

# Benchmarking VAE latent features in downstream tasks for cancer related predictions

Boris van Groenigen

June 27, 2021

<b>Responsible Professor:</b>	Marcel Reinders
<b>Other Supervisors:</b>	Tamim Abdelaal, Mohammed Charrouf, Mostafa elTager, Stavros Makrodimitris
<b>Peer group members:</b>	Raymond d’Anjou, Armin Korkic, Ivo Kroskinski, Bram Pronk

## Abstract

Using RNA sequence data for predicting patient properties is fairly common by now. In this paper, Variational Auto-Encoders (VAEs) are used to assist in this process. VAEs are a type of neural network seeking to encode data into a smaller dimension called latent space. These latent features are then used to do downstream task analysis such as cancer types, survival time and cancer stages, with the help of a MLP classifier. Furthermore, the training process itself is also analyzed with the usage of UMaps. The purpose of this paper is to compare different VAE models on their effectiveness in providing training data used for the predictions. The predictions mostly consist of guessing when using any of the latent spaces, constructed by the VAE models, as input data for the MLP classifier. The NoVAE model is the only model with slightly better performance when it comes to mean accuracy and standard deviation. The guessing issue is further analyzed with the help of UMaps. The VAEs are able to classify the input data during the training process, but when faced with new data, this end up not being the case. Both the learning rate and  $\beta$  term yield interesting results regarding the modification of the input data and variational property respectively. A lower learning rate leads to better classification, but this is due it deviation less from the original input data. When using a small  $\beta$  term with the  $\beta$ -VAE, the output is similar to that of the VanillaVAE. Meaning the VanillaVAE does not perform better than a regular autoencoder.

## 1 Introduction

Cancer comes in many different forms and has taken many lives due to high lethality rating depending on where the tumor is located within the body according to Nuffield Trust [1]. Fortunately, molecular data such as RNA and DNA profiling, coming from sequencing technologies, can assist in studying and understanding cancer [2]. The analysis of such data, is, however, a challenging task due to the data being quite complex. In this paper, Variational Autoencoders (VAE) are used for interpreting this complex data. These VAEs extract key

pieces of data from a large data set in order to reconstruct them in a meaningful way. A VAE’s main purpose is to regularize the training process to avoid overfitting such that the latent space has good properties to enable a generative process [3]. These techniques allow us to take a different approach in predicting cancer related properties, such as cancer stage, survival time and cancer type. Normally, the raw data would just be used to do these predictions with the help of predictive models [4]. This brings us to the question: Are latent features more useful in predicting patient cancer related properties?

To answer this question, different types of VAEs will be compared with different prediction models, as well as training these models without the use of any VAE, so by just using the raw input data. Answering this research question can prove to be useful for future research into cancer predictions and prognosis. The conclusion will tell if VAEs have any added value in doing these predictions.

## 2 Methodology

### 2.1 VAE

A VAE seeks to reduce the dimensions of the input data and recreate the original data using this lower dimensionality latent space as seen in Figure 1. The VAE is a form of neural network which corrects itself based on the reconstruction loss when encoding and decoding to find the most optimal latent space. This loss is obtained from the comparison of the reconstruction with the input data and the KL divergence between the learned distribution and the standard Gaussian distribution [3, 5].

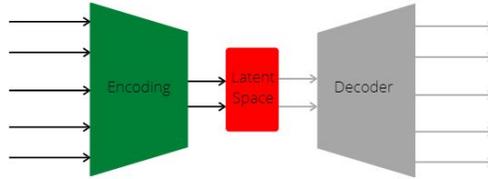


Figure 1: High level visualization of a VAE

As mentioned before, this lower dimensional data will be used to do predictions such as cancer stages, survival time and cancer types. Obtaining the latent features will be done by the use of different VAE models, which are:

- VanillaVAE
- $\beta$ -VAE
- $\beta$ -TCVAE
- Categorical VAE

The differences between these VAE lie in the construction of the latent features. The  $\beta$ -VAE seeks to discover disentangled latent factors [6], whilst the  $\beta$ -TCVAE is a refinement of the  $\beta$ -VAE more catered toward total correlation [7]. Finally, the Categorical VAE (CatVAE for short) makes it possible to learn the latent space of discrete random variables [8]. The most common initialization parameters will be used for these VAEs as seen in Table

1. These have partially been obtained from research colleagues/peers who dive deeper into initialization importance itself.

