UNRAVELING THE ISOFORM WORLD: A VAE APPROACH WITH A FFNN (PROJECT 29)

Ana Pastor Mediavilla (s222761), Carlos de Santiago León (s222766),
Anu Oswal (s222498), Laura Figueiredo Tor (s222797)

Technical University of Denmark (DTU)

ABSTRACT

Due to alternative splicing, a single gene can be transcribed into different mRNAs which will be translated into different protein isoforms with distinct functions. This project introduces a semi-supervised model making use of a Variational Autoencoder (VAE) to map high-dimensional gene expression data from 54 human tissues into an informative latent space, which will be used as the features for a Feed Forward Neural Network (FFNN) to predict isoform expression. The FFNN performance using VAE-encoded data as input significantly outperformed both a baseline regression model and a FFNN which directly took gene-expression data (non-encoded) as input. Hence, we demonstrated that VAEs provide an effective approach for feature reduction in predicting isoform expression. Integrating VAEs and FFNNs offers a valuable framework for understanding complex relationships in high-dimensional gene-expression data. Further exploration could involve generalization testing on independent datasets, deeper VAE training for improved performance, and hyperparameter tuning.

The developed code is accessible on GitHub: `https://github.com/CarlosSanti00/deep_learning_project.git`

1. INTRODUCTION

1.1. The importance of addressing isoforms' expression

Alternative splicing is a cellular process in which exons from the same gene are joined in different combinations. As a result, one gene can produce different mRNA transcripts, which can be translated to different proteins with diverse structures and functions[1]. It has been suggested that alternative mRNA splicing is a major source of cellular protein diversity[1].

The different proteins that can arise from the same gene are called ”isoforms”. Analysis of isoform expression is essential, as isoforms play key roles in biological processes and can be used as therapeutic targets or biomarkers. For instance, some studies have reported associations between isoforms and drug response or resistance[2].

Despite this, the extent and biological relevance of splicing are currently unknown[3], and the majority of studies that analyze gene expression disregard them[4]. Thus, predicting isoform expression from transcriptomics data using Deep Learning models would allow researchers to understand the impact of these isoforms without performing lab-analyses, shedding light on the effects of genetic variants and protein functions on specific tissues.

In this study, a latent-feature discriminative model using a Variational Autoencoder (VAE) was built to map the input data to a reduced but informative latent representation, therefore capturing the key characteristics. Such a model would provide a low-dimensional representation of the data, which is of special interest in high-dimensionality datasets, as is the case of transcriptomics datasets[5].

More specifically, our purpose was to build a VAE capable of capturing the most important information of gene-expression human data from 54 different tissues, in a latent space. Afterwards, the VAE-encoded data was used as input for a Feed Forward Neural Network (FFNN), which expanded the data into a higher dimensionality to predict the levels of the isoforms corresponding to the genes in the original data.

1.2. Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) are a class of generative models designed to learn a probabilistic mapping between high-dimensional input data and a lower-dimensional latent space. Unlike traditional autoencoders, VAEs introduce a probabilistic interpretation, treating the latent variables as drawn from probability distributions[5].

As a regular autoencoder, a VAE is composed of an encoder and a decoder that are trained to minimize the reconstruction error between the encoded-decoded data and the initial input data. However, to become a generative model, it introduces a regularization of the latent space. This way, it is possible to randomly sample a point from the latent space and decode it to generate new content. To do so, the VAE encodes the input as a distribution over the latent space[6].

Thus, when building a VAE it is important to consider the following:

- **Posterior Distribution ($q_\phi(z|x)$):** Represents the distribution of the latent variables (z) given the observed data (x). The encoder maps the input data to the parameters of the posterior distribution.
- **Prior distribution ($p(z)$):** Represents the assumed distribution of the latent variables in the absence of any specific observed data.
- **Observation model ($p_\theta(x|z)$):** Represents the distribution of the observed data (x) given a sample from the latent space (z). The decoder maps a latent variable into the parameters of the distribution of the observed data[5, 6].

VAEs use backpropagation for training and optimizing the model parameters to maximize the Evidence Lower Bound (ELBO). The ELBO consists of two terms: the **reconstruction term**, which represents the log-likelihood of the observed data given the latent variable, and the **Kullback-Leibler (KL) divergence** term, which penalizes deviations of the learned latent distribution from the specified prior distribution. β is a hyperparameter that controls the trade-off between the likelihood and the KL divergence.

$$\text{ELBO} = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta\mathcal{D}_{\text{KL}}[q_\phi(z|x)||p_\theta(z)]$$

1.3. Objectives

The aim of this project is to develop a predictive machine learning algorithm capable of learning the relationships from gene-expression data from 54 human tissues to predict the level of expression from a huge repository of different isoforms (proteins) encoded from the studied gene pool. These isoforms are defined as continuous numerical float values, which makes our machine learning algorithm, which is a Feed-Forward Neural Network, work as a regression model that predicts the expression of 156958 isoforms.

