Summary

The report presents the application of a variational autoencoder (VAE) tested on synthetic datasets. This β-VAE model was developed to uncover mutational signatures, estimate exposures, and provide confidence intervals, offering a probabilistic approach to understanding mutation patterns.

Mutational signatures represent characteristic mutation patterns within a genome, often indicative of underlying processes such as exposure to environmental factors or biological mechanisms like DNA repair defects. Accurate extraction and estimation of these signatures and exposures are crucial for tailoring specific cancer treatments.

Traditional methods for extracting mutational signatures typically use Non-Negative Matrix Factorization (NMF). NMF is favored due to its ability to decompose the mutational frequency matrix into non-negative components, representing signatures and exposures. State-of-the-art methods have been successful but lack the ability to provide confidence intervals, which can limit the understanding of the certainty of the result.

The β-VAE developed in this study consists of an encoder, a probabilistic latent space, and a decoder. The encoder maps input data into a lower-dimensional latent space, which captures the exposures of mutational signatures. The decoder reconstructs the input data from this latent representation. The unique aspect of this model is its ability to estimate confidence intervals by analyzing the means and variances in the posterior distribution. This allows the model to not only capture the mutational signatures (in the decoder weights) but also estimate the exposures (in the latent space) with a measure of certainty.

The model was tested on several synthetic datasets, designed to simulate various cancer types and mutational scenarios. The experiments revealed that the β-VAE performs competitively when β is set to lower values, achieving high cosine similarity with ground truth signatures. However, compared to established methods, the VAE showed slightly lower precision in signature extraction. The introduction of confidence intervals, however, added a valuable dimension to the analysis, enhancing the interpretability of the results.

However, the current performance suggests that additional refinement and exploration of alternative probabilistic models and mixture distributions are necessary to achieve competitive accuracy with state-of-the-art tools. The report suggests integrating these models with the VAE framework to enhance its predictive capabilities and provide a more comprehensive tool for mutational signature analysis.

The study concludes that while the β-VAE shows promise in extracting mutational signatures and estimating exposures, it requires further refinement to match the precision of current state-of-the-art methods. Future research should explore alternative probabilistic models and mixture distributions to improve prediction accuracy. The unique contribution of confidence intervals remains a significant advancement, providing deeper insights into the certainty of the extracted mutational signatures and exposures.

Describing mutational signatures using variational autoencoders

Casper Gislum CGISLU19@STUDENT.AAU.DK
Department of Computer Science, Software
Aalborg University

Mathias Vestergaard Jensen MATJEN19@STUDENT.AAU.DK
Department of Computer Science, Software
Aalborg University

Abstract

This report investigates the use of variational autoencoders for identifying mutational signatures within cancer genomics data. Mutational signatures represent characteristic patterns of mutations that can indicate underlying mutational processes, such as exposure to environmental factors or defects in DNA repair mechanisms. Traditional methods for extracting these signatures often employ Non-Negative Matrix Factorization (NMF). However, recent research explores the potential of autoencoders as a viable option within this field. This paper developed a β-VAE to find exposures, mutational signatures, and confidence intervals. The contribution of confidence intervals is unique to this paper and is derived by analyzing the probabilistic latent space. While experimental results demonstrate that theβ-VAE can achieve competitive performance, it lags behind state-of-the-art methods in terms of signature extraction. The findings highlight the need for further refinement and suggest future directions, including the exploration of alternative probabilistic models to enhance prediction accuracy.

Keywords: Variational autoencoder, Mutational signatures, Confidence intervals

1 Introduction

A characteristic pattern of mutations is called a mutational signature. A mutational signature reflects the underlying mutational processes that have taken place within a genome. These mutational signatures are typically related to environmental factors such as exposure to smoking and sunlight. Otherwise, they can be associated with internal biological processes such as defective DNA repair. Identifying the mutational signatures and their exposures can assist in specializing treatment [1][2][3]. Moreover, there exist multiple categories of mutations such as base substitutions (single and doublet), insertion, and deletion. In this project, we will exclusively focus on single base substitution (SBS) mutations. SBS mutations occur when a certain nucleotide base is replaced. There are 6 different possible substitutions (when considering strand symmetry): C > A, C > G, C > T, T > A, T > C and T > G [4]. If one includes the neighboring bases of the substituted base, which can be: A, C, T, and G it results in 96 combinations.

For some time researchers have been trying to uncover these mutational signatures, using techniques suited for this category of problem. This means the category of "cocktail party" problems, which entails separating the individual sources from a mixture of sources. The individual sources can be likened to mutational signatures in this instance and a mutational

frequency matrix to a mixture. The most commonly used method within this field is NMF (Non-Negative Matrix Factorization), which state-of-the-art methods described by Alexandrov et al. [5], Alexandrov et al. [6] Islam et al. [7] utilize. NMF is especially suited for this problem as it requires a non-negative constraint to model mutational signatures and exposures. However, recently researchers have studied whether autoencoders (AEs) can uncover mutational signatures, namely, Pancotti et al. [8] and Pei et al. [9]. AEs can put a non-negative weight constraint on the weights and use linear activation functions such as ReLU needed to model mutational signatures and exposures. The results of [8] indicate that AEs serve as a competitive method, in a scenario where the true signatures are known.

An aspect AEs and NMF do not consider when producing the mutational signatures and exposures for each signature is the distribution of the original data. This can be a problem as the certainty of the results is unaccounted for. It only concerns itself with producing similar results in terms of comparing the input to the output, i.e., minimizing the reconstruction loss. A branch of AEs that has not yet been explored within this field (to our knowledge) is the variational autoencoder (VAE). VAEs consider the distribution of the original data by incorporating the KL divergence into their loss function alongside the reconstruction loss.

This project focuses on developing a VAE based on the AE defined in Pancotti et al. [8]. For this approach, a β-VAE whose latent space follows a folded normal distribution has been developed, alongside the alterations to facilitate utilization of the additional data, specifically, the means and variances. The means and variances of the posterior distribution are used to approximate the exposures and confidence intervals. The structure of the β-VAE closely follows the one described in Pancotti et al. [8], meaning the signature extraction and refitting are handled nearly identically, but with significant change and addition in the production of exposures and confidence intervals. A theoretical use-case of such a model would be to evaluate the certainty of a predicted exposure of a mutational signature within a genome. Doctors would be able to use the confidence intervals to ascertain with what certainty a patient had the predicted exposure of any given mutational signature.

In this paper, Pancotti et al. [8] was expanded by developing a β-VAE, in addition to the capability to estimate confidence intervals by sampling the means and variances of the posterior distribution. The model can successfully produce signatures, exposures, and confidence intervals. When compared to 11 other signature extraction methods, it ranks in the middle, proving it is competitive, yet falls behind start-of-the-art methods such as SigProfilerExtractor [7] and MUSE-XAE [8]. Furthermore, the experiments found that using a β of 0 produced the most competitive results, which renders the loss function equivalent to the autoencoder [8]. Showing the β-VAE is not using the capabilities of a VAE to its utmost potential.