Besides these VAEs, the predictions will also be done using the input normally used for the VAE, thus skipping the dimensionality reduction process. This will assist in answering the research question - whether VAEs are useful for doing predictions. This method of doing predictions will be called NoVAE from here on out for the sake of simplicity.

Table 1: Initialization parameters for the VAE models.

	VanillaVAE	$\beta$ -VAE	$\beta$ -TCVAE	CatVAE
Learning rate	0.005	0.005	0.005	0.005
Epochs	40	40	40	40
Latent dimensions	30	30	30	30
$\beta$	-	0.1	1	-
$\alpha$	-	-	1	1
$\gamma$	-	-	1	-

## 2.2 Datasets

Since different cancer types can cause much different outcomes for patients, predictions for survival time and cancer stages can be less accurate when done over different cancer types. Therefore, three of the most common cancer types to compare VAE models will be used to make such predictions, which are:

- Breast cancer (BRCA) [9]
- Lung cancer (LUAD) [10]
- Kidney cancer (KIRC) [11]

To achieve this, two datasets are used. The first consisting of patient samples containing their RNA data, which is used to train the VAE models, is taken from The Cancer Genome Atlas (TCGA)[12]. The second dataset [13], survival time and cancer types by matching the patient IDs with the first dataset and taking their respective RNA data (TCGA). In addition, the 5000 most variable genes/features are taken instead of the original 20,000 in order to speed up the training process while maintaining as much of the key data as possible. These 5000 genes are also scaled to [0, 1] for easier feature comparisons. To clarify, the most common cancer types are based on the number of entries in the TCGA datasets and does not necessarily reflect real world commonness.

When one wants to make predictions for the breast cancer type and its survival time, then all breast cancer patients with present survival time information IDs will be taken from the RNA sequence dataset as well as a separate array containing the respective survival time for each patient. However, this does not always lead to a fruitful sample size. An overview of the number of samples and classes for each cancer type and their respective cancer property can be found below in Table 2.

Table 2: Number of samples per cancer type as well as the number of classes used for predictions. Note that the cancer type predictions use all three of the cancer type’s samples combined as a dataset.

	BRCA samples	LUAD samples	KIRC samples	Classes
Cancer stage	1190	567	603	4
Survival time	1214	567	606	3
Cancer type	2397			3

Keep in mind that these are the number of samples in the curated clinical dataset. This does not necessarily mean that the number of samples are equivalent to those in the RNA sequence dataset, due to the possibility of a patient having multiple entries in this dataset. If this is the case, all these samples will be taken for training purposes, since the sample size can become quite sparse. It might be possible that the patient sample does not exist in the RNA sequence dataset, whilst existing in the curated clinical dataset and is therefore left out.

Now that a general overview of the dataset has been shown, a more in-depth look into the different predictions can be given.

### 2.2.1 Cancer stages

The curated clinical dataset contains many different class possibilities for predictions. There are a total of 13 classes present. This is however quite a large number for the number of samples available when doing predictions. Thus, the classes are merged into 4 classes as seen in Table 3.

Table 3: Classes used for cancer stage predictions. In the first row are the classes used for the predictions made up by their sub-classes below.

Stage I	Stage 2	Stage 3	Stage 4	Not used
'Stage I'	'Stage II'	'Stage III '	'Stage IV'	'Stage X'
'Stage IA'	'Stage IIA'	'Stage IIIA'		'Discrepancy'
'Stage I'	'Stage IIB'	'Stage IIIB'		
		'Stage IIIC'		

### 2.2.2 Survival time

Survival time predictions are not amongst the most trivial to do, but it might still prove useful in comparing the VAE against one another. For this patient property the *OS.time* (Overall Survival Time) column is used. A continuous range prediction will not be done since this will lead to a considerable error. Instead, 3 classes will be constructed split by the lowest, middle and highest survival times for a classification approach as seen in Table 4. The classes are split by ascending *OS.time* in such a way to have the same number of samples in each class.

Table 4: *OS.time* range per class for each cancer type. Each of the classes has an equal number of samples.