For that implementation, a prior dimensionality reduction is developed for capturing the essential gene-expression correlations in a lower feature space. This is achieved by developing a Variational AutoEncoder (VAE). This approach of combining an unsupervised methodology for feature reduction with a supervised algorithm is known as semi-supervised learning. Moreover, the approach of combining an autoencoder approach with a machine learning predictor is known as the latent-feature discriminative model (M1) approach[6].

2. DATA

Gene-expression data from human samples across 54 different tissue types was used to carry out this project. The data is log-transformed and normalized in terms of $log_2(TPM + 1)$, meaning that the data was normalized into Transcripts Per Million (TPM) and then log-transformed. The '+1' is to deal with 0 values when log-transforming the data.

The data was contained in the following files:

- **archs4 gene-expression dataset [167883, 18965]**: Big unlabeled gene-expression data set, containing 167883 samples (rows) and 18965 genes (columns). In this context, "unlabeled" means that there is information for the levels of expression of each gene, but not for the isoforms corresponding to the genes. This file was used for training the VAE, to encode the gene expression features into a latent space.
- **gtex gene-expression dataset [17355, 18965]:** Small labeled paired gene-expression data set, containing 17355 samples (rows) and 18965 genes (columns). The file structure was the same as in the archs4 gene-expression dataset, but in this case, there was information about the expression levels for each isoform of each gene in another file. In a predictive model, this file would be the X (independent variables).
- **gtex isoform-expression dataset [17355, 156958]:** Small paired data containing the isoform expression for the genes and samples found in the gtex gene-expression dataset. The number of samples (rows) is the same as in this dataset, 17355, but the dimensions have been expanded to 156958, as each gene has been split into different isoforms. In a predictive model, this file would be the y (target variable(s)).
- **gtex gene isoform annotation file:** Gene-isoform relationship for the gtex dataset. Each isoform is assigned to a gene id.
- **gtex tissue annotation:** Human tissue types for each sample. In total, there were 54 tissues.

The described files are available in the high-performance computing server (https://www.hpc.dtu.dk/) in the following directory: */dtu-compute/datasets/iso_02456*. The files are compressed because of their huge size. Due to this, HDF5 files were created to facilitate the accession to the data when programming the deep learning algorithms. These files are available in this path: */dtu-compute/datasets/iso_02456/hdf5/*.

3. METHODS

With the aim of building a model that can predict the expression levels of isoforms of selected genes, the following workflow was followed:

1. Training of a VAE using the **archs4 gene-expression** dataset to reduce dimensionality and map the input data to a lowered latent space, while capturing the essential information.
2. Application of an FFNN, as a regression model, to predict isoform levels (156958 continuous output values) from the latent features of the VAE. Since the aim was predicting the levels of isoform expression, the FFNN was trained using the **gtex gene expression and isoform-expression** files.

All of the methods described were developed in Python, and run in the high-performance computing (HPC) system available for DTU students (`https://www.hpc.dtu.dk/`)

3.1. VAE

3.1.1. Architecture

As mentioned above, a VAE was trained using the archs4 gene-expression unlabelled dataset to find an appropriate way of reducing the dimensionality of the data while capturing the most important information.

In this case, the VAE implemented consisted of a 5-layer encoder that mapped the data into a 256-dimensions latent space, and a 5-layer decoder that mimicked the encoder's architecture but in the opposite direction (hour-glass shape), reconstructing the data to the original dimensions (Figure 1).

Specifically, the input data, with 18965 features, was forwarded to the first layer, which applied a linear transformation to map it to 4096 features and applied the ReLu activation function. This same procedure was repeated in the following 4 fully connected layers, each of them reducing the dimensionality to 2048, 1024, 512, and, finally, 256 features in the latent space by applying linear transformations and ReLu activations. Afterwards, the decoder used 5 fully connected layers to map the data in the latent space to 512, 1024, 2048, 4096 and, finally, 18965 features, therefore performing a reconstruction of the data. In the decoder, the 4 first layers applied both a linear transformation and ReLu activation, while the last one performed only a linear transformation. The VAE's parameters were upgraded using the Adam optimizer, and the learning rate was set to 10^{-4}. β was given a value of 1, to strike a balance between faithful reconstruction and a well-structured latent space.

