Assessing the Capability of Multimodal Variational Auto-Encoders in Combining Information From Biological Layers in Cancer Cells

Bram Pronk , Marcel Reinders , Stavros Makrodimitris , Tamim Abdelaal , Mohammed Charrout , Mostafa elTager
Delft University of Technology
June 27, 2021

Abstract

Personalized treatment methods for a complex disease such as cancer benefit from using multiple data modalities from a patient's cancer cells. Multiple modalities allow for analysis of dependencies between complex biological processes and downstream tasks, such as drug response and/or expected survival rate. To this end, it is important to gain an understanding of the relationships between modalities in tumor cells. Multimodal Variational Auto-Encoders (MVAEs) are a combination of generative models trained on different sets of data modalities. In this research, the ability of MVAEs to capture common information between different data views from the same tumor cells is assessed. MVAE models discussed here are a Mixture-of-Experts (MoE) and a Product-of-Experts (PoE) approach to combining the generative model posterior distributions into a single common latent space. The performance assessment is done by: i) comparing the loss of information when reconstructing the training data to MOFA+, a linear method for combining multimodal data, and ii) measuring if one modality of a tumor cell can generate another modality, based on characteristics of the latent space learned by the MVAE. Biological data modalities considered are RNA-seq, gene-level copy number and DNA methylation (DNAme), gathered by The Cancer Genome Atlas. It is found that PoE reconstructs data from all data types with a higher accuracy compared to MoE and MOFA+. The mean squared error of PoE's average reconstruction loss is about a quarter of MOFA+'s, and less than a seventh of the MoE's average reconstruction loss. In terms of predicting modalities from other modalities, the PoE again outperforms MoE on all cross-modal predictions. Additionally, it can be concluded that both models have higher losses in their prediction of DNAme from other modalities, indicating a lesser correlation between this data type and the others.

Keywords — Multimodal Variational Auto-Encoder, The Cancer Genome Atlas, Deep Learning, Data integration

1 Introduction

Cancer is a common and dangerous disease with many different manifestations in the body. Most forms also have a wide array of treatment options, such as chemotherapy, radiation therapy and surgery [1]. While many people will deal with some form of cancer in their lives, treatment methods are often "effective in only a subset of the patient population" [2], due to the disease's heterogeneity. Searching a primary treatment method that is tailored to an individual patient is therefore critical, contributing to the growing field of precision treatment [3]. One of the foundations of precision treatment is assaying biological "omics" data from cells in individuals; measuring levels of proteins, genes or specific mutations. Observing across modalities in cancer cells can bring much insight into its development in the patient.

Retrieving omics data, correlating with clinical outcome and building a treatment plan from it can be a time-consuming and costly endeavour [4]. In order to alleviate this challenge, this research explores the modeling of multiple biological data layers from cancer cells by a deep learning model. This modeling produces a common latent space representation of these different data types. Based on the quality of this latent space, it can bring insights into underlying systematic relationships between the different data modalities, aiding correlation with clinical outcome. This understanding of cell dynamics could additionally help predict data modalities based on other modalities. The data gathering process could be less intensive, as data on fewer modalities needs to be retrieved.

One of the models proposed for the modeling of cancer cell data are Variational Auto-Encoders (VAE) [5]. VAE's are based on a general Auto-Encoder (AE) framework, consisting of an encoder-decoder pair. A standard AE is deterministic. It encodes data into a lower-dimensional latent space and then reconstructs it. The sole AE training objective is "to find the best encoder-decoder pair" [6] that minimizes the loss of the data compression. VAE's are in contrast stochastic, where input is not encoded as a single point, but rather as a distribution over the latent space. A VAE regularizes the training process, and instead of deterministic encoding uses variational inference to approximate a posterior of the encoder. These properties make VAE a generative model, able to encode high-dimensional data into a meaningful, smaller-dimensional latent space, "regular enough to be used for generative purpose" [6]. This process is illustrated in Figure 1.

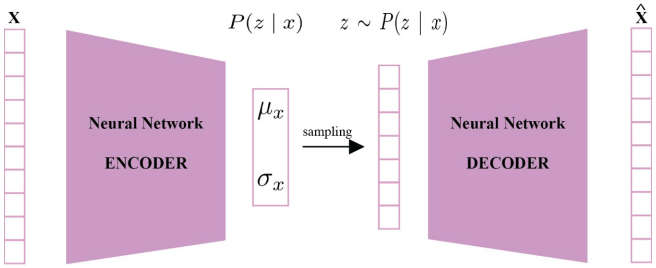

Figure 1: Schematic of a Variational Auto-Encoder. The neural network encoder takes input (x) and creates a distribution over the latent space $P(z \mid x)$. This distribution is then sampled to create the latent space representation of the input $z \sim P(z \mid x)$. Then this latent space is decoded into meaningful content.

Distinctive from typical VAE usage, this research does not solely want a representation for a single data modality, but aims to find a latent space that captures common information between multiple modalities. To that end, this research is focused on a VAE variant that can incorporate these different modalities, called Multimodal Variational Auto-Encoder (MVAE or MMVAE) [7] [8] [9]. A MVAE can be broadly defined as a combination of VAE models, each trained on different sets of data modalities, whereafter a probabilistic variational posterior distribution over the individual modalities is taken to form one joint posterior distribution. The generative properties of VAE's in this combination allow for the data generation of all input modalities from this singular posterior distribution. In this work, the Mixture-of-Experts (MoE) and Product-of-Experts (PoE) MVAE models are examined. Their names refer to the approach taken to combine the generative model posterior distributions into a single common latent space. The MoE MVAE model is based on the model introduced by Shi et al. [8], and the PoE MVAE model is based on the implementation by Wu and Goodman [7].

The dimensionality reduction of input data into a single common latent space, intrinsic to MVAE, can be beneficial for discovering (hidden) correlations or topics in data. The central dogma of molecular biology [10] describes the pathway of genetic information in cells, and proves there are relations between features in the cell data. For example, gene-level copy number and gene expression are expected to be strongly correlated, because if a gene is not present in a chromosome it can't produce RNA, and over representation of that gene can make it produce more RNA than normal. Therefore, if the latent space of the MVAE has good properties, there must be similar common features that appear in the generative data produced by the MVAE's.