	Low	Mid	High
BRCA	0 - 584	585 - 1363	1364 - 8605
LUAD	0 - 488	489 - 944	945 - 7248
KIRC	0 - 749	750 - 1731	1732 - 4537

### 2.2.3 Cancer types

Predicting what type of cancer a patient has based on their RNA sequence is also an interesting idea. Once again, the three most common cancer types are used. This is done because of the variance in sample sizes per cancer types, since they are not evenly distributed amongst types, ranging from 1190 (BRCA) to 36 (CHOL) samples. This will also increase the predictive models' accuracies and in turn help with the VAE comparisons.

## 2.3 Predictive models

The predictions are done by a neural network, a multilayer perceptron (MLP) [14] classifier to be exact, as depicted in Figure 2. It is important to keep the initialization the same when comparing the VAEs to ensure for an even playing field.

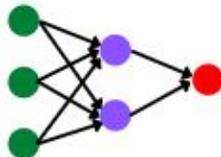


Figure 2: High level visualization of a MLP Classifier.

## 2.4 Prediction evaluation

It is worth noting that for the training and test sets are split per 70/30% for each of the classes for the predictions, thus containing 70% of each class to make sure they are all present during both the training and testing process. The VAE models are trained using the 70% split and once that is done, both the original training and test sets are encoded by the trained model in order to obtain the latent features. The latent space has the same number of samples, but the number of features have drastically changed from 5000 to the desired latent dimension, which will be set to 30 during these experiments. After having obtained the two datasets containing latent features, an MLP classifier will be fit using the new training set accompanied by the class labels for each sample. Once this model is trained, the new test set will be used to do predictions. These class predictions are compared to the actual classifications and an accuracy score is calculated accordingly. This fitting and predicting process will be performed ten times since the MLP classifier's initialization is somewhat random. Both the mean and standard deviation of these accuracies are taken to evaluate their goodness as well as consistency.

## 2.5 Learning process evaluation

The VAEs’ training process will also be analyzed with the usage of UMaps [15] to visualize the class distributions. The same 70/30% split will be used to create these plots. First, the VAE model is trained using the training set. Once the model is trained, both the training and test set will be plotted with their respective class labels. These plots will be indicate whether the VAE can provide a good latent representation for the predictions.

## 2.6 Repositories

The VAEs used during this research will not be built from the ground up. Instead, the Pytorch-VAE library will be used [16]. However, before being able to use the desired datasets as input, the library has to be altered in order to use this type of input data, since it is, in its original state, designed for images. In order to perform the transformation from images to RNA sequences, the Tybalt paper repository has been used [17]. The main advantage of using the Pytorch-VAE library is that the VAE models used for this research are already implemented. Furthermore, scikit-learn is used for the predictive models [18]. This library contains all models needed to do the predictions needed to benchmark the VAEs.

# 3 Results

This section does not yet contain all results as they are still in development. For all predictions using an MLP classifier network, learning rate is equal to 0.001. The number of neurons in the hidden layer differs, however. Since the NoVAE model has samples with 5000 features, the number of hidden neurons are set to 256 to mimic the encoding procedure in the VAEs.

## 3.1 Cancer stages

A table containing the accuracies per cancer type for the predicted cancer stages can be seen below in Table 5.

Table 5: Accuracy results as a percentage per VAE model denoted by the mean and its standard deviation.

	NoVAE	VanillaVAE	$\beta$ -VAE	$\beta$ -TCVAE	CatVAE
BRCA Mean	55.56	42.1	42.82	39.22	57.82
BRCA SD	2.27	5.31	4.24	3.79	3.21
LUAD Mean	53.76	36.71	38.21	38.21	48.73
LUAD SD	0.0	7.25	4.42	6.32	6.81
KIRC Mean	27.03	36.10	36.48	35.49	48.41
KIRC SD	8.85	2.77	3.51	6.05	2.53

The NoVAE prediction for the LUAD cancer type is the only one to always predict one class. This can be the result of one class containing most of the samples as seen in Table 6.

Table 6: Number of samples per cancer cancer stage for the LUAD cancer type.

Stage 1	93
Stage 2	41
Stage 3	30
Stage 4	9

Drawing any conclusions on the data provided in Table 5 is not trivial. The NoVAE model seems to perform best for the BRCA cancer type, but strange results occur for the LUAD cancer type as mentioned above. And for KIRC cancer type, it ranks amongst the worst accuracies. The VanillaVAE and the  $\beta$ -VAE have the most similar results, primarily varying in standard deviation.