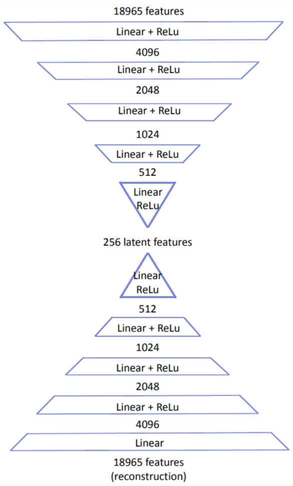


Fig. 1. Architecture of the VAE. The encoder maps the input data to the latent space (256 dimensions), and the decoder reconstructs the data and restores the original dimensions.

3.1.2. Parameters

Gaussian distribution was assumed for the prior and the posterior. The distribution used for the observational model was log-normal with unit variance. The Log-normal distribution was chosen to represent the observed data given a sample of the latent space, as the log-transformed expression data consisted of only positive values.

3.1.3. Training and validation

Given the enormous size of the archs4 gene-expression dataset (167883 samples), the VAE was trained using batches of size 64, and a total of 100 epochs were performed due to the computational limitations when working on the HPC. The batches were shuffled in each epoch.

In each iteration inside an epoch, validation was performed by calculating the ELBO and the loss (negative ELBO), and the parameters were updated. The ELBO for all batches in one epoch was averaged to calculate the value of the ELBO for that epoch.

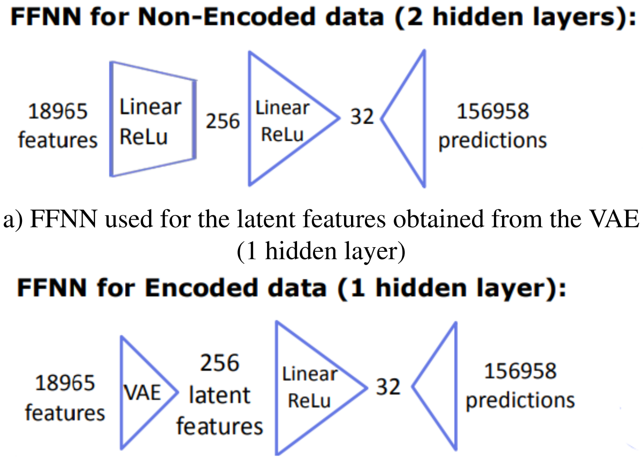
3.2. Standard AutoEncoder

As mentioned above, the VAE is developed for feature reduction that will be introduced in an FFNN for isoform-level prediction. However, as our objective was not to generate new data, but to capture an informative representation of the input data in a reduced dimensionality, we considered the possibility that a VAE might introduce unnecessary complexity. Thus, we opted to transform the VAE into a regular autoencoder and evaluate if the model's performance improved.

3.3. Feed-Forward Neural Network (FFNN) as a regression model

The purpose of building and optimizing a VAE was to find a good way of encoding the data into a latent dimension with reduced features without losing important information. Once this was done, we implemented a neural network on the encoded data to end up in a 156958-units output layer, with the aim of predicting the expression levels for each isoform.

However, it is crucial to assess if the implementation of VAE for feature reduction truly boosts the further performance of a neural network. To assess this, we additionally implemented another neural network architecture with an additional hidden layer that reduces the original feature dimension of our dataset to 256 units, simulating the dimensions obtained from the encoder of the VAE. The basic architectures for both FFNNs are shown in Figure 2.


FFNN for Non-Encoded data (2 hidden layers):

a) FFNN used for the latent features obtained from the VAE (1 hidden layer)

FFNN for Encoded data (1 hidden layer):

b) FFNN used for the original feature-dimension dataset (2 hidden layers)

Fig. 2. FFNN designed architectures for predicting isoform expression levels.

To define if our model's performance is significant or behaves randomly, a baseline regression model was defined by making the predictions on a validation set according to the true observed mean isoform-expression levels value in the training set. Therefore, if the loss values from the neural networks are close to this baseline, we can conclude that our model is not performing accurately.

Since the aim of the project is to predict the expression levels of isoforms (156958 features), the data used in this case was from the ”gtex” files. This labeled dataset includes information from the tissue origin of the samples. To train a model that learns the correlations and specificities from the different tissues sampled in our dataset, a stratification procedure was implemented to divide the data into 5 folds to implement cross-validation to successfully measure the performance of the models created.