2 Related Work

Many methods have been developed to uncover mutational signatures and exposures from cancer genomics data obtained through cancer genome sequencing, with most employing variations of NMF [7]. In this chapter, a few of these methods are summarized with a focus on autoencoders, as this paper aims to develop a variational autoencoder, which is uncommon in this area.

Pei et al. [9] uses a denoising sparse autoencoder, whose encoder is a linear layer followed by a ReLU activation function. The decoder is a linear layer followed by a softmax function. It is a denoising sparse autoencoder as it adds a noise matrix to the input matrix. It trains over the mutation types and reduces the dimension of N cancer samples to 200. Meaning it trains over the features rather than the samples. This structure means the mutational signatures are captured within the latent space z and the exposures in the encoder weight matrix W.

Alexandrov et al. [5] described a framework utilizing NMF to extract signatures from the dataset. This framework was revised in Islam et al. [7] and named SigProfilerExtractor. SigProfilerExtractor decomposes the input matrix M and searches for the number of active signatures k. In each decomposition, SigProfilerExtractor performs 100 independent factorizations, and for each repetition, M is Poisson resampled and normalized. Subsequently, M is factorized with a multiplicative update NMF algorithm, minimizing the objective function based on the Kullback-Leibler divergence. Lastly, a custom partition clustering, which uses the Hungarian algorithm to compare repetitions, is applied to the 100 factorizations to find stable solutions. SigProfilerExtractor selects the centroid of stable clusters as the optimal solution. This process stabilizes the solutions as NMF lacks uniqueness with multiple convergent stationary points [7].

Pancotti et al. [8] proposes one of the few existing autoencoders with promising results. The autoencoder has a hybrid architecture where the encoder is nonlinear and the decoder is linear. The benefit of this is the autoencoder will be able to identify nonlinear patterns while still ensuring the interpretability of the data. The decoder also has a non-negative weight constraint, as this is where the mutational signatures are captured. The exposures are modeled in the latent space. The encoder has 3 linear layers with a batch normalization and softplus activation function after each linear layer. The decoder has a singular linear layer without any activation function. Compared to state-of-the-art mutational signature extraction techniques, such as SigProfilerExtractor, it performs favorably. It scores a mean AUC F1-score of 0.92 with a standard deviation of 0.05, while SigProfilerExtractor scores a mean of 0.90 with a standard deviation of 0.06 [8].

These methods have proven to be reliable in extracting signatures and exposures, but are unable to provide confidence intervals. The contribution of this paper is to develop a variational autoencoder and use the means and variances in the posterior distribution learned by the model to give this estimation. Additionally, the capabilities of the variational autoencoder will be evaluated, both in terms of extracting signatures and estimating exposures.

3 Background

In this section, we provide a detailed description of the problem and the model employed to address it.

3.1 Variational autoencoders

The variational autoencoder [10] consists of an encoder, a multivariate Gaussian distribution in the latent layer, and a decoder. With this regularized latent space, it allows for the ability to generate new data from the latent space. The objective of the encoder is to map the input

into a lower-dimensional latent space, with the objective of the decoder being to reproduce the input. Its structure can be seen in Figure 1. It is possible to make it conform to different distributions, however, modeling it after a multivariate Gaussian distribution is standard practice. To sample the latent, it uses the mean and log-variance learned by the encoder. It uses the log-variance since it can have both positive and negative values, which is beneficial for the learning process. Whereas, the variance can only have positive values. Secondly, it uses an epsilon which is drawn from a standard Gaussian distribution i.e. $\mathcal{N}(0, 1)$. Thereby, a sample is given by $z = \mu + \sigma \cdot \epsilon$. The loss function consists of two terms, one term is responsible for minimizing the reconstruction loss (RL) i.e. creating an output similar to the input, typically measured using Mean Squared Error (MSE). The second term is ensuring the latent space is continuous and conforms to a standard Gaussian distribution, typically measured using KL divergence. There exist multiple types of VAEs, one of which is the β-VAE [11]. The β-VAE adds a hyperparameter in front of the KL divergence that can be seen in Equation 1, this can enable the VAE to learn complex patterns in the data by adding more flexibility in the latent distribution.

$$Loss = RL + \beta \cdot KL \tag{1}$$

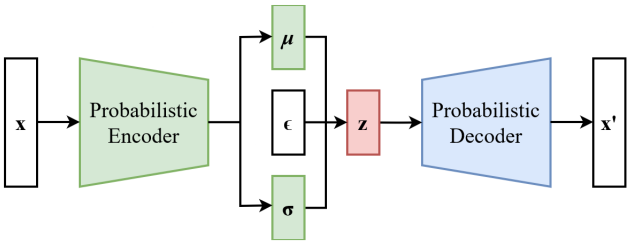

Figure 1: Architecture of a standard VAE [12]

3.2 Uncovering mutational signatures

In the area of uncovering mutational signatures from a mutational frequency matrix a commonly used approach is non-negative matrix factorization (NMF). NMF algorithms factorize an input matrix X into two matrices W and H, with all three matrices having all positive elements [13]. This is essential as X is the mutational frequency matrix, the rows of which are mutation types and columns of cancer samples, the entries in the matrix count the times a specific mutation type has occurred within each cancer sample. The W matrix has the NMF components, which in this case are mutational signatures. A mutational signature can be described as a vector of length 96, each entry has the rate at which the specific mutation type occurs in a cancer sample, and the sum of these rates adds to 1 if it has been normalized. The H matrix, its rows has the mutational signatures and the columns are the cancer samples, each entry is the amount of mutations a mutational signature has contributed to in every cancer sample. Summing a column in this matrix would have the total mutation count for that cancer sample. The product of W and H approximates X,

which can be written as $WH \approx X$. NMF can be used for source separation, as it is in this case.

Autoencoders can be constructed to mimic the same concept, where the product of the latent space z and the decoder weight matrix W reconstruct the dataset X used as input for the model. This concept can be written as $\hat{X} = zW^T$, where $\hat{X}$ is the reconstructed dataset. Using this architecture the mutational frequency matrix would be X, z the exposures, and W the mutational signatures.

3.3 MUSE-XAE framework

The variational autoencoder model operates within the framework developed by Pancotti et al. [8], where it replaces the autoencoder model described by [8]. In this framework, the mutation frequency p for each of the mutational types gets determined. New data points are generated by bootstrapping cancer samples t times, using a multinomial distribution $\mathcal{M}(N, p)$, where N is the total number of mutations. This sequence produces the augmented mutational matrix, which gets repeated t times to increase the size of the dataset. This augmented mutational matrix is subsequently used to train the model. To select the optimal amount of signatures K, a revised version of the NMFk approach was utilized. NMFk was originally described in [14]. Specifically, for each number of candidate signatures k, the model is trained n times. Afterward, k-means clustering with matching is performed on the set of decoder weights matrices $\{W_{1k}...W_{nk}\}$, using cosine similarity as a distance measurement. This results in finding a consensus signatures matrix S_k. The k-means clustering uses the Jonker-Volgenant algorithm [15] in the linear assignment problem to find k clusters of equal size n. Only solutions with a mean- and minimum silhouette score above a predetermined threshold are considered. Finally, the signature matrix with the lowest reconstruction error is the optimal solution [8].