Various studies have previously assessed the efficacy of VAE models in learning biological data. Research by Greene and Way has introduced a VAE model specifically trained on RNA-seq data used in this study, and found learned features of their model "were generally non-redundant and could disentangle large sources of variation in the data" [11]. Though this is a promising result, questions about a combination of these types of models for multi-omics data is left unanswered. That is in contrast to the MVAE model introduced by Zhang et al., where their OmiVAE model was also trained on two of the three modalities used in this work. Their model's accuracy was determined to be "better than existing methods" [12] in terms of tumour type classification. Compared to [12], the novelty of the research in this paper is found in determining accuracy of MVAE in predicting modalities based on other modalities. This is akin to the research by Minoura et al. [9], whose Mixture-of-Experts approach to a MVAE model is adjusted and re-purposed for this research. Their work uses transcriptome and surface protein measurements, whereas this work attempts to expand the cross-modal predictions to three modalities using RNA sequencing, gene level copy number and DNA methylation.

This research examines the capability of MVAE's to capture common information between different data views and then check if one data modality can generate other data modalities, based on the quality, continuity and uniformity of the latent space learned by the MVAE. This will be examined in twofold, by: i) comparing the loss of information when reconstructing the training data to Multi-Omics Factor Analysis v2 (MOFA+) [13], a linear method of combining multimodal data, and ii) measuring if one modality of a tumor cell can generate another modality, based on characteristics of the latent space learned by the MVAE. This means both the lossyness of the encoder's dimensionality reduction and the predictive abilities of MVAE models are assessed.

To answer the research question in a scientifically responsible manner, section Two discusses why this approach to measuring efficiency of MVAE's was chosen, and which tumor-cell modalities are selected and processed. Then section Three provides the requisite elaborations of the linear model MOFA+ and the two MVAE models, Mixture-of-Experts and Product-of-Experts. In section Four, the experimental setup and results are presented. A discussion of the results and scope-limitations or shortcomings are provided in section Five. Section Six will highlight methods undertaken to establish reproducibility, and a reflection is given on the ethical considerations of this work. The conclusion and recommendation of future work is given in section Seven.

2 Materials and Methods

2.1 Research Approach

Assessment of the MVAE models will be based on two factors: i) the mean squared error (MSE) when reconstructing data from the model, and ii) the MSE of cross-modal predictions compared to the omitted data.

Reconstruction loss is often considered when using dimensionality reduction algorithms. When reconstructing data encoded into a smaller-dimension, a perfect algorithm should return the exact data entered into the algorithm. The goal for a deep learning model such as a VAE is to capture underlying patterns or systems in the dimensionality reduction from high-dimensional input data to a smaller latent space, aiming that when data is reconstructed, it does so more accurate than a linear method. Therefore a good benchmark is comparing the reconstruction loss to such a linear method. For multimodal data, a state-of-the-art algorithm for linear dimension reduction is Multi-Omics Factor Analysis V2 (MOFA+) [13].

To measure predictive capabilities of a MVAE, a part of the original training data is omitted during training. Only after the model is trained, will one modality of this missing data be inserted into the model. The joint posterior distributive method of the MVAE allows this data to be used to generate other modalities. This predicted data can be compared against the other modalities from the omitted data during training, the actual measurements. Comparing two MVAE models on the MSE of this measured data with the cross-modal prediction, will provide a benchmark of the models, in addition to measuring predictive capabilities. Models used in this research, MOFA+ for the linear comparison, and Mixture-of-Experts and Product-of-Experts MVAE models for prediction, will be elaborated upon in section Three. To aid reproducibility of results, further details of the experimental setup will be stated in Section Four and exact model implementations are found in Appendix A.

2.2 Datasets

The biological information in this research is tumor cell data from The Cancer Genome Atlas (TCGA) pan-cancer multi-omics datasets [14]. Three types of high-dimensional omics data are selected; gene expression RNA sequencing (RNA-seq) [15], gene-level copy number variation (GCN) [16] estimated using the GISTIC2 [17] method, and DNA Methylation [18]. RNA-seq data is given by a $\log_2$-transform of normalized mRNA data. GCN data is expressed in the Gistic2 copy number, and "measured experimentally using whole genome microarray at Broad TCGA genome characterization center" [16]. Finally, DNA methylation is expressed in beta values, which are "continuous variables between 0 and 1, representing the ratio of the intensity of the methylated bead type to the combined locus intensity" [19]. All data is available publicly and exact sources are provided.

These datasets were selected for this research due to a belief in high common signal. By the dogma of molecular biology [10], these three datasets are each part of the general transfer of sequence information in cells, where DNAme is more upstream than RNA-seq and GCN, respectively. The three datasets have 8,440 samples in common. For analysis of the results, it was further required that for each sample the cancer type was known. This curated clinical data is also provided by the Pan-cancer Atlas [20]. All clinical data has been anonymized and is appropriately discussed in the Ethics section of this research. The cancer type was known for 8,418 samples of the common modality space, therefore 8,418 is the sample size in this research. This sample space includes samples from 33 different tumor types [21].

The input of the TCGA data in this research can be described similarly to that of Machiraju et al. in research with similar data [22]. We define input matrix $\mathbf{X}$ per data modality m in Equation 1, and X is visualized in Figure 2.

$$\mathbf{X}_{\{m\}} \in \mathbb{R}^{8418 \times 3000_m} \; \forall m \in \{\text{RNA-seq, GCN, DNAme}\} \tag{1}$$

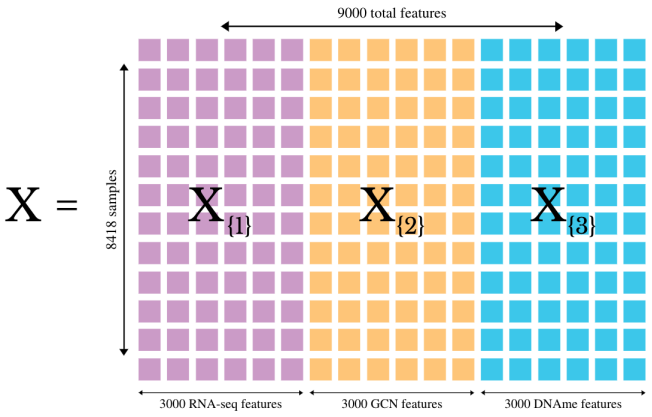

Figure 2: Visualization of multimodal dataset X in this research, composed of three individual datasets. Features are placed on the columns and denoted per datatype. In this research datasets are RNA-seq, Gene-level Copy Number and DNA Methylation.