### 3.2 Survival time

The prediction results for the survival time classes can be found in Table 7.

Table 7: Accuracy results per as a percentage VAE model denoted by the mean and its standard deviation.

	NoVAE	VanillaVAE	$\beta$ -VAE	$\beta$ -TCVAE	CatVAE
BRCA Mean	37.90	34.58	35.01	36.63	34.47
BRCA SD	0.73	2.23	1.90	1.51	1.84
LUAD Mean	36.41	32.81	33.22	36.02	30.41
LUAD SD	1.23	1.42	4.62	4.58	2.41
KIRC Mean	40.32	33.52	39.95	39.29	32.69
KIRC SD	1.42	2.49	2.53	2.11	2.30

Once again, the accuracies are the result of the model guessing. Survival time should be harder to predict than cancer stages or types, but the accuracies lie in the same ballpark due to the outcome being mostly random. The only notable observation would be the NoVAE’s scores. Since the accuracy mean lies just slight above the rest of the VAEs. Also the standard deviation seems to be lower, meaning the predictions are more consistent. This likely is the result of the VAE models negative impact on the predictive model’s training and testing data, which will be elaborated upon in section 3.4.

### 3.3 Cancer types

A overview of results for an MLP network is given in Table 8.

Table 8: Accuracy results per as a percentage VAE model denoted by the mean and its standard deviation.

	NoVAE	VanillaVAE	$\beta$ -VAE	$\beta$ -TCVAE	CatVAE
Mean	40.05	35.94	36.63	37.79	39.87
SD	7.85	1.45	2.07	3.15	2.20

The models do not seem to be actually learning. Since the accuracies mostly lie in the 35% to 40% range. The predictions, however, do not exist of only one class, but appear

to just be randomly guessing. When just using the original input however, a higher mean accuracy is found, but with much more fluctuating results. This can possibly be the result of the larger number of features that carry noise with them. Another interesting observation is the loss curve, since the NoVAE model converges after only a few epochs (Figure 3a) compared to the VanillaVAE for example (Figure 3b). The other VAE models have similar loss curves to the VanillaVAE, but the VanillaVAE is just used as an example for the comparison with NoVAE.

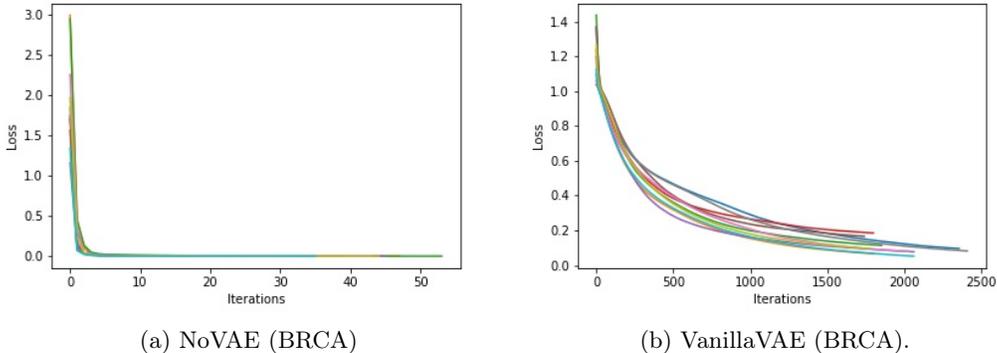


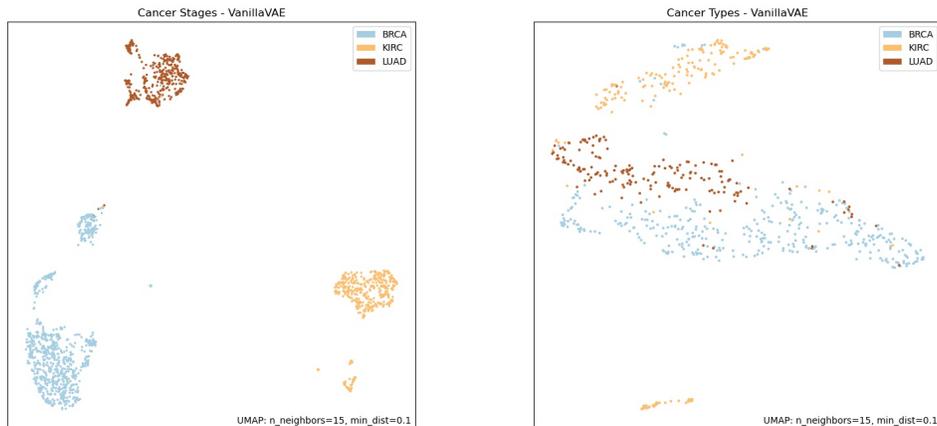
Figure 3: Loss curve comparison between VAE models. The colored lines represent the 10 times the model was run.