After finding the optimal signature matrix S_k it gets normalized, and the matrix is used to initialize the decoder weight matrix W. The model trains a second time, where the decoder is frozen, which means it is not trainable. This process is referred to as refitting, its objective is to find the exposure of the mutational signatures within each cancer sample.

4 Methodology

This section describes the datasets and databases utilized in this paper, as well as the framework within which the model operates. The process of converting the MUSE-XAE autoencoder [8] into a variational autoencoder, along with the decisions made during this transformation, will also be detailed.

4.1 Datasets

Each scenario has a mutational frequency matrix that the variational autoencoder trains on. The mutational frequency matrix is a 96 x n matrix, where every row is a mutation type, and the columns are the cancer genome samples. For every mutational frequency matrix, there is a 96 x m matrix and a corresponding m x n matrix, where m is the number of mutational signatures. The 96 x m matrix denotes the signatures, and the m x n matrix denotes the exposure of each signature within each cancer sample. The signature can be seen

as a vector of length 96, often summing to 1 as it has been normalized. Every scenario was fetched from an FTP server (`ftp://alexandrovlab-ftp.ucsd.edu/pub/publications/Islam_et_al_SigProfilerExtractor/`) provided by the authors of Islam et al. [7].

PCAWG

PCAWG is an international effort to identify mutational patterns in more than 2,800 cancer whole genomes from the International Cancer Genome Consortium. To facilitate apt comparison, the tumors have been subjected to rigorous quality control testing. The research has been coordinated by a series of work groups comprising 700 scientists [16].

COSMIC

COSMIC is a database of mutational signatures. It was created by extracting mutational signatures from the PCAWG dataset made available by the ICGC Data Portal. It is described as a reference set of high-confidence signatures, curated by experts in the field [17], some of the signatures do not have known causation and are less established. The naming conventions follow the structure of starting with the mutation category, SBS in this case, and ending with numbering to distinguish them. An example of such a signature can be seen in Figure 2, with mutation types on the x-axis and rates on the y-axis.

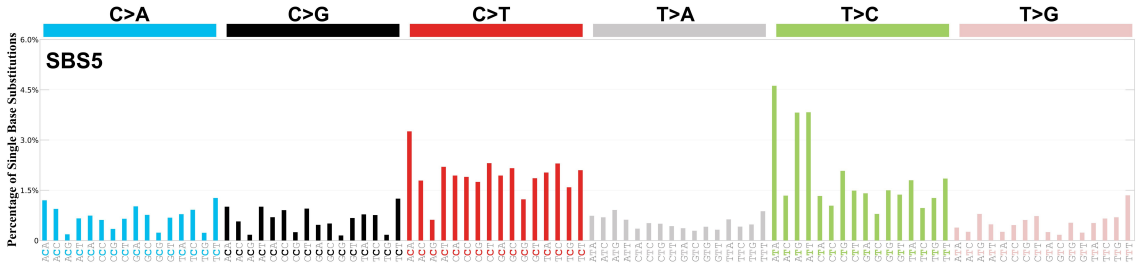

Figure 2: SBS-5 from the COSMIC database

Scenario 2

Based on adenocarcinoma, which is the most common type of pancreatic cancer, accounting for 90 percent of pancreatic cancer diagnoses. The SBS signatures primarily contributing include SBS1, SBS3, and SBS5 amongst others [4]. It contains 1,000 synthetic samples, which model a subset of the PCAWG dataset. In total, 11 ground-truth signatures were based on COSMIC [7].

Scenario 4

Based on renal cell carcinoma (most common kidney cancer) and ovarian adenocarcinoma. The tumors were generated using flat, relatively featureless mutational signatures in a realistic context, one of which can be seen in Figure 2. The primary contributing signatures include SBS3, SBS5, and SBS40 [4]. It contains 1,000 synthetic samples, 500 of each cancer. In total, there are 11 ground-truth signatures that were based on COSMIC [7].

Scenario 6

Based on transitional cell carcinoma of the bladder and skin melanoma. The tumors were generated using signatures with overlapping and potentially interfering signature profiles. The primary contributing signatures include SBS2, SBS13, SBS7a, and SBS7b amongst others [4]. It contains 1,000 synthetically generated samples. In total, 11 ground-truth signatures were based on COSMIC. The potential interference between SBS2 and SBS7a, SBS7b [7].

Scenario 8

Based on renal cell carcinoma and ovarian adenocarcinoma. The tumors were generated using flat, relatively featureless mutational signatures in a realistic context. Renal cell carcinoma has a lot of load from SBS5 and SBS40, whereas ovarian adenocarcinoma has a high load from SBS3 [4]. It contains 1,000 synthetically generated samples, 500 of each cancer. In total, there are 3 ground truth signatures based on COSMIC. This dataset was generated in a simplified fashion, where only three 3 signatures are present [7].

Scenario 14

Whole genome samples that match the ones found in PCAWG. It includes 300 spectra of 9 cancer types, these cancer types being: bladder transitional cell carcinoma, esophageal adenocarcinoma, breast adenocarcinoma, lung squamous cell carcinoma, renal cell carcinoma, ovarian adenocarcinoma, osteosarcoma, cervical adenocarcinoma, and stomach adenocarcinoma. Totaling in 2,700 cancer samples. In total, there are 21 ground truth signatures based on COSMIC [4]. This is a synthetic recreation of the overall PCAWG dataset to ascertain the performance with a dataset trying to mimic the mixture of signatures and cancers [7].

4.2 Variational autoencoder construction

In this chapter, we describe the steps to convert the MUSE-XAE autoencoder into a variational autoencoder. We chose MUSE-XAE as a starting point because it has demonstrated superior results in uncovering mutational signatures compared to NMF methods such as SigProfilerExtractor proposed by Alexandrov et al. [5]. As can be seen in Figure 3 the encoder consists of 3 linear layers. L1 is a linear layer with 96 input neurons and 96 output neurons. L2 is a linear layer with 96 input neurons and 48 output neurons. L3 is a linear layer with 48 input neurons and 24 output neurons. It uses layer normalization and GELU activation function between the linear layers. The output of L3 is reduced to the latent dimension in the μ and σ linear layers. The results of the μ and σ are passed through the sampling layer, which produces the latent space z. The decoder has a singular linear layer, with a non-negative weight constraint and a minimum volume regularizer. The exposures will be captured in the latent space z and the signatures within the weights of the decoder, where each signature is a vector of size 96, making up a matrix of size 96 x m. Where m stands for the number of mutational signatures.