3.3.1. FFNN for encoded data

After trying multiple different architectures, we concluded that the best results were obtained using a very simple neural network, with only 1 hidden layer. Thus, after encoding 18965 features to 256 using the VAE, the hidden layer applied a linear transformation, mapped the data to 32 features, and applied ReLu activation. After this transformation, the output layer expanded the dimensionality to 18965 features, following the total number of isoform levels to predict, through a linear transformation.

During the training, stratified 5-fold cross-validation was implemented. By applying stratification, we made sure that each fold had a similar distribution of tissues as the entire dataset, thus ensuring that each fold was representative of the overall tissue distribution. In each fold, the training set consisted of 13885 samples and the validation set of 3471 samples. Due to the limited computing resources of the queuing system on the HPC server, each fold was trained for 100 epochs. Furthermore, given the size of the dataset, the data in each iteration was presented into the network by batches of size 64. The Adam optimizer was used for gradient-descent optimization of the network after each iteration, and the learning rate was set to 10^{-4}

For validation, in each fold, the average Mean Squared Error (MSE) for each epoch was computed after calculating the value for each batch. This same procedure was applied with the data encoded by the standard autoencoder, rather than the VAE.

3.3.2. FFNN for non-encoded data

As mentioned above, a regular neural network was built for using the complete dimensionality of the original ”gtex” dataset. Rather than the data encoded by the VAE, it took as input the original gtex expression data, with 18965 features. This FFNN was designed with 2 hidden layers: the first one, in an analog function to the VAE's encoder, reduced the dimensionality to 256 features by applying a linear combination and a ReLu activation. The second one also used a linear transformation and ReLu activation to achieve a space of 32 dimensions, which were expanded to the final number of features, 156958, by the output layer. Regarding training and validation, the same procedure that was applied to the FFNN for encoded data was used in this case.

4. RESULTS

4.1. VAE

After training the VAE for 100 epochs, we achieved a model in which the ELBO increases across epochs (Figure 3a). Even if this is much clearly appreciated in the training data, an increasing tendency for the ELBO can also be seen in the validation data. It was also proven that the VAE is capable of finding a good representation of the data in the latent space.

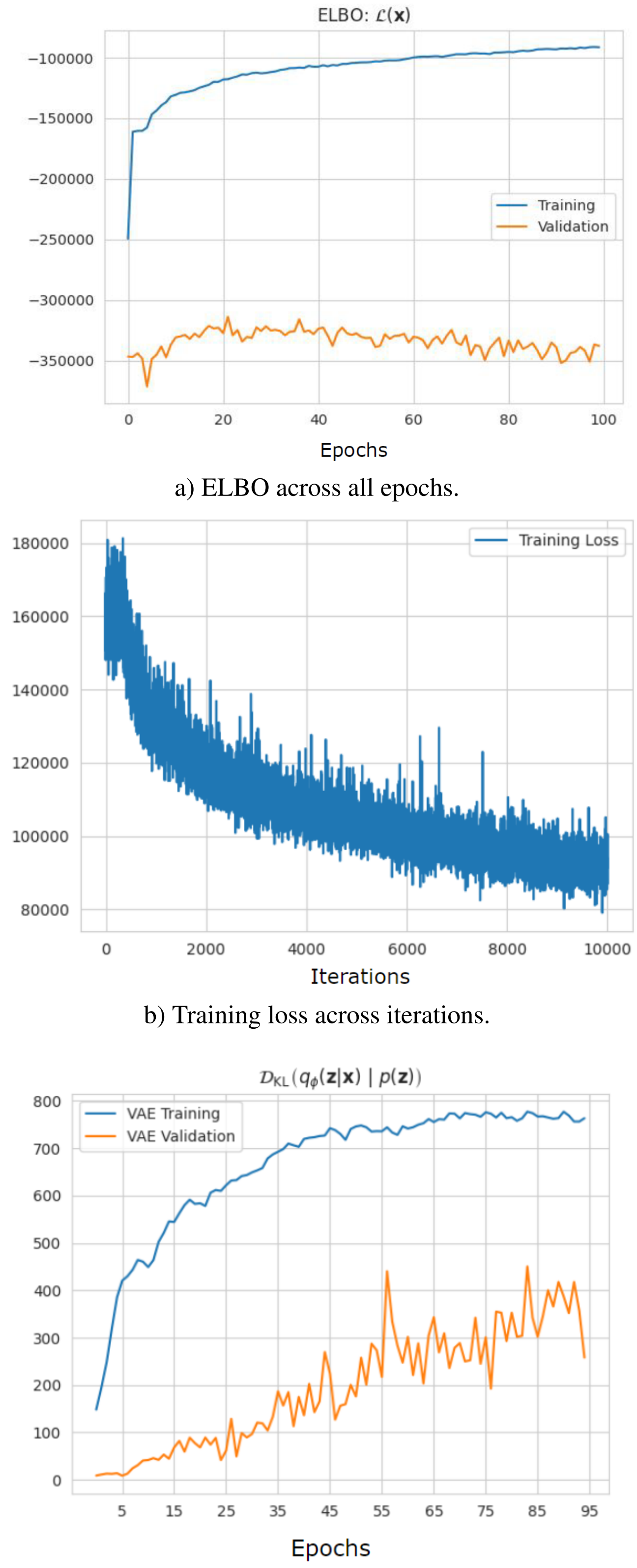
During the training process, the loss was calculated as the negative ELBO, and minimized. By plotting the training loss across all iterations for all the batches, it is possible to appreciate how it is minimized along the training process (Figure 3b). The loss reduction correlates with the ELBO improvement across the training iterations.