The difference in loss curves can be due to the number of features the MLP classifier is able to learn from, since NoVAE uses 5000 features instead of just 30 latent features like the other models. It is however remarkable that the model converges just after a couple epochs. The MLP classifier has also been run with a lower learning rate of 0.001 to slow down the learning process. The number of epochs until convergence increased but the accuracies stayed mostly the same, however.

### 3.4 VAE learning process

Since the results do not indicate a clear winner due to the models mostly guessing, a more in-depth look into learning process of the VAE is taken with the usage of UMaps [15], which are able to represent the latent representation as a 2D plot. To this end, the VanillaVAE is used with the cancer type dataset, since this VAE gives the best yields the most expressive results when it comes to UMaps. The cancer type dataset is chosen, because it should be the easiest for the VAE models to identify its classes.

Interestingly, the VAEs have the ability of distinguishing the different cancer types during training as seen in Figure 4a. The different cancer types are easily identifiable due to their clusters lying far apart. This figure is obtained by training the model using the same learning rate as before (0.005) and feeding the trained model the original dataset again. However, once the model is trained and represented with new data, it is unable to reproduce the results found with the training set as seen in Figure 4b.

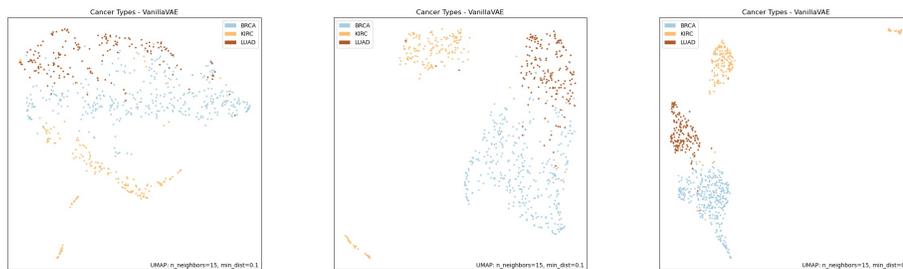


(a) VanillaVAE training set.

(b) VanillaVAE test set.

Figure 4: Two UMaps for cancer types learning process using the VanillaVAE with learning rate 0.005.

Since this test set is eventually used for predictions, it is understandable that these will mostly consist of guessing, since the latent representation does not contain a clear classification unlike the training set. Another interesting finding is the effect of the learning rate on this class segregation. A lower learning rate will lead to a higher class divisiveness as seen in Figure 5.



(a) Learning rate: 0.001

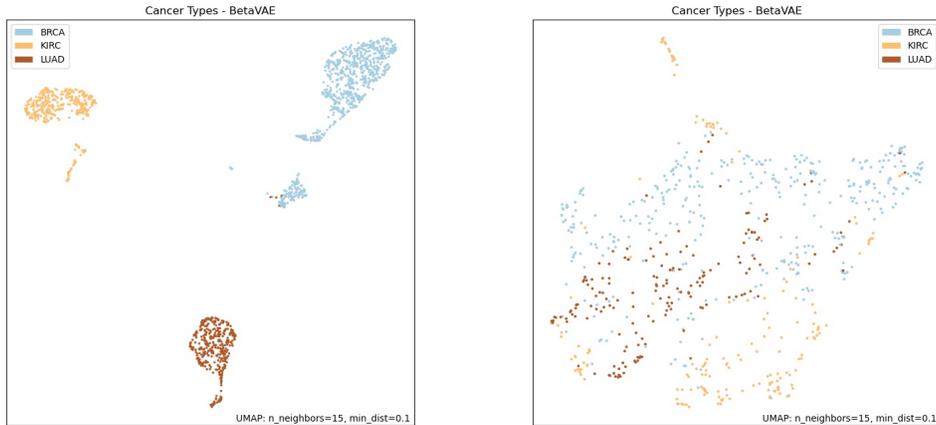
(b) Learning rate:  $10^{-4}$

(c) Learning rate:  $10^{-8}$

Figure 5: Three UMaps for the cancer types learning process using the VanillaVAE and the test set with different learning rates.