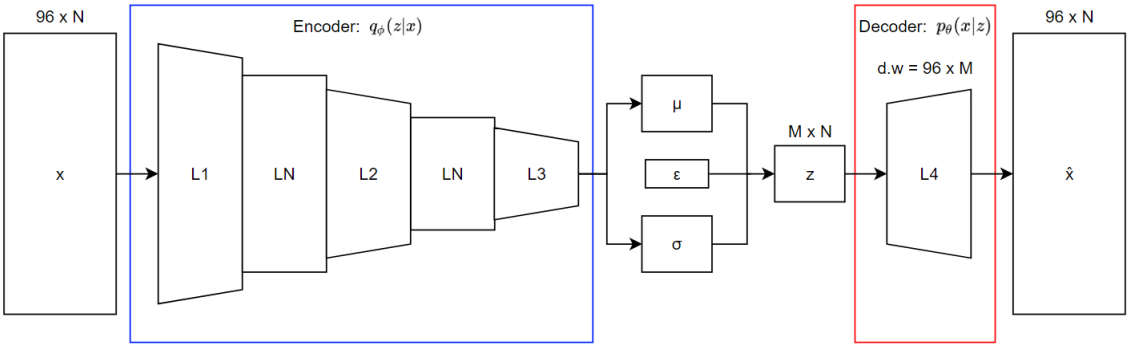

Figure 3: Conversion of the MUSE-XAE autoencoder [8] into a VAE, with the changes detailed in this chapter

4.2.1 Distributions

The prior distribution is a half-normal distribution derived from a Gaussian distribution with a mean of 0 and a standard deviation of 1. Meaning, it would be a fold at the mean of 0 of a standard normal distribution. The posterior distribution is a folded normal distribution. The reasoning behind using these distributions is that it forces the latent space to exclusively have positive numbers, which is necessary for modeling mutational count data. Additionally using a Poisson distribution is impossible as using a discrete probability distribution does not work with the reparameterization trick [18].

The folded normal distribution [19] is related to the Gaussian distribution, as it is a transformation of the Gaussian distribution using the absolute value. A folded normal distribution records the magnitude for each absolute value, as the sign is not considered. This is the reasoning behind its name, as the distribution's negative entries are folded onto its positive counterpart.

A half-normal distribution [19] is a special case of a folded normal distribution. It is a transformation of the Gaussian distribution $\mathcal{N}(0,\sigma^2)$ using the absolute value. Thereby, it is a fold at the mean of an ordinary normal distribution when the mean is 0.

However, a point of contention arises when using a folded normal distribution, as it is impossible to calculate the ICDF (Inverse Cumulative Distribution Function) since it has an intractable integral. The ICDF can be used to get the likelihood of realizing a random variable within a given range for a probability distribution. Meaning, it could be used to derive a confidence interval. Thereby, a method of gaining the confidence interval for a folded normal distribution is by sampling from the distribution and examining the numbers at the desired percentile.

4.2.2 Variational autoencoder definition

The function in Equation 2 is the encoder, that is responsible for mapping the matrix X into the latent space z. The function in Equation 3 is the decoder, which is responsible for mapping the latent space into a reconstruction matrix $\hat{X}$. The latent space follows the

posterior distribution, which is a folded normal distribution in this instance, mathematically it is written in Equation 4.

$$f(X) = z \tag{2}$$

$$g(z) = \hat{X} \tag{3}$$

$$z \sim \text{FoldedNormal}(\mu, \sigma^2) \tag{4}$$

4.2.3 Loss function

The loss function is defined as:

$$Loss = RL + \beta \cdot KL + \alpha \cdot \det(WW^T + I) \tag{5}$$

Where the Kullback-Leibler (KL) divergence [20] measures the distance between the prior and posterior distributions. The mathematical definition of KL divergence between two Gaussian distributions, where the prior has been set to $\mathcal{N}(0,1)$ and the posterior has a variable mean and standard deviation:

$$\begin{aligned} D_{KL}(N(\mu, \sigma^2)||\mathcal{N}(0,1)) &= log(\sigma) + \frac{1+\mu^2}{2\sigma^2} - \frac{1}{2} \\ &= \frac{1}{2}\left(log(\sigma^2) + \frac{\mu^2+1}{\sigma^2} - 1\right) \\ &= -0.5\left(1 + log(\sigma^2) - \mu^2 - \sigma^2\right) \end{aligned} \tag{6}$$

As evident by the definition of KL divergence, it is used to measure how different two probability distributions are. The KL divergence in variational autoencoders is used to push a posterior distribution towards a prior distribution, where the prior distribution is predetermined.

The β variable is used to control the impact of the KL divergence, in this instance, the value of the β variable has been set to varying values which will be expounded upon during the experiments. The reconstruction loss (RL) used in this instance is the Poisson loss function, Poisson is commonly utilized to model count data [21].

The KL divergence is calculated using the formula, which was derived in Equation 6:

$$KL = -0.5\left(1 + log(\sigma^2) - \mu^2 - e^{log(\sigma^2)}\right) \tag{7}$$

where taking the exponential of the expression $log(\sigma^2)$ is equivalent to the variance i.e. σ^2. The reasoning behind using the KL divergence between two Gaussian distributions instead of the KL divergence between a folded normal distribution and a half-normal distribution is that the latter does not have a closed-form solution. This simplifies the problem significantly and provides an estimate as to the KL divergence between the corresponding half-normal distribution and folded normal distribution.

In Equation 7, μ^2 penalizes means far from zero, ensuring the posterior centers around the mean of the prior. $log(\sigma^2)$ penalizes very large and very small values as it moves away from 1. $e^{log(\sigma^2)}$ directly penalizes large values of σ^2, and since it appears with a negative sign a higher value of σ^2 increases the KL divergence, contributing to the penalty. Thereby,

minimizing the term favors σ^2 being close to 1. This means the posterior near an underlying $\mathcal{N}(0,1)$, which becomes a half-normal(1) when taking the absolute value.

The RL is calculated using the formula as provided by [22]:

$$RL = \hat{X} - X \cdot log(\hat{X}) \tag{8}$$

The Poisson loss function, also known as negative log-likelihood for the Poisson distribution, is commonly used to model count data [21]. Firstly, $\hat{X}$ discourages the model from predicting higher values than necessary, pushing the model to not overestimate counts. Secondly, $-X \cdot log(\hat{X})$ encourages the model to predict values as close as possible to the true counts, as an accurate prediction would ensure that $log(\hat{X})$ is close to $log(X)$, leading to minimized loss. $log(X!)$ has been omitted as it is constant w.r.t to the inputs.

The last term, $\alpha \cdot \det(WW^T + I)$, is the minimum volume regularizer used by Pancotti et al. [8]. It is part of the decoder with the objective of finding a more disentangled representation. α is used to control the strength of the regularizer and is 0.001 by default.

4.2.4 Sampling

The sampling is accomplished by using the formula:

$$x = |\mu + \sigma \cdot \epsilon| \tag{9}$$

Sampling from the underlying normal distribution that the folded normal distribution is modeling and taking the absolute value is equivalent, to directly sampling from the folded normal distribution. The ϵ variable is drawn from a standard Gaussian distribution.