2.3 Data Preprocessing

The 3,000 most variable features of each of the three modalities in TCGA's dataset were used for VAE training. Highest variability is defined by a feature's median absolute deviation (MAD). This was done since less variable features contribute less to the variability and distinction between cells. In prior research by Way and Greene, who ran VAE's on similar biological data, the 5,000 most variable features in the data were selected [11]. After experimentation in the early stages of this research, it was decided to use only the 3,000 most variable features for each dataset. This makes training the MVAE models smaller, easier, and ensures each modality is represented with similar importance and reduced noise.

It is important to note the original RNA-seq and GCN datasets contained negative values for some features in samples, while DNAme was presented as values between zero and one. For data consistency, all datasets were normalized to values between zero and one in preprocessing. Results presented in this paper were based on this normalized data. Besides consistency, normalization also prevents factorization or posterior distributions of input data being skewed by the larger or negative values from the RNA-seq and GCN data, in comparison to normalized DNAme values.

3 Analysis of Multimodal Data Integration Models

In order to critically assess the results and reproduciblity of this research, it is important to in-depth discuss the models used in multimodal data integration. In this section the methods are explained to obtain reconstruction loss and prediction losses, in addition to the internal working of the models. It is vital to note these models are based on existing literature, and software used is provided publicly by the original authors. Since those works do not consider this specific data input, or were not programmed to express reconstruction loss specifically, they had to be modified. These modifications are discussed in detail in the respective sections of the models, where applicable.

3.1 MOFA+: A Linear Method

Principal component analysis (PCA) is one of the most common linear dimensionality reduction algorithms. Though it lies at the basis of this method, it needs alterations order to suit data from multiple modalities. Multi-Omics Factor Analyis V2 (MOFA+) can in that sense be described as "a versatile and statistically rigorous generalization of principal component analysis to multi-omics data" [23]. MOFA+ infers a low-dimensional latent space from a high-dimensional set of data. This process is presented in Figure 3. Each modality is presented to the algorithm as a separate view (Y_1 ... Y_m in Figure 2). Each sample is then decomposed into ten factors and this low-dimensional representation is presented as the Z matrix in Figure 2. Ten factors were chosen since previous research on RNA-seq and DNA methylation showed 10 factors explained all the variance [13]. Then "for each factor, the weights (W) link the high-dimensional space with the low-dimensional manifold and provide a measure of feature importance" [13]. Further details are withheld here but are explained thoroughly in the original paper.

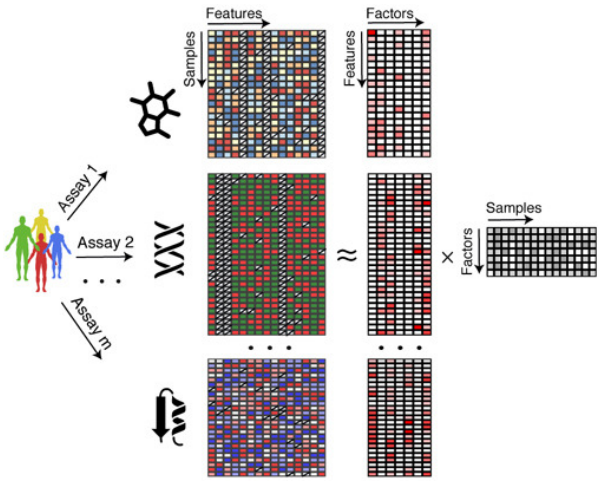

Figure 3: MOFA+ takes M data matrices as input $(Y_1, ..., Y_M)$, one from each data modality. The M matrices are decomposed into a matrix of factors (Z) for each sample and M feature weight matrices $(W_1, ..., W_M)$. Figure adapted from [24].[1]

This decomposition is the basis for the computation of the reconstruction loss, given by Equation 2:

$$\text{Reconstruction Loss} = \sum_{m=1}^{3} ||Y_m - \hat{Y}_m||^2 \tag{2}$$

where $\hat{Y}$ is given by Equation 3.

$$\hat{Y}_m = W_m \times Z \tag{3}$$

Equation 2 indicates the reconstruction loss is the total MSE of each modality. Per modality reconstruction losses will be presented in the Results section.

[1]Image licensed for use in any medium under Creative Commons (CC BY 4.0 license)

3.2 MVAE models

The MoE model in this work is adapted from scMM, a MoE MVAE introduced by Minoura et al. [9]. The PoE model in this work is adapted from the PoE MVAE introduced by Wu and Goodman [7]. All software is available publicly in the provided repositories [25] [26].

Both repositories were adapted to better fit this research project. Implementations of the unimodal VAE's which compromise the MVAE models are derived from the Vanilla-VAE model [5]. The implementation of this Vanilla-VAE model is derived from the PyTorch-VAE repository [27]. Both MoE and PoE originally had different VAE models, but this change allowed for comparison of performance of singular VAE's with the peer group. It was also established by the supervisors to use a singular hidden dimension in the VAE models, with a size of 256 (see Appendix A).

The Vanilla-VAE model implementation also redefined the training objective of the models. For both models, originally "the training objective is to maximize the marginal likelihood approximated by optimizing the evidence lower bound (ELBO) by stochastic gradient" [9]. Their definitions are given in the scMM paper [9] and PoE paper [7]. However, the training objective, or loss function, used by the Vanilla-VAE and therefore this paper, is defined by [5] [27] and shown in Equation 4.

$$\text{loss} = ||x - \hat{x}||^2 + KL\text{weight} \cdot KL[N(\mu_x, \sigma_x),\ N(0,1)] \tag{4}$$

So the loss is defined by a reconstruction term $||x - \hat{x}||^2$, the MSE of the model's input and output where x represents the input into the model and $\hat{x}$ is the outputted data (visualized in Figure 4). The KL term is the Kullback-Leibler divergence between the returned distribution and a standard Gaussian. The KL term is used to regularise the organisation of the latent space, an important role in the generative capacity of the VAE. For that purpose it is accounted for in the models' training objective (loss function). Elaborated upon in the original VAE paper [5]. To summarize, the objectives of the MVAE models were changed by the author to more resemble the Vanilla-VAE models. The next sections explain the differences between MoE and PoE, and a visualization of a MVAE is given in Figure 4.