Finally, we saw an increase in the KL-divergence (Figure 3c), suggesting that the model refines its latent space representation according to the regularization of the KL once the first training epochs have concluded. The KL term acts as a regularizer, preventing the latent space from becoming overly complex and helping to ensure a smooth and well-behaved latent distribution. The slow increase in the KL term indicates that the model is adjusting the latent space to adhere to the desired distribution, balancing the trade-off between a faithful reconstruction of input data and regularization of the latent space.

Overall, we can conclude that our VAE implementation is performing accurately by observing a minimization of the ELBO and losses across the training epochs while observing a rise in the KL regularization term. This suggests that the VAE is fulfilling the task of creating a significant latent space, recording meaningful characteristics of the gene-expression dataset.

4.2. Standard AutoEncoder

After replicating the training of the VAE but setting β to 0, the results did not improve. As shown in Figure 4 , the new ELBO (which consisted only on the reconstruction term) barely changed. This may be because the values for the KL



a) ELBO across all epochs.

b) Training loss across iterations.

c) KL-divergence across epochs.

Fig. 3. Results of the VAE implementation. ELBO (a), KL-divergence (b), and loss (c) results were included. The validation set consisted of random selections from the gtex gene-expression dataset.

divergence term calculated in the VAE had a much lower magnitude compared to the reconstruction term (Figure 3), and, therefore, removing them did not result in an impactful change.

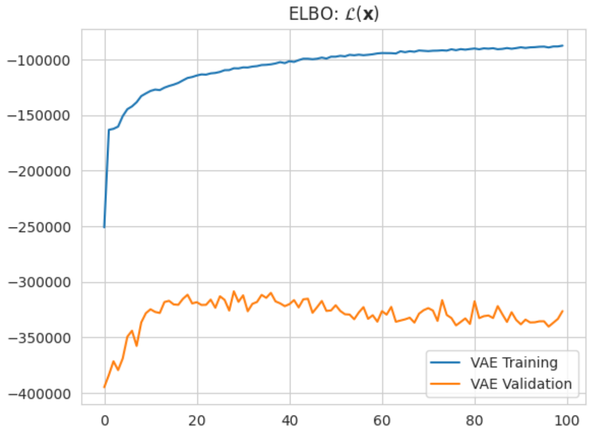


Fig. 4. ELBO results across all epochs for the standard autoencoder ($\beta = 0$).

4.3. FFNN

Table 1 presents the MSE for each cross-validation fold, comparing the performance of the FFNN with 1 hidden layer that used the VAE-econced data, the FFNN with 2 hidden layers that used the original data without feature reduction (non-encoded data), and a baseline that consisted on assigning the means across the training labeled samples. As can be observed, the FFNN that used the VAE-encoded data outperformed the other two models in all folds.

By computing the mean across all folds, it is possible to confirm that the highest error was obtained when using the baseline (0.48529), followed by the FFNN that did not benefit from the encoding of the VAE (0.476632). The 1-hidden layer FFNN outperformed its competitors, with a mean MSE across folds of 0.32126.