4.2.5 Layers, normalization and activation functions

First and foremost batch normalization was changed to layer normalization as it does not infer bias within the samples, also, batch normalization is mostly used when the scale of the data is important to preserve. However, before the data goes into the model it has been normalized, therefore it removes the incentive to use batch normalization.

Additionally, the last dense layer of the encoder was changed to two dense layers, one for the means and one for the log-variance. Also, it includes a sampling step where it utilizes these means and log-variances to construct the latents as demonstrated in Equation 9.

No activation function is employed after the two new dense layers as the goal of one would only be to ensure the data is positive, this is already accomplished by sampling from a folded normal distribution. Furthermore, using ReLU would only ensure the mean and log-variance is positive. Thereby, if the mean was sufficiently close to 0 it could still generate negative samples when using a Gaussian distribution as the standard deviation could swing the samples into the negatives.

GELU (Gaussian Error Linear Units) is the activation function between each linear layer of the encoder. It weights inputs by their percentile, rather than gating them by their sign. Thus, it can be thought of as a smoother ReLU [23]. The activation function in [8] is softplus in the signature extraction phase and ReLU during refitting. Hendrycks and Gimpel [24] finds that GELU improves performance across various machine intelligence tasks. Anecdotally, performance improvements were noticed when converting to GELU.

4.2.6 Exposures (latent space)

After the model has been trained, the exposures are calculated using inference. Firstly, the means and variances of the posterior distribution still follow a Gaussian distribution as it applies the absolute function in the sampling step. Thereby, the means and variances need to be converted to the corresponding means given a folded normal distribution, by using the formula in Equation 10 [25]. Once this is accomplished, the folded mean of each signature is compared to the total sum of folded means within a cancer sample, and the percentage each signature makes up is multiplied by the mutation count for the given sample in the input matrix to calculate the scaled exposures.

$$folded_\mu = \sigma \cdot \sqrt{\frac{2}{\pi}} \cdot e^{-0.5\left(\frac{\mu^2}{\sigma}\right)} + \mu \cdot erf\left(\frac{\mu}{\sigma \cdot \sqrt{2}}\right) \tag{10}$$

4.2.7 Confidence intervals

The dataset gets passed through the trained model from the refitting part. The exposures of each cancer sample get sampled 1000 times using the posterior distribution. The 1000 samples are the input for the percentile function [26], which determines the samples at the 0th and 95th percentile. The percentiles are scaled by the same factor as the means of each signature described in Section 4.2.6.

5 Experiments

In this chapter, the results regarding the effects of utilizing a variational autoencoder will be explored. Both in terms of accurately determining the underlying signature and whether the statistical component provides a meaningful benefit to assessing the exposures of signatures.

5.1 Procedure

In conducting the experiments, three distinct configurations of the framework were used to evaluate our solution.

- The first configuration utilizes the VAE to reveal these signatures by leveraging probabilistic inference in the latent space. The objective of this configuration is mainly to determine the ability of the VAE to extract signatures, secondary to estimating exposures and confidence intervals using these signatures.

- The second configuration employs the MUSE-XAE autoencoder to uncover signatures embedded within the input data. The objective of this configuration is to determine the ability of the VAE to estimate exposures and confidence intervals in the refitting process given signatures extracted by a state-of-the-art method.

- The third configuration skips the signature extraction step and is directly given the ground truth signature to perform the refitting step. The objective of this configuration is to determine the ability of the VAE to estimate exposures and confidence intervals in the refitting process given the optimal case.

For all three configurations, the VAE was utilized to generate exposures during the refitting process. The parameters (excluding β) for the models were: learning rate of 0.001 and Adam Optimizer. The latent space size varied throughout the uncovering of signatures. The optimal number of signatures was selected by the MUSE-XAE framework, the latent size was hereby determined before the refitting. It is limited to 1000 epochs for uncovering signatures, however, it uses early stopping with the patience set to 30. Refitting is limited to 10000 epochs, it uses early stopping with the patience set to 100. In each of the three variations the tests of the β value were conducted using every scenario, where the β value ranged from 0 to 1.

5.2 Metrics and baselines

To compare signatures we utilize cosine similarity, which is standard practice. A cosine similarity of 1 indicates the signatures are identical, whereas a cosine similarity of 0 indicates entirely independent signatures. Mean Squared Error (MSE) is used when calculating the squared difference between the actual exposures and the exposures predicted by the refitting part. The metric, lower, refers to the actual exposures above the lower bound of the confidence interval, upper refers to actual exposures below the upper bound of the confidence interval and coverage refers to the actual exposures within the coverage of the confidence interval.

5.3 Signature extraction

In this section the effects of involving KL divergence as part of the loss function when optimizing the signature uncovering part of the model, specifically the relation of the cosine similarity between the extracted and ground truth signatures will be explored. This will be done while changing the β value to observe its impact.

Dataset	$\beta = 1$	$\beta = 0.1$	$\beta = 0.01$	$\beta = 0.001$	$\beta = 0$
Scenario 2	0.669 ± 0.072	0.660 ± 0.075	0.739 ± 0.137	0.881 ± 0.141	0.973 ± 0.036
Scenario 4	0.833 ± 0.062	0.821 ± 0.064	0.833 ± 0.061	0.829 ± 0.096	0.967 ± 0.050
Scenario 6	0.693 ± 0.121	0.693 ± 0.119	0.813 ± 0.117	0.925 ± 0.085	0.967 ± 0.064
Scenario 8	0.889 ± 0.040	0.927 ± 0.036	0.926 ± 0.037	0.922 ± 0.045	0.965 ± 0.035
Scenario 14	0.633 ± 0.112	0.656 ± 0.109	0.741 ± 0.108	0.853 ± 0.126	0.966 ± 0.039

Table 1: The avg. cosine similarity and standard deviation for varying β values in each scenario

Figure 4 displays a signature extracted using the first configuration outlined in Section 5.1. The ground truth signature is SBS3 from the COSMIC database, which is located at the bottom of the figure. The extracted signature is VAE-SBSA, which is located at the top of the figure. The signatures presented have a cosine similarity match of 0.9813, which indicates that they resemble each other closely. These signatures were explicitly chosen as they had a cosine similarity match of 0.9813, which is close to the average of 0.973 when using a β value of 0.