As seen in the figures above, the lower the learning rate, the better the latent representation in terms of class separation. Figure 5c has the best classification since it represents the original data most accurately because of its low learning rate. Although this latent space does seem to have better class divisiveness, the MLP model still mostly guesses as the results are mostly the same as reported in the previous sections.

Besides the learning rate, the  $\beta$  when using the  $\beta$ -VAE is also worthy of mentioning. Since setting this  $\beta$  term to a low value such as  $10^{-5}$  makes the VAE function more like a regular autoencoder. Such a low  $\beta$  term makes the KL divergence negligible, which is precisely what makes a VAE variational. When performing the same experiment as conducted for Figure 4, a similar result can be observed as seen in Figure 6.



(a)  $\beta$ -VAE training set.

(b)  $\beta$ -VAE test set.

Figure 6: Two UMaps for the cancer types learning process using the  $\beta$ -VAE with a learning rate of 0.005 and  $\beta$  term of  $10^{-5}$ .

This outcome questions the variational property of the VAEs used during this project since the same type of result can be recreated with a very low  $\beta$  term, essentially representing a regular autoencoder.

## 4 Responsible Research

### 4.1 Ethics

The ethical aspects of this research are mostly concentrated around the use of patient samples. Namely, their RNA sequences or genes and their additional clinical data. It is essential that this is and stays anonymized during the process of conducting experiments. Fortunately, this is already the case in the initial state of the datasets. Only containing The Cancer Genome Atlas ID per sample to be able to identify and match the samples, which is also used in this paper to match the clinical data to the respective RNA sequence.

### 4.2 Reproducibility

Since the VAE models are essentially built up of neural networks, there is some randomness present when constructing the latent space, as well as reconstruction the original data whilst decoding. This can cause issues when trying to reproduce the results. The same would apply

to the MLP classifier used for the predictions, it also possesses a random aspect. However, when running this models multiple times and taking their mean and standard deviation, anyone trying to find the same result will come fairly close to the original result. Which is indeed done to compare the VAE models with each other.

## 5 Discussion, Conclusion and Future Work

In this report, different VAE models have been compared in their ability to construct a useful latent representation for downstream task predictions. Knowing each of the VAE models' most optimal initialization would increase the accuracies of the predictions. This does, however, lie beyond the scope of this paper, since during this research we seek to compare the VAEs based on their most default parameters. This would otherwise lead to too many configurations since there are already three different cancer types and three different patient properties to do the predictions for.

Furthermore, the MLP predictions seem to mostly consist of guessing, with only the NoVAE model having a slightly higher accuracy mean and lower standard deviation between runs. This can be due to the dataset being erroneously constructed, but this is quite unlikely since they have been extensively tested to make sure that the patient samples match when extracting RNA sequence information based on the curated clinical dataset properties. At first, a support-vector machine (SVM) was also planned to be used in doing these predictions. But after seeing the MLP classifier's results, the SVM was left out. Instead, the guessing issue was further analyzed with the usage of UMaps to see whether or not the VAEs are actually able to produce a good latent representation. This is the case during training the process, but when confronted with new data after the model has been fit, the latent space does not accurately represent the different classes.

Moreover, different learning rates have been compared, since these do have an substantial effect on the class separations. The lower the learning rate, the better the models are able to differentiate between these classes. However, this will result in the latent space being more representative of the original data, rather than actually classifying the different cancer stages, survival times and cancer types.

Besides the learning rate, the  $\beta$ -VAE also yielded some interesting results. Since setting the  $\beta$  term to a low value will make the  $\beta$ -VAE function like a regular autoencoder. The differences between training and test for the Vanilla- and  $\beta$ -VAE are quite similar, meaning the variational property does not achieve much when constructing latent features.