However, before assuring with rigor that one model performs better than another, it is necessary to conduct a statistical test. Therefore, we performed paired t-tests to compare the three selected models: FFNN with 1 hidden layer and VAE-encoded data as input, FFNN with 2 hidden layers and non-encoded data as input, and the baseline. The corresponding p-values can be observed in Table 1. When considering a significance of $\alpha = 0.05$, we can confidently assure that the combination of VAE and FFNN outperforms both the baseline (p-value = 4.446×10^{-47}) and the FFNN which does not use the VAE-encoded data p-value = (9.755×10^{-41}). Furthermore, the importance of incorporating the VAE into the model is highlighted when observing that the Non-VAE FFNN does not significantly outperform the baseline (p-value = 0.0532).

As a last approach, the FFNN that used the VAE-encoded data was replicated, but this time taking as input data encoded by the standard autoencoder. After 100 epochs and 5 cross-validation folds, the performance was worse than the VAE FFNN in all folds. Therefore, we opted to not include those results and discard this alternative by not pursuing further exploration of this option.

5. DISCUSSION AND CONCLUSIONS

We can conclude that VAEs can be useful tools for semi-supervised approaches. The VAE developed in this project proved to capture the most important information of high-dimensional gene-expression data into latent features that, when used as input for an FFNN for isoform-level predictions, outperformed both a baseline and an FFNN that did not benefit from VAE-encoded data as input. For the prediction of isoforms, we can state that performing feature reduction is efficient and advisable. When using the complete expression levels dimensionality, the model performs as a random model when comparing it to the performance of a baseline.

Moreover, a VAE performed a better feature reduction than an autoencoder which did not penalize divergence from a pre-defined distribution. We found this surprising, as we had hypothesized that reducing this constriction would allow the model to capture better our particular expression data. Regular autoencoders or other feature reduction algorithms like PCA could be trained and tested in further exploration to get a more informed and valid conclusion regarding this topic.

Furthermore, while exploring different architectures for FFNNs, we found it difficult to build networks that performed better than the baseline. This may be because the majority of isoforms have rather low expression. This is also probably why very simple neural networks perform better than more complex ones, and why the baseline got a low MSE error. In addition, it would be interesting to analyze how the network would generalize to unseen tissues by testing it on an independent test set. This could be implemented with our dataset while keeping the stratification structure.

It is important to note that, due to the limited computational power and the massive size of the dataset, it was not possible to train the VAE on all the data. Most probably, the VAE's performance could improved if training with all the data points was possible. Finally, a deeper analysis for refining the network hyperparameters should be carried out to achieve optimal performance.

Table 1. Performance results for the VAE-encoded data FFNN, for the non-encoded data FFNN, and the baseline for each cross-validation fold. Mean-Squared Errors (MSE) are included for each fold and network. Paired t-test analysis were run to measure statistical significance. The p-values results were included for each fold. The mean across all folds for all of the explained metrics were included as the last row of the table.

Cross-validation Fold	**MSE VAE Encoded Data**	**MSE Non Encoded Data**	**MSE baseline**	**VAE vs Non-VAE p-value**	**VAE vs baseline p-value**	**Non-VAE vs baseline p-value**
1	0.31993	0.47646	0.48407	8.375×10^{-46}	1.015×10^{-50}	0.042
2	0.31285	0.47814	0.48469	5.167×10^{-44}	1.841×10^{-59}	0.113
3	0.32756	0.47812	0.48801	4.132×10^{-40}	1.857×10^{-57}	0.027
4	0.32657	0.47506	0.48632	1.002×10^{-45}	2.223×10^{-46}	0.006
5	0.31940	0.47538	0.48334	7.449×10^{-41}	2.040×10^{-52}	0.078
Mean	0.32126	0.476632	0.48529	9.755×10^{-41}	4.446×10^{-47}	0.0532

6. REFERENCES

[1] National Human Genome Research Institute, “Alternative splicing,” 2023, Accessed: December 10, 2023.

[2] Zhaleh Safikhani, Petr Smirnov, and Kelsie L. et al Thu, “Gene isoforms as expression-based biomarkers predictive of drug response in vitro,” *Nature Communications*, vol. 8, pp. 1126, 2017.

[3] Søren H. Dam, Lars R. Olsen, and Kristoffer Vitting-Seerup, “Expression and splicing mediate distinct biological signals,” *BMC Biology*, vol. 21, no. 1, pp. 220, 2023.

[4] M Stastna and JE Van Eyk, “Analysis of protein isoforms: can we do it better?,” *Proteomics*, vol. 12, no. 19-20, pp. 2937–2948, 2012.

[5] Carl Doersch, “Tutorial on variational autoencoders,” 2021.

[6] Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling, “Semi-supervised learning with deep generative models,” 2014.