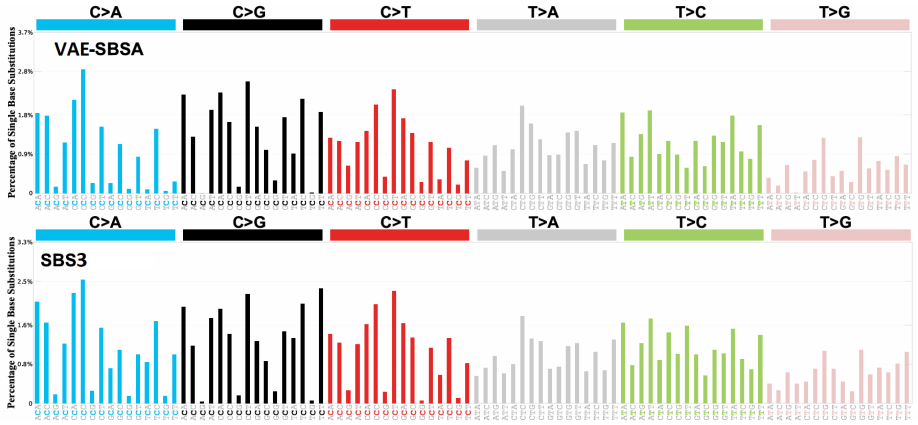

Figure 4: Sig. match when $\beta = 0$, the x-axis denotes mutation types and the y-axis denotes the percentage a specific mutation type makes up of the total signature composition

Figure 5 displays a signature extracted using the first configuration outlined in Section 5.1. The ground truth signature is SBS40a from the COSMIC database and is located at the bottom of the figure. The extracted signature is VAE-SBSG and is located at the top of the figure. The signatures presented have a cosine similarity match of 0.6814, which indicates that they loosely resemble each other. These signatures were chosen to provide a signature match that had a cosine similarity close to the average of 0.678 when using a β value of 1.

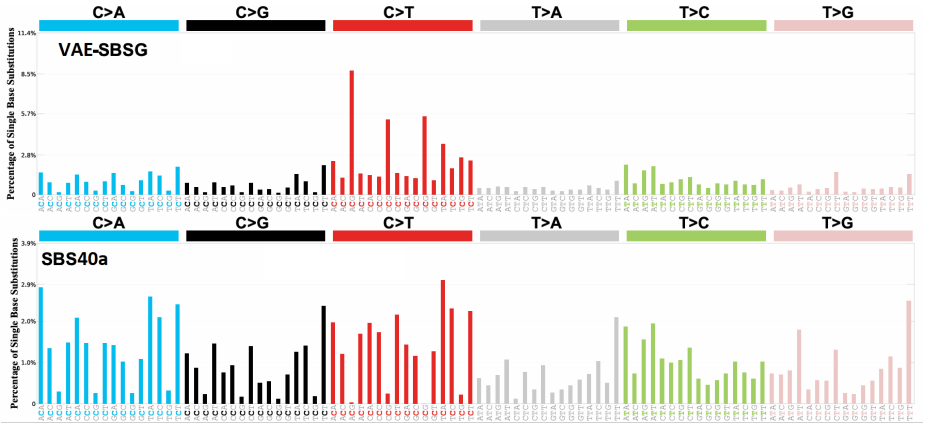

Figure 5: Sig. match when $\beta = 1$, the x-axis denotes mutation types and the y-axis denotes the percentage a specific mutation type makes up of the total signature composition

As evidenced by the results showcased in Table 1, Figures 4 and 5 the VAE extracts signatures matching the ground truth better the lower the β value is set. This is illustrated by the fact that it has a higher averaging cosine similarity match along with a lower standard

deviation. Thereby, the VAE will use a β value of 0 when conducting the experiments from which the F1-score, precision and sensitivity will be calculated across the five scenarios. Table 2 showcases a comparison of prominent signature extraction tools, which predominantly utilize NMF with the exceptions of MUSE-XAE and the VAE that were proposed in this paper, which both use variations of autoencoders.

Method	**AUC Precision**	**AUC Sensitivity**	**AUC F1-score**
MUSE-XAE	0.911 ± 0.057	0.929 ± 0.035	0.920 ± 0.044
SigProfilerPCAWG	0.893 ± 0.069	0.910 ± 0.047	0.901 ± 0.056
SigProfilerExtractor	0.889 ± 0.070	0.913 ± 0.043	0.901 ± 0.055
SigneR	0.869 ± 0.090	0.910 ± 0.117	0.887 ± 0.094
SignatureAnalayzer	0.855 ± 0.085	0.905 ± 0.031	0.878 ± 0.054
VAE (Ours)	0.857 ± 0.090	0.889 ± 0.054	0.873 ± 0.072
SignaturesToolsLib	0.843 ± 0.081	0.868 ± 0.066	0.854 ± 0.072
MutationPatterns	0.804 ± 0.105	0.921 ± 0.031	0.856 ± 0.066
MutSpec	0.760 ± 0.144	0.919 ± 0.034	0.827 ± 0.095
SomaticSignatures	0.682 ± 0.187	0.860 ± 0.082	0.754 ± 0.142
Maftools	0.639 ± 0.266	0.809 ± 0.130	0.695 ± 0.220
SigMiner	0.541 ± 0.202	0.850 ± 0.119	0.652 ± 0.192

Table 2: Comparison of tools in descending order based on **AUC F1-score** averaged across the five scenarios, these results are calculated using the Jupyter Notebook meant for reproducing the results of Pancotti et al. [8], located at `https://github.com/compbiomed-unito/MUSE-XAE/blob/main/notebook/Paper_results_reproducibility.ipynb`

As can be observed in Table 2 the VAE performs slightly lower than average comparatively. Therefore, in the experiments concerned with estimating exposures the second and third variations outlined in Section 5.1 will be utilized. These two disparate variations will then be compared to capture the significance of having the exact signatures.

5.4 Estimating exposures

Similarly in the experiments showcased in Table 3 the effects of varying the β value can be observed, it uses the second configuration detailed in Section 5.1 on scenario 2. The entries are the calculation ascertaining the % of samples contained within the lower, upper, and coverage of the confidence interval starting at the 0th percentile and extending until the 95th percentile.

Metrics	$\beta = 1$	$\beta = 0.1$	$\beta = 0.01$	$\beta = 0.001$	$\beta = 0$
Coverage	69.69	69.59	73.69	80.73	22.18
Lower	83.54	83.29	84.88	90.38	41.79
Upper	86.15	86.30	88.81	90.35	80.39
MSE	1426386	1406328	1246748	811503	62718

Table 3: The % of samples that exists within the lower, upper and both bounds in addition to the MSE (Mean Squared Error) of the exposures as compared to the true values, for varying β values (using the second configuration on scenario 2)

The analysis of these results, however, is ambiguous. A β value different from 0 does appear to increase the amount of samples encompassed within the coverage. Additionally, the MSE increases, signifying the exposure amounts are inaccurately estimated. Furthermore, the exposures of each signature are observed to be identical within the same sample, indicating that the latent space has become uninformative. Also, the lower bound can be largely discarded as it is virtually impossible for the folded normal distribution to include the exposure amount of 0, which a lot of samples contain. To conclude the chosen β value for upcoming experiments across all scenarios will be 0. In Table 4 the results can be seen using signatures extracted using the second variation outlined in Section 5.1, where approximately 72.47% of the ground truth exposures are below the upper bound on average.