Mixture-of-Experts MVAE Model

In the Mixture-of-Experts (MoE) MVAE model as proposed by Shi et al., the joint variational posterior is given as a combination of unimodal posteriors, using a Mixture-of-Experts approach. Further details on the mathematics are presented in the original paper [8]. The joint variational posterior is created as stated in Equation 5.

$$q_\phi(\mathbf{z} \mid x_{1:M}) = \sum_{m=1}^{M} \frac{1}{M} \cdot q_{\phi_m}(\mathbf{z} \mid x_m) \tag{5}$$

M in equation 5 is the number of modalities used in this research (3). $q_\phi(z \mid x_{1:M})$ denotes the variational joint posterior, or the latent space (z) under the condition of each input data's (x). The joint posterior is defined as a summation

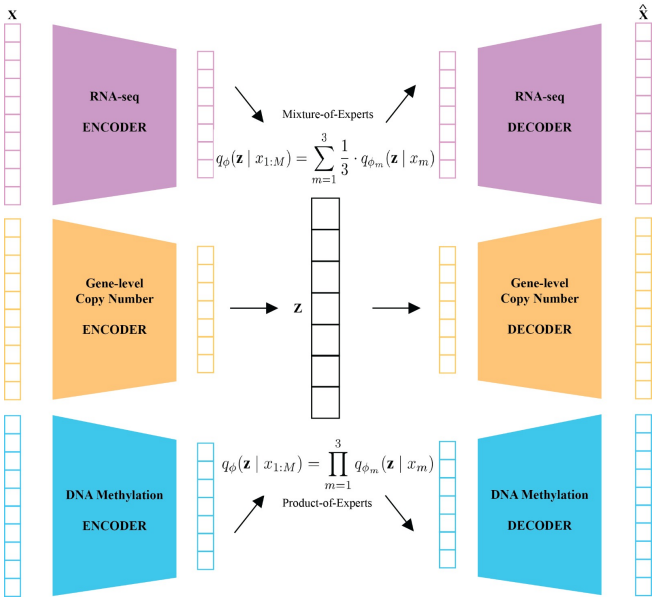

Figure 4: Schematic of general MVAE model. Input data (x) from each modality is entered into an encoder for that modality. Then for this research, either a Mixture-of-Experts *OR* Product-of-Experts approach is used to create a joint posterior distribution z. This joint distribution is used in the decoder of each modality to generate new data ($\hat{x}$).

of each export, scaled down by a factor $\frac{1}{M}$, where M is the number of modalities (3 in this case).

This repository was originally written for dualomics analysis, transciptome and surface protein data with chromatin accessibility data. This was expanded to the three different modalities in this paper. Feature vectors in the original paper are modelled by a negative binomial distribution for transciptome and surface protein data and with a zero-inflated negative binomial for chromatin accessibility data [9]. In this work, each modality is modelled by a normal distribution, akin to the Pytorch-VAE Vanilla-VAE model. With a normal distribution, the encoder can be trained to return the mean and the covariance matrix that describe the posterior distributions.

Product-of-Experts MVAE Model

For the Product-of-Experts (PoE) MVAE model, proposed by Wu and Goodman [7], the joint posterior is a product of individual posteriors. This approach is originally introduced by Hinton [28]. The idea is that each unimodal VAE in the model is considered an expert. In PoE, "each expert holds the power of veto—in the sense that the joint distribution will have low density for a given set of observations if just one of the marginal posteriors has low density", as explained by a contrasting section in the original MoE paper [8]. This implies that experts with high precision are weighing more heavily in determining the posterior distribution than lower density experts, since we take a product. The general formula from the paper [7] is given in Equation 6. A more in-depth look is also given in the original paper's supplement.

$$q_\phi(\mathbf{z} \mid x_{1:M}) = \prod_{m=1}^{M} q_{\phi_m}(\mathbf{z} \mid x_m) \tag{6}$$

$q_\phi(\mathbf{z} \mid x_{1:M})$ denotes the variational joint posterior, or the latent space (z) under the condition of each input data's (x). M is the number of modalities in this research. The equation makes clear the name of this approach, as each unimodal VAE's posterior is factorised to create a joint posterior.

MVAE Model Comparison

For clarity it is appropriate to contrast and compare both MVAE models. PoE suffers from potentially biased experts weighing in too heavily on the joint posterior. Since the joint distribution is formed from factorisation, a low density posterior distribution from one of the experts can make the joint distribution have low density. Besides this caveat, it was noted by Wu and Goodman that PoE requires a sub-sampled training paradigm [7], where a sub-sampling is required of the loss functions (ELBO in the original paper) for combined and individual modalities' to optimize each gradient step. In contrast to Equation 6, the MoE approach takes an equal vote among its experts, a summation of experts scaled down by a factor $\frac{1}{M}$. This spread of densities over all experts does not require the sub-sampled training paradigm from PoE. Furthermore, the original paper claims that "this characteristic makes them better-suited to latent factorisation, being sensitive to information across all the individual modalities" [8].

4 Results

4.1 Evaluation setup

Training data was splitted using a 70/10/20 ratio for the training set, validation set and prediction set respectively. This split was implemented to ensure enough samples can be used for prediction, whilst not hindering the training process of the models. This means that from a total of 8,418 samples, 5,892 were used to train the MVAE models, 841 to use as the validation set during training, and 1683 were omitted during training and only entered into the already trained model for measuring predictive capabilities. The implementation of the linear method MOFA+ does not require such a data split. This is an advantage of this method over the MVAE models. However, MOFA+ is not suited for prediction, while this research does omit 20% of the data for MVAE prediction. Therefore only 80% of samples were entered into the MOFA+ model. This ensures the comparison of models is fair, while keeping the advantage of not requiring a split of data into a training and validation set for MOFA+ intact. Exact models used, including hyperparameters and configurations are provided in Appendix A.

All three datasets are prepared in advance to have the same ordering of cancer cell samples. This ordering is randomized, so that batches of data contained a random selection of cancer types, preventing overfitting a model on a single type. Then on program execution, random indices are defined for each set used in execution. Features from each modality of the same samples were added to a training, validation or prediction set.

These indices are saved in files, so later downstream tasks are made aware of all samples entered into the model.

Reconstruction loss was calculated per epoch, and represent the reconstruction losses of the validation set, not the training data. Reconstruction loss results from the MVAE models presented here are the reconstruction losses of the final epoch. For MOFA+ the reconstruction loss is explained in the MOFA+ analysis chapter. Y axis of all shown results has been fixed for legibility.