Additionally, future work can dive deeper into the patient properties such as drug responses. However, different cancer types have to be chosen since the LUAD cancer type is the only one of the three used during this research that has a sufficient number of samples available. Regarding the prediction evaluations, the accuracy score can be replaced by an F1 score since it combines both accuracy and recall when computing its score. Most importantly, a deeper kook into the VAEs construction of latent spaces can be taken. It is evident that they are capable of differentiating between classes. But this is not the case when encountering new data. Seeing how regular autoencoders perform versus VAEs for downstream task analysis could also yield interesting results.

## References

- [1] “Cancer survival rates,” 05 2020. [Online]. Available: <https://www.nuffieldtrust.org.uk/resource/cancer-survival-rates#background>
- [2] P. Milanez-Almeida, A. J. Martins, R. N. Germain, and J. S. Tsang, “Cancer prognosis with shallow tumor RNA sequencing,” *Nature Medicine*, vol. 26, no. 2, pp. 188–192, Feb. 2020. [Online]. Available: <https://doi.org/10.1038/s41591-019-0729-3>
- [3] J. Rocca, “Understanding Variational Autoencoders (VAEs) - Towards Data Science,” 03 2021. [Online]. Available: <https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>
- [4] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2001037014000464>
- [5] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2014.
- [6] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework,” 2016.
- [7] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, “Isolating sources of disentanglement in variational autoencoders,” 2019.
- [8] D. Stutz, “Categorical Variational Auto-Encoders and the Gumbel Trick,” 07 2018. [Online]. Available: <https://davidstutz.de/categorical-variational-auto-encoders-and-the-gumbel-trick/>
- [9] “BRCA: The Breast Cancer Gene,” 10 2020. [Online]. Available: <https://www.nationalbreastcancer.org/what-is-brca>
- [10] D. Anusewicz, M. Orzechowska, and A. Bednarek, “Lung squamous cell carcinoma and lung adenocarcinoma differential gene expression regulation through pathways of Notch, Hedgehog, Wnt, and ErbB signalling,” *Scientific Reports*, vol. 10, no. 1, 2020.
- [11] L. Yin, W. Li, G. Wang, H. Shi, K. Wang, H. Yang, and B. Peng, “NR1B2 suppress kidney renal clear cell carcinoma (KIRC) progression by regulation of LATS 1/2-YAP signaling,” *Journal of Experimental & Clinical Cancer Research*, vol. 38, no. 1, 2019.
- [12] “Ucsc xena,” 12 2016. [Online]. Available: [https://xenabrowser.net/datapages/?dataset=EB%2B%2BAdjustPANCAN\\_IlluminaHiSeq\\_RNASeqV2.geneExp.xena&host=https%3A%2F%2Fpancanatlas.xenahubs.net&removeHub=https%3A%2F%2Fxena.treehouse.gi.ucsc.edu%3A443%7D](https://xenabrowser.net/datapages/?dataset=EB%2B%2BAdjustPANCAN_IlluminaHiSeq_RNASeqV2.geneExp.xena&host=https%3A%2F%2Fpancanatlas.xenahubs.net&removeHub=https%3A%2F%2Fxena.treehouse.gi.ucsc.edu%3A443%7D),
- [13] “Ucsc xena,” 12 2016. [Online]. Available: [https://xenabrowser.net/datapages/?dataset=Survival\\_SupplementalTable\\_S1\\_20171025\\_xena\\_sp&host=https%3A%2F%2Fpancanatlas.xenahubs.net&removeHub=https%3A%2F%2Fxena.treehouse.gi.ucsc.edu%3A443](https://xenabrowser.net/datapages/?dataset=Survival_SupplementalTable_S1_20171025_xena_sp&host=https%3A%2F%2Fpancanatlas.xenahubs.net&removeHub=https%3A%2F%2Fxena.treehouse.gi.ucsc.edu%3A443),
- [14] “1.17. neural network models (supervised)¶.” [Online]. Available: [https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html#multi-layer-perceptron](https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron)

- [15] Lmcinnes, “lmcinnes/umap.” [Online]. Available: <https://github.com/lmcinnes/umap>
- [16] A. Subramanian, “Pytorch-vae,” <https://github.com/AntixK/PyTorch-VAE>, 2020.
- [17] Greenelab, “greenelab/tybalt.” [Online]. Available: <https://github.com/greenelab/tybalt>
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.