Metrics	**Scenario 2**	**Scenario 4**	**Scenario 6**	**Scenario 8**	**Scenario 14**
Coverage	22.18	8.24	19.86	19.85	1.79
Lower	41.79	34.97	40.94	76.45	15.43
Upper	80.39	73.27	78.92	43.40	86.37
MSE	62718	78338	99156814	253131	35568035

Table 4: The % of samples that are encompassed within the lower bound, upper bound, and coverage in addition to the MSE of the exposures as compared to the true values, across the five scenarios (using the second configuration)

Lastly, the ground truth signatures will be inserted into the decoder weights to ascertain the performance of the refitting if the extraction part is ideal, this is the third configuration outlined in Section 5.1. Table 5 shows the results, where 76.05% of the ground truth exposures are below the upper bound on average. The MSE also decreased in every scenario, apart from Scenario 6, meaning it resulted in an improvement when it comes to the correctness of predicted exposures.

Metrics	Scenario 2	Scenario 4	Scenario 6	Scenario 8	Scenario 14
Coverage	30.05	38.10	8.53	45.40	7.47
Lower	53.12	529.1	41.18	79.30	22.79
Upper	76.93	85.19	67.35	66.10	84.69
MSE	4205	11025	28437795	13417	9501559

Table 5: The % of samples that are encompassed within the lower bound, upper bound and coverage in addition to the MSE of the exposures as compared to the true values, across the five scenarios (using third configuration)

6 Discussion

This chapter discusses the results, their interpretation, potential alternative approaches, and possible factors influencing the outcomes.

6.1 Prior distribution

Throughout conducting experiments, it was deduced that a β value of 0 produced the best results, both when it comes to extracting signatures and when estimating exposures. This implies that the more the latent space gets pushed towards a half-normal distribution derived from $\mathcal{N}(0,1)$ the worse it performs. This could mean that the selected prior distribution does not match the data. Employing the half-normal distribution was an initial decision since sampling the posterior distribution should only result in positive numbers. However, as was previously mentioned, an equally valid distribution for modeling the latent space is the exponential distribution. Using the exponential distribution would still fulfill the requirement of having positive numbers in the latent space, also, the formula for calculating the KL divergence between two exponential distributions is in closed form [27]. This means the KL divergence can be directly calculated, and no estimations are needed. Lastly, the reparameterization trick is possible on this distribution and it is continuous.

The model could theoretically employ a mixture distribution [27] as the prior distribution, utilizing the VAE to learn the mixture through the KL divergence term in the loss function. This approach would enhance the expressiveness of the VAE, enabling it to learn a more complex representation of the data, rather than conforming to a simple half-normal distribution derived from a standard normal distribution $\mathcal{N}(0,1)$. The mixture could be comprised of various distributions, such as folded normal distributions or exponential distributions.

6.2 KL divergence and beta

A key difference between a standard autoencoder and a variational autoencoder is the addition of the KL divergence as part of the loss function. Specifically, the variational autoencoder developed is a β-VAE, whose loss function has β variable in front of the KL divergence to control its influence on the total loss. When varying the β value across multiple datasets a pattern was observed, where a lower β value produced better results both in terms of extracting signatures fitting the ground truth signatures and accurately

estimating exposures. The cosine similarity match between the signatures extracted by the model and the underlying ones increased by an average of 32.4%, when comparing the extremes. [5] mentions a cosine similarity match of 0.95 indicates that the signatures are identical, as is evident from Table 1 the instances where it surpasses this threshold is when β is set to 0. This is fairly uncommon as [11], which proposed the β-VAE, does not mention the setting β to a value less than 1. Also, when estimating the exposures in scenario 2 a similar pattern was observed. As is showcased in Table 3, the MSE increased from 55500 to 1425701 when varying β from 0 to 1. This indicates the estimated exposures are more accurate when using a β less than 1. The observations within the lower bound and the coverage of the confidence interval did increase, however, this is not significant as it was more inaccurate in the predicted exposures. The model generally has a hard time predicting an exposure amount of 0, as it samples from a folded normal distribution, which can account for these observations. When using a higher β the means and variances of each sample were nearly identical, a lot more variation was observed when decreasing the β.

A β of 0 means the model has a loss function equivalent to an autoencoder. Additionally, the posterior distribution is no longer pushed towards the prior distribution. The latent space is still constructed by sampling from this posterior distribution. Although it is not explicitly pushed towards a prior, the VAE still attempts to find means and log-variances resulting in the best outcomes for reconstructing the data.

6.3 Latent space

Examining the latent space of the MUSE-XAE autoencoder and the VAE with varying β by using t-SNE plots with 2 components reveals a pattern, the plots of which are shown in Figure 6. By increasing the β, the variational autoencoder appears to be experiencing a phenomenon known as posterior collapse. Posterior collapse happens when the latent space becomes uninformative, and the model solely relies on the decoder to reconstruct the data [28]. The t-SNE plots are based on the latent space produced by the model when trained on scenario 14. This scenario was chosen as it has 9 different cancer types, making the problem more pronounced. The cancer samples have been colored based on the cancer type. Ideally, the samples are clearly separated into clusters, as they are in the upper left and upper right quadrant of Figure 6. As mentioned in [28] many papers have proposed ways of solving this problem, such as more complex priors and modified decoder architectures. Tomczak and Welling [29] proposes the VampPrior, which is a mixture of variational posteriors. Ultimately, it is difficult to know exactly what causes the posterior collapse.