4.2 Comparison of Data Reconstruction Losses

Reconstruction losses are shown in Figure 5. Results in the figure are grouped by modality per model. The losses answer the first part of the research question's assessment, giving a comparison in reconstruction performance between the MVAE models and the MOFA+ model. The most interesting outcome is the Mixture-of-Experts MVAE under performing to both the Product-of-Experts MVAE *and* the MOFA+ linear model across all modalities. This observed disparity could be a result of the model not being appropriate for this kind of data, conflicting with findings from the original paper [9]. Another explanation could be the different implementation of this model compared to the Product-of-Experts MVAE, a concern further discussed in the Discussion section. Section 4.5 and 4.6 will also highlight this disparity by modelling each model's latent space.

In addition to poor Mixture-of-Experts results, it can be seen that Product-of-Experts outperformed both models on average, and is sizably better than MOFA+ in reconstructing all three data types. A promising result, it tells on the ability of the PoE model to capture common data in the latent space. Also of note, all three models have a larger loss when reconstructing DNAme. This indicates lesser correlation from this modality compared to the others. Finally, GCN is relatively easier to reconstruct for MOFA+ and MoE, and harder for PoE.

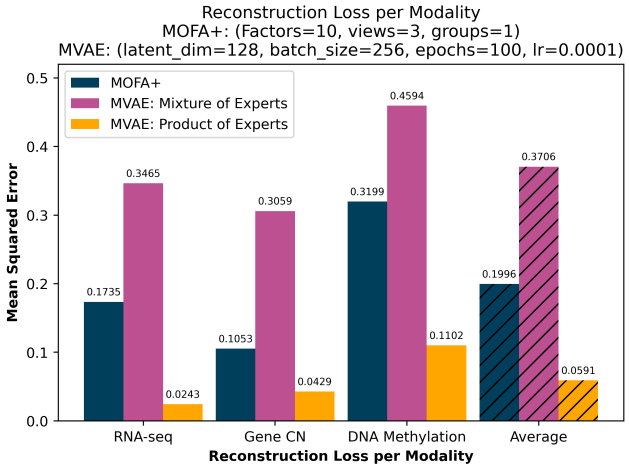

Figure 5: Mean squared error reconstruction losses of all data integration models. A higher mean squared error means a higher difference between input and output data, indicating unfavourable reconstructive capabilities.

4.3 MVAE Unimodal Predictions

In Figure 6 it is clearly shown how much the trained model has learned after 100 epochs. Omitted data during training is entered into each respective encoder of the trained model. In Figure 6, data is then decoded by the respective decoder. Data entered into both MVAE models is predicted with almost similar results to normal reconstruction losses presented in Figure 5. This suggests the validation split of the data in these models could be a good indicator of how well the model can reconstruct newly entered data in the future.

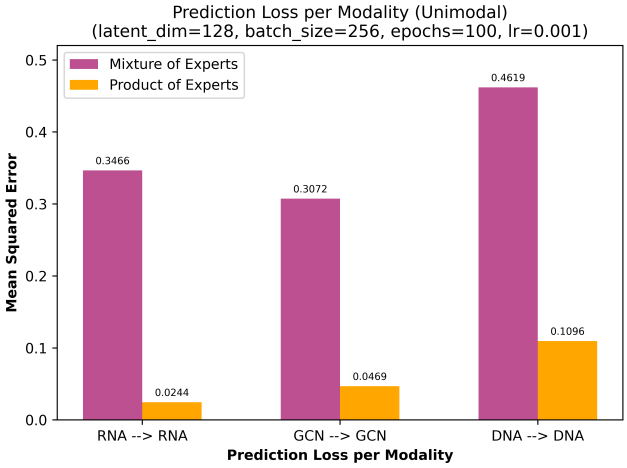

Figure 6: Mean squared error predictive loss of singular *uni-modal* predictions. The arrow $A \rightarrow A$ indicates omitted data from modality A was entered into the trained models' respective encoder, and was used to predict the equal modality A.

4.4 MVAE Cross-modal Predictions

Finally, Figure 7 shows answers to the cross-modal predictive capabilities of MVAE's as mentioned in the second part of the research question's assessment. In this figure, samples of only one of three modalities was entered into the respective encoder of the trained model, and the other two modalities were then reconstructed by their decoders based on the entered modality. These reconstructed modalities were then compared to the withheld sample data of those modalities using MSE. Interestingly enough, Product-of-Experts strongly outperforms Mixture-of-Experts here as well, with lower predictive MSE losses across all modalities. In addition to that, it is interesting to see both MVAE models have higher losses when predicting DNA methylation from other modalities. Both models are seemingly struggling more in learning features representative of DNA methylation when creating a common latent space. This was also reflected in the reconstruction losses for DNAme shown in Figure 5.

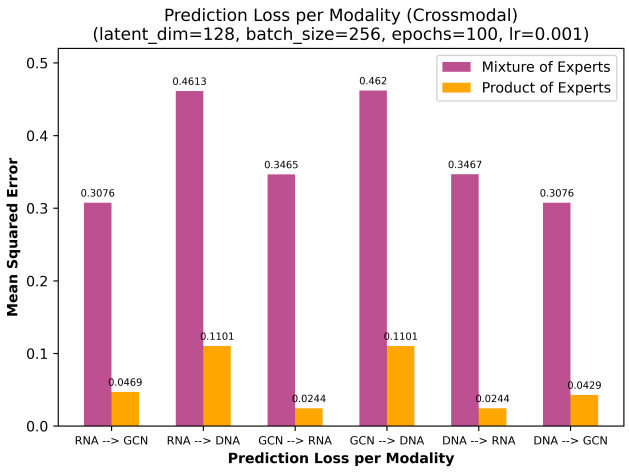

Figure 7: Mean squared error predictive loss of singular *cross-modal* predictions. The arrow $A \rightarrow B$ indicates omitted data from modality A was entered into the trained models' respective encoder, and was used to predict modality B.

4.5 Latent Space Models for Result Interpretation

To place the results in a broader context and to find reasoning behind the aforementioned results, the learning capabilities of the three models are modelled in this section. As mentioned, a common latent space representation of multiple data modalities can shed light on the underlying systematic relationships between them in cells. This means the latent space representation of the 33 different cancer types should be able to distinguish between the different cancer types. For this reason, the latent space of each data integration model ($\mathbf{Z}$) has been plotted using UMAP [29] visualization. UMAP is a dimensionality reduction algorithm that creates a 2-dimensional map (embedding) of high-dimensional data. To see if the model's latent space has learned something meaningful from the data, the UMAP point of every latent space representation of each sample has been coloured by the sample's cancer type. Results are shown in Figure 8.

Ideally, each model has learned to differentiate these cancer types, and can therefore better reconstruct or predict modalities from that cancer type. In the UMAP plot, clustering of each different cancer type informs on the quality of the local structure. Furthermore, similar categories tend to colocate in UMAP, revealing a global structure. Figure 8 clearly shows this clustering taking place in MOFA+ and PoE, while MoE's latent space is scattered seemingly at random. The UMAP of the MoE MVAE model indicates it is not learning properly, defining the bad performance relative to the others. Additionally, the cancer type clustering of the PoE model has more clusters and they also seem to be more tightly condensed than MOFA+. From this it is gathered that PoE distinguishes the different cancer types better than MOFA+, and this could be an indication to the lower reconstruction losses boasted by the PoE model when compared to MOFA+.

4.6 Regularisation Term Weighting for Improved Learning

Continuing the investigation into poor learning of the input data by the MoE model has led to examination of the Vanilla-VAE model as implemented in [27]. The MoE model is compromised of three Vanilla-VAE's, and thus learning capabilities of the Vanilla-VAE models reflect on the MoE's overall learning. Using UMAP visualization of the Vanilla-VAE's latent space using the same configurations, results showed no coherency in clustering of cancer types indicating no local or global structure. Results are shown in Appendix B.

When comparing to the UMAP of the MoE latent space in Figure 8, it does seem as if the MoE latent space is in fact showing really small clustering at the borders of its circular shape. This could indicate the KL term is weighing heavily in the model's training objective. The regularisation term is important for continuity of the latent space (close points in the latent space should give close results), and is therefore naturally impeding on the models' reconstruction accuracy.

In order to test the hypothesis of heavy regularization impeding on the model's learning capabilities, MoE and Vanilla-VAE models were retrained with a flat scalar applied to the KL term in the training objective. Also, for a clearer indi-

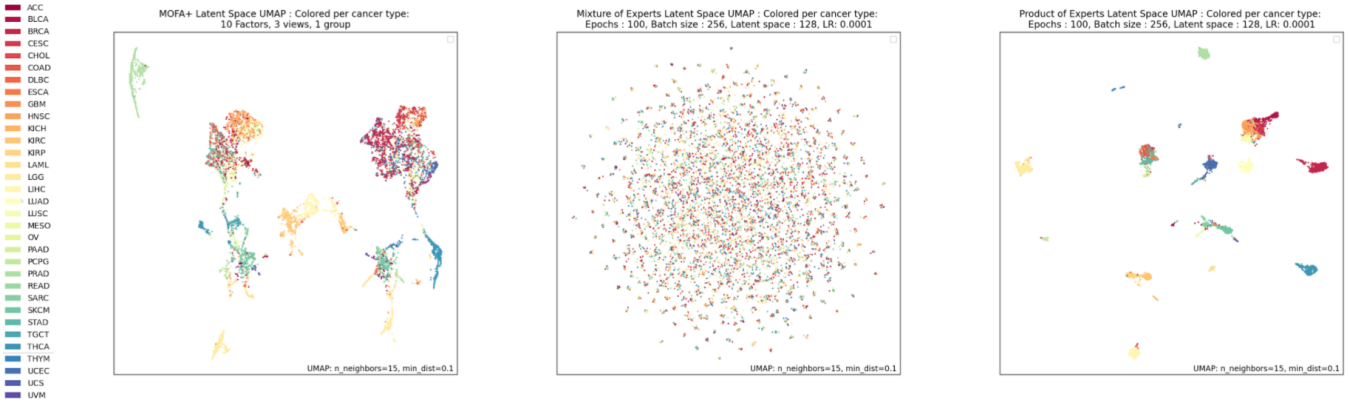

Figure 8: UMAP visualization of each model's latent space (z). Each sample's latent space representation is reduced to two dimensions using UMAP, and are then colored by the cancer type of the sample. Cancer type names in the figure are abbreviated and are found in the TCGA Study supplement [21].

cation of learning, the number of cancer types in the data were reduced to the three most prevalent types. Results are shown in Appendix C. In this configuration, it was found that for a scalar of 0.00001 and lower applied to the KL term, the Vanilla-VAE's latent space clearly distinguishes between cancer types. This scalar of 0.00001 was then also applied to the KL term in the MoE model. Unfortunately this did not lead to increased cancer type distinction in MoE.

5 Discussion

Taken together, the investigation in section 4.6 suggests that weighting down the KL term is the clear solution for fixing learning in Vanilla-VAE, but this should not be the conclusion drawn from this investigation. Pushing the KL term causes the model to learn a decoding of the data rather than a representation of data. This clashes with the goal of the VAE to predict or generate new data. Weighting the KL term also deviates from the definition of the Vanilla-VAE, and changes the model to a β-VAE [30]. In doing so the comparison between PoE or other models using Vanilla-VAE becomes faulty, and is something to consider for future work.

Models were provided by earlier literature [7] [9] [13]. The logic and implementation were mostly kept as intended by their original authors. This disclaimer is mentioned, since specifically the implementations of the MVAE models varied wildly. A note of caution is due since the performance of MoE was below expectations given the findings in the MoE model paper [9].

For instance, the Mixture-of-Experts model in this research performed a prior and posterior Laplace distribution on the data. Also, the manner in which a common latent space is obtained was implemented differently software-wise, besides of course being logically different. The author wants to acknowledge this research has merit on the basis of using these provided software libraries, but as discussed this approach has its shortcomings. Furthermore, choice of hyperparameters was based on the work of peers, who benchmark the influences of certain hyperparameters or initialization methods on the VAE networks. To rule out faulty hyperparameters configurations explaining poor MoE results, different values of the MoE model's learning rate and latent dimension were also considered. A small grid search of learning rate values $\{0.0001, 0.001, 0.01\}$, with latent dimensions of $\{128, 64, 32\}$ was performed. This grid search indicated results did not much differ between configurations.

Finally, the evidence presented by this research could also imply the 3,000 features were still much too noisy. Data with too much noise can skew the completeness of the latent space. Less variable features also contribute less to distinction between cells. A further study could assess if the data is in fact too noisy by reducing the number of features in the configuration.

6 Responsible Research

This section touches upon the reproducibility and ethical responsibility of this research, in order to verify the trustworthiness of the results and to build upon this research in the future. The scientific community should be able to critically asses and validate presented results. For that reason, measurements taken to increase the reproducibility will be listed in paragraph 6.1. Furthermore, the goal of this section is also to discuss the ethical implications of this research. Since the results of this research can potentially influence the decision-making process for personalized cancer treatment, these implications, shortcomings and disclaimers should be discussed thoroughly. This is discussed in paragraph 6.2.

6.1 Reproducibility in Data and AI

The Materials and Methods section and the experimental setup are described as accurate as possible, to accomplish open and reproducible science. In this section, this effort is highlighted by further discussing measures taken to make clear the experimental parameters.

A primary concern of machine learning algorithms is that they are highly sensitive to intrinsic factors such as; effect of random seeds or environment properties, and extrinsic factors; hyperparameters, codebases, resulting in a wide range of results [31]. Therefore it is paramount when using machine learning algorithms that all factors, intrinsic and extrinsic, are made public to be able to reproduce the results. To this end, the exact PyTorch [32] neural network module [33] configurations have been added for the Mixture-of-Experts and Product-of-Experts MVAE models in Appendix A. Added to these model descriptions are the exact input parameters and random seeds used to obtain results presented in this paper. Besides the deep learning MVAE models, Appendix A also contains model- and training options for MOFA+ used in this research.

The MVAE models were taken from the GitHub repositories provided by the MVAE models' research papers this research was based on [25] [26]. Any modifications are referenced and discussed accordingly in their respective sections in the Analysis. All software written and repurposed for this research is published on GitHub.[2] Any software dependencies or environment settings are also uploaded to the repositories in the form of an Anaconda Environment file.

The Materials and Methods section discusses the preprocessing done on this research's input data is discussed. To expunge any confusions on the preprocessing, a documented script is written that does all the preprocessing from raw source data to the processed data used in this research. This is included in this project's personal GitLab repository.[3]

6.2 Ethical Considerations

As mentioned in the Materials and Methods section of this paper, data used in this research is from the Pan-Cancer Atlas [14]. The results gathered here are thus in whole based upon data generated by The Cancer Genome Atlas (TCGA) Research Network. The datasets used here are released publicly by TCGA. The overarching National Cancer Institute has developed research policies "to address many ethical and logistical considerations associated with collecting, analyzing, and providing access to data from human tissue specimens" [34]. This research does not publish any form of the TCGA data itself, and references to the download links of the raw data

[2]Repository: https://github.com/brmprnk/bachelor-thesis

are made in the Materials and Methods section. It should be noted that the quality of this data was not verified as part of this research. The training data must be interpreted with caution, as it could be biased, or not representative of all patients with these cancer types. Therefore these findings cannot be extrapolated to all patients, a concern that is shared with previous research on the RNA-seq dataset by Way and Greene [11].

In addition to the training data, it is important to bear in mind that developed systems in this research were never tested in a clinical setting. For the results of this research to have any weight in the decision-making process of cancer treatment, there is a definite need for a monitored approach and a slow implementation in a real life setting. Furthermore, the discussion section tried to be as forthcoming as possible when discussing the limitations of this research.

7 Conclusions and Future Work

This project was undertaken to examine the capability of MVAEs to capture common information between different data views and then check if one data modality can generate other data modality, based on the quality, continuity and uniformity of the latent space learned by the MVAE. Quality of findings are measured by comparing reconstruction losses to MOFA+ [13], a linear data-integration method, and by comparing a Mixture-of-Experts (MoE) MVAE model [9] with a Product-of-Experts (PoE) MVAE model [7] on data losses when making predictions.

The first major finding was that PoE outperforms the other models on all fronts. Both MOFA+ and MoE had higher losses when reconstructing all modalities, and MoE had significantly higher losses when predicting other modalities. A promising result, it tells on the ability of the PoE model to capture common data in the latent space. This observed disparity between the MVAE models could be a result of the MoE model not being appropriate for this kind of data, conflicting with findings from the original paper [9]. Another explanation could be the different implementation of this model compared to the PoE MVAE. Investigation into the poor results further came up with UMAP visualizations of each models' latent space, and found MoE was not learning any meaningful representations of the 33 cancer types.

The second major finding was that reconstructing and predicting DNA methylation data was found to be more difficult for all models. This indicates lesser systematic relationships between DNAme data and RNA-seq and GCN data, possibly since DNAme is considered to be a more upstream process in cancer's life cycle. The empirical findings in this study provide arguments towards the assessment posed in the research goal, the lossyness of the encoder's dimensionality reduction is established together with the predictive abilities of MVAE models.

To explicitly mention the limitations of this study, it should be noted implementations of models were taken as is from provided software repositories, which could have implications on the way these models perform and the results presented. Furthermore, training data must be interpreted with caution, as it could be biased, or not representative of all patients with these cancer types. Therefore these findings cannot be extrapolated to all patients, a concern that is shared with previous research on the RNA-seq dataset by Way and Greene [11].

Greater efforts are needed to ensure any application of these models or incorporation of these results in a clinical setting. In future work, the MVAE models should be rewritten to the exact same environment besides their core logic. There should be no factors influencing the results other than the MoE or PoE factorisation of the joint distribution. As it stands, models implemented ran with equal hyperparameters, but inner logic differs in areas mentioned in the Discussion section. The UMAP modelling of the latent space per cancer type showed the MoE implementation requires substantially more tweaking in order to make it learn a useful data representation. Finally, it is important to bear in mind that developed systems as implemented in this research were never tested in a clinical setting. Thus for the results of this research to have any weight in the decision-making process of cancer treatment, further research has a definite need for a monitored approach, possibly involving slow implementation of these models in a real life setting.

References

[1] National Cancer Institute, "Types of cancer treatment," https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6352312/#R2, 2017.

[2] P. Krzyszczyk, A. Acevedo, E. J. Davidoff, L. M. Timmins, I. Marrero-Berrios, M. Patel, C. White, C. Lowe, J. J. Sherba, C. Hartmanshenn, K. M. O'Neill, M. L. Balter, Z. R. Fritz, I. P. Androulakis, R. S. Schloss, and M. L. Yarmush, "The growing role of precision and personalized medicine for cancer treatment," *TECHNOLOGY*, vol. 06, no. 03n04, pp. 79–100, Sep. 2018. [Online]. Available: https://doi.org/10.1142/s2339547818300020

[3] S. C. Williams, "News feature: Capturing cancer's complexity," *Proceedings of the National Academy of Sciences*, vol. 112, no. 15, pp. 4509–4511, 2015.

[4] M. Verma, "Personalized medicine and cancer," *Journal of Personalized Medicine*, vol. 2, no. 1, pp. 1–14, 2012. [Online]. Available: https://www.mdpi.com/2075-4426/2/1/1

[5] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *2nd International Conference on Learning Representations*, 2013.

[6] J. Rocca and B. Rocca, "Understanding variational autoencoders (vaes)," 2019. [Online]. Available: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[7] M. Wu and N. D. Goodman, "Multimodal generative models for scalable weakly-supervised learning," *CoRR*, vol. abs/1802.05335, 2018. [Online]. Available: http://arxiv.org/abs/1802.05335

[8] Y. Shi, N. Siddharth, B. Paige, and P. H. S. Torr, "Variational mixture-of-experts autoencoders for multi-modal deep generative models," 2019.

[9] K. Minoura, K. Abe, H. Nam, H. Nishikawa, and T. Shimamura, "Scmm: Mixture-of-experts multimodal deep generative model for single-cell multiomics data analysis," *bioRxiv*, 2021. [Online]. Available: https://www.biorxiv.org/content/early/2021/02/19/2021.02.18.431907

[10] F. Crick, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970.

[11] G. P. Way and C. S. Greene, "Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders," *Pacific Symposium on Biocomputing. Pacific Symposium Biocomputing, 23, 80–91*, vol. 23, p. 80–91, 2017.

[12] X. Zhang, J. Zhang, K. Sun, X. Yang, C. Dai, and Y. Guo, "Integrated multi-omics analysis using variational autoencoders: Application to pan-cancer classification," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp. 765–769.

[13] R. Argelaguet, D. Arnol, D. Bredikhin, Y. Deloro, B. Velten, J. C. Marioni, and O. Stegle, "MOFA+ : a statistical framework for comprehensive integration of multi-modal single-cell data," *Genome Biology*, vol. 21, no. 111, May 2020.

[14] K. Chang, C. J. Creighton, C. Davis, L. Donehower, J. Drummond, D. Wheeler *et al.*, "The cancer genome atlas pan-cancer analysis project," *Nature Genetics*, vol. 45, no. 10, pp. 1113–1120, Oct 2013. [Online]. Available: https://doi.org/10.1038/ng.2764

[15] The Cancer Genome Atlas, "Pan-cancer atlas dataset: gene expression rnaseq - batch effects normalized mrna data," 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, Dec 2016, accessed: 20-04-2021.

[16] The Cancer Genome Atlas, "Pan-cancer atlas dataset: copy number (gene-level) - gene-level copy number (gistic2)," https://xenabrowser.net/datapages/?dataset=TCGA.PANCAN.sampleMap%2FGistic2_CopyNumber_Gistic2_all_data_by_genes&host=https%3A%2F%2Ftcga.xenahubs.net&removeHub=https%3A%2F%2Fxena.treehouse.gi.ucsc.edu%3A443, Aug 2016, accessed: 20-04-2021.

[17] C. H. Mermel, S. E. Schumacher, B. Hill, M. L. Meyerson, R. Beroukhim, and G. Getz, "Gistic2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers," *Genome Biology*, vol. 12, no. 4, Apr 2011.

[18] The Cancer Genome Atlas, "Pan-cancer atlas dataset: Dna methylation - dna methylation (methylation450k)," https://xenabrowser.net/datapages/?dataset=jhu-usc.edu_PANCAN_HumanMethylation450.betaValue_whitelisted.tsv.synapse_download_5096262.xena&host=https%3A%2F%2Fpancanatlas.xenahubs.net&removeHub=https%3A%2F%2Fxena.treehouse.gi.ucsc.edu%3A443, Dec 2016, accessed: 20-04-2021.

[19] D. J. Weisenberger, D. Van Den Berg, F. Pan, B. P. Berman, and P. W. Laird, "Comprehensive dna methylation analysis on the illumina® infinium® assay platform," *Application Note: Epigenetic Analysis*, 2010.

[20] The Cancer Genome Atlas, "Pan-cancer atlas dataset: Phenotype - curated clinical data," 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, Sep 2018, accessed: 08-06-2021.

[21] N. G. D. Commons, "TCGA study abbreviations," gdc.cancer.gov/resources-tcga-users/tcga-code-tables/tcga-study-abbreviations, accessed: 14-06-2021.

[22] D. A. G. Machiraju and E. Ashley, "Multi-omics factorization illustrates the added value of deep learning approaches," Stanford University, 2019.

[23] R. Argelaguet, D. Arnol, D. Bredikhin, Y. Deloro, B. van Velten, J. C. Marioni, and O. Stegle, "MOFA+ [Source code]," https://biofam.github.io/MOFA2/, Mar 2020, accessed: 23-04-2021.

[24] R. Argelaguet, B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buettner, W. Huber, and O. Stegle, "Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets," *Molecular Systems Biology*, vol. 14, no. 6, p. e8124, 2018. [Online]. Available: https://www.embopress.org/doi/abs/10.15252/msb.20178124

[25] Y. Shi, "Multimodal mixture-of-experts vae [Source code]," 2020. [Online]. Available: https://github.com/iffsid/mmvae

[26] M. Wu, "Multimodal variational autoencoder [Source code]," 2018. [Online]. Available: https://github.com/mhw32/multimodal-vae-public

[27] A. Subramanian, "PyTorch-VAE [Source code]," https://github.com/AntixK/PyTorch-VAE, 2020.

[28] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002. [Online]. Available: https://doi.org/10.1162/089976602760128018

[29] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," Sep 2018.

[30] 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," in *ICLR*, 2016.

[31] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *CoRR*, vol. abs/1709.06560, 2017. [Online]. Available: http://arxiv.org/abs/1709.06560

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, Chanan *et al.*, "Pytorch: An imperative style, high-performance deep learning library," http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf, pp. 8024–8035, 2019.

[33] Torch Contributors, "Pytorch neural network module class documentation," https://pytorch.org/docs/stable/generated/torch.nn.Module.html, 2019, accessed: 03-06-2021.

[34] The Cancer Genome Atlas, "The cancer genome atlas - Ethics and Policies," Mar 2019. [Online]. Available: https