly converged to values close to 0 (Figure 5, a). The datapoints of the prediction and the target were plotted and the Pearson Correlation Coefficient (PCC) was calculated, showing excellent values. The results showed a similar performance of the regressor model based on VAE encodings and the PCA-based regressor (Figure 5, b). Similarly, the PCC was evaluated on the last 16-sample batch of the VAE-encoded input data, and the values can be observed in Table 1.

The GTEx_gene dataset was also split into training and test sets based on tissue. The rationale was to hold out entire tissues as test set to see whether it was possible to predict isoform expression from encoded gene expression profiles from tissues which the regression network had never seen during training. The overall performance (as measured by the PCC) was lowered only slightly compared to randomly selected test sets, but with higher values of expression showing a larger spread in predictions, see Figure 6.

5. DISCUSSION

The results demonstrate that the VAE can indeed be used to reduce the dimensionality of gene expression data, resulting in an informative compression. The compressed data can then be used for the prediction of protein isoform expression, with a surprisingly high degree of precision as measured by the PCC. However, there are some potential concerns.

First, it appears that simply applying PCA to the gene expression data and making isoform predictions based on this yields results identical to the much more complicated and non-linear approach of using a VAE to reduce dimensionality. This brings into question whether the gene expression dataset is far less complex than first assumed, or whether the VAE could eventually surpass the PCA approach if optimized further. Figure 4 shows that the VAE manages quite good separation by tissue, but it could perhaps be improved even further. The tissue annotations provided in the materials had 54 tissue specifications for the samples, but for readability they have been condensed down to 24. Indeed, we see that e.g. all tissues of the brain cluster together in the UMAP, and so does most of the tissues for the gastro-intestinal tract. Perhaps a further optimized VAE could separate even these subsets of the tissues from one another. With a PCC above 0.99 for both the VAE and PCA based approaches already, it would however be hard to evaluate if these efforts paid off.

Table 1. Per sample Pearson correlation coefficient between real and predicted 965-dimensional isoform expressions for a batch of 16 VAE-encoded test samples.

Sample	1	2	3	4	5	6	7	8
Pearson corr.	0.993	0.989	0.991	0.992	0.987	0.991	0.985	0.994
Sample	9	10	11	12	13	14	15	16
Pearson corr.	0.990	0.992	0.993	0.990	0.981	0.991	0.990	0.992

Second, the PCC seems to be very high but by investigating the density of target vs predicted values Figure 5, the high value of this measure appears primarily driven by log2-values found between 0 and $\sim$2 where the density is very high. This could mean that the model is very good at predicting whether a gene is expressed or not, but not as good at estimating the actual value once it is expressed, particularly if it is highly expressed. It could perhaps also mean that expression values above some threshold could be considered outliers. Given more time, it would have been desirable to either handle the outliers in the data pre-processing step or perhaps try to correct for the large density at the lower values and evaluate its effect on the PCC. If it were to become apparent that the prediction strength was not as high as reported here, this would provide potential avenues for further improving the model. First of all, the ANN for regression could be optimized further. With a Pearson Correlation above 0.99, not much time was spent improving the ANN for regression. As described above, it may also be worthwhile to revisit the VAE in order to potentially improve it further and obtain more informative latent encodings of the data, which should facilitate learning for the model.

This leads to the next point, which is the assumptions made about out data during construction of the VAE model. It was observed that the overall distribution of the entire dataset follows a delta log-normal distribution. One could imagine however that each individual gene follows its own distribution: some might be more uniformly distributed,

such as housekeeping genes, whereas others may be highly tissue-specific and show drastic differences between samples. Currently, the VAE developed in this project optimizes a multivariate hurdle-normal distribution to fit to the entire dataset. However, if each gene's distribution were modeled separately, it would result in a mixture of models on a per-dimension (per-gene) basis. This would be a much more complex approach as it would necessitate an encoder which outputs different parameters (and different numbers of parameters) for each dimension of the input data.

It was observed that the standard deviation σ of the output hurdle normal model was smaller than the standard deviation in the data itself, while the parameters μ and p were very close to what could be observed in the data. The reason for the VAE model underestimating the standard deviation in the data is not known, but investigating this would be another obvious starting point for improving the model.

The distribution of the latent space is also assumed to be as close to a multivariate standard normal distribution (the prior) as possible, as the overall loss is penalized by Kullback-Leibler divergence regularization. Experimentation with other possible distributions in the latent space could perhaps also improve the amount of information captured in the latent space.

6. CONCLUSION

This project illustrated that VAEs can be used in a semi-supervised machine learning pipeline by taking a larger dataset with no labels to train on, and then using the trained model to generate informative representations of the underlying data structure in a smaller dataset. When the smaller dataset has target labels, supervised learning can be performed. Thus, the entire model learns from the structure of larger and simpler datasets to enable powerful predictions on smaller but more extensive datasets. In the present case, the encodings from the VAE did not outperform a simple PCA approach. It could be that the complexity of the data is not high enough to warrant using the VAE approach, or it could be due to erroneous assumptions about the data and latent space distributions during modeling. As the model barely overfits, it is also possible that with more time for tuning it would eventually surpass simpler approaches.

7. REFERENCES

[1] Diederik P. Kingma and Max Welling, "Auto-Encoding Variational Bayes," 2013.

[2] L. Alexander, T. Denis, and Alexandra B.K. et al., "Massive mining of publicly available RNA-seq data from human and mouse," 2018.

[3] L. John, T. Jeffrey, and Mike S. et al., "The Genotype-Tissue Expression (GTEx) project," 2013.

[4] John Aitchison, "On the Distribution of a Positive Random Variable Having a Discrete Probability Mass at the Origin," 1955.