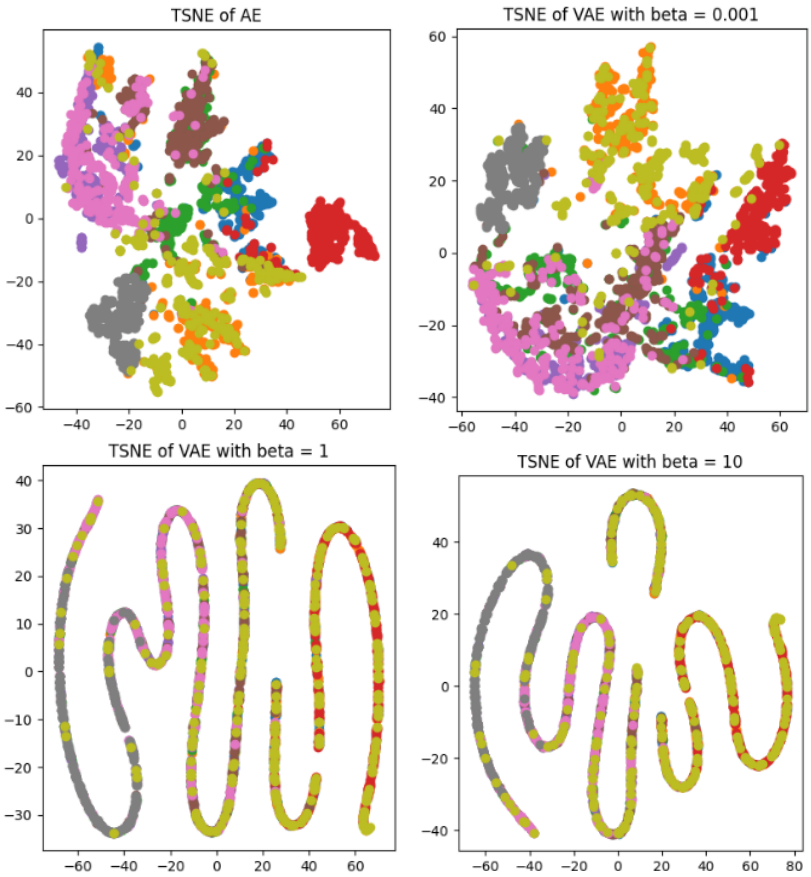

Figure 6: t-SNE plots of the latent space when trained on scenario 14

7 Conclusion

In this paper, the aim was to create a model that could provide confidence intervals, exposures, and mutational signatures from a mutational frequency matrix. The confidence intervals would have the application of providing a certainty estimate on exposure for given signatures. To achieve this the MUSE-XAE framework [8] was extended by adding the possibility of arriving at confidence intervals. This was accomplished by replacing the autoencoder with a β-VAE, in addition to providing the necessary changes to the framework to handle the findings of the β-VAE. By training the β-VAE on the different scenarios, it was found that a β of 0 produced favorable results. Signatures extracted using the variational autoencoder had an average cosine similarity match varying from 0.965-0.973 when using a β of 0, it ranged from 0.633-0.889 when using a β of 1 depending on the scenario. This corresponds to an average increase of 32.4% to the cosine similarity match when comparing the results produced by the various β values.

Similarly, the exposures derived with the model using a β of 0 had an MSE of 62718 compared to the ground truth exposures. The MSE increased to 1426386 once β was set to 1. The observations within the lower bound and coverage of the confidence interval did increase, however, this was not significant as the predicted exposures were more inaccurately

estimated. This can mostly be explained by the fact that the folded normal distribution by nature rarely predicts the exposure to be 0, which is the case for many cancer samples. With this in mind, the most likely case is that the half-normal distribution selected does not accurately model the data. Resolving this dispute could possibly be achieved by modeling it after the exponential distribution instead. Otherwise, using a mixture distribution could be an interesting future direction as it allows for complex probabilistic modeling, where multiple distributions are used to explain the data.

8 Acknowledgements

We would like to thank our supervisors, Daniele Dell'Aglio and Rasmus Froberg Brøndum, for their guidance, support, and encouragement throughout this project. Their expertise and insights were instrumental in the development and completion of this work.

9 Code Availability

The code developed to achieve the goal of this project can be found at the GitHub repository (`https://github.com/TheGislum/DRP10`).

References

[1] Kornelius Schulze, Sandrine Imbeaud, and Eric Letouzé. Exome sequencing of hepatocellular carcinomas identifies new mutational signatures and potential therapeutic targets. *nature*, 2015.

[2] Maria Secrier, Xiaodun Li, and Nadeera de Silva. Mutational signatures in esophageal adenocarcinoma define etiologically distinct subgroups with therapeutic relevance. *nature*, 2016.

[3] Jennifer Ma, Jeremy Setton, Nancy Y. Lee, Nadeem Riaz, and Simon N. Powell. The therapeutic significance of mutational signatures from dna repair deficiency in cancer. *nature*, 2018.

[4] COSMIC. Single base substitution (sbs) signatures, 2024. URL `https://cancer.sanger.ac.uk/signatures/sbs/`.

[5] Ludmil B. Alexandrov, Serena Nik-Zaina, David C. Wedge, Peter J. Campbell, and Michael R. Stratton. Deciphering signatures of mutational processes operative in human cancer. *Cell Reports*, 2013.

[6] Ludmil B. Alexandrov, PCAWG Mutational Signatures Working Group, and PCAWG Consortium. The repertoire of mutational signatures in human cancer. *Nature*, 2020.

[7] S.M. Ashiqul Islam, Steven G. Rozen, and Ludmil B. Alexandrov. Uncovering novel mutational signatures by de novo extraction with sigprofilerextractor. *Cell Genomics*, 2022.

[8] Corrado Pancotti, Cesare Rollo, Giovanni Birolo, Piero Fariselli, and Tiziana Sanavia. Muse-xae: Mutational signature extraction with explainable autoencoder enhances tumour type classification. *biorxiv*, 2023.

[9] Guangsheng Pei, Ruifeng Hu, Yulin Dai, Zhongming Zhao, and Peilin Jia. Decoding whole-genome mutational signatures in 37 human pan-cancers by denoising sparse autoencoder neural network. *Oncogene*, 2020.

[10] Keras. Variational autoencoder, 2024. URL `https://keras.io/examples/generative/vae/`.

[11] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. -vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[12] Wikipedia. Variational autoencoder, 2024. URL `https://en.wikipedia.org/wiki/Variational_autoencoder#/media/File:Reparameterized_Variational_Autoencoder.png`.

[13] scikit learn. Nmf, 2024. URL `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html`.

[14] Benjamin T. Nebgen, Raviteja Vangara, Miguel A Hombrados-Herrera, Svetlana Kuksova, and Boian S. Alexandrov. A neural network for determination of latent dimensionality in nonnegative matrix factorization. *IOPscience*, 2020.

[15] Ray Jonker and Ton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Springer*, 1987.

[16] ICGC. Pcawg - pancancer analysis of whole genomes, 2024. URL `https://dcc.icgc.org/pcawg`.

[17] COSMIC. Mutational signatures, 2024. URL `https://cancer.sanger.ac.uk/signatures/#introduction`.

[18] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh et al. The concrete distribution: A continuous relaxation of discrete random variables. *NIPS*, 2016.

[19] Kyle Siegrist. The general folded normal distribution, 2024. URL `"https://stats.libretexts.org/Bookshelves/Probability_Theory/Probability_Mathematical_Statistics_and_Stochastic_Processes_(Siegrist)/05%3A_Special_Distributions/5.13%3A_The_Folded_Normal_Distribution"`.

[20] The Book of Statistical Proofs. Proof: Kullback-leibler divergence for the normal distribution, 2024. URL `https://statproofbook.github.io/P/norm-kl.html`.

[21] UCLA. Poisson regression, 2024. URL `https://stats.oarc.ucla.edu/stata/dae/poisson-regression/`.

[22] Keras. Probabilistic losses, 2024. URL `https://keras.io/api/losses/probabilistic_losses/#poisson-class`.

[23] Paperswithcode. Gaussian error linear units, 2024. URL `https://paperswithcode.com/method/gelu`.

[24] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arxiv*, 2023.

[25] Wikipedia. Folded normal distribution, 2024. URL `https://en.wikipedia.org/wiki/Folded_normal_distribution`.

[26] NumPy. numpy.percentile, 2024. URL `https://numpy.org/doc/stable/reference/generated/numpy.percentile.html`.

[27] PyTorch. Probability distributions - torch.distributions, 2024. URL `https://pytorch.org/docs/stable/distributions.html`.

[28] David Wipf Bin Dai, Ziyu Wang. The usual suspects? reassessing blame for vae posterior collapse. *arxiv*, 2019.

[29] Jakub M. Tomczak and Max Welling. Vae with a vampprior. *arxiv*, 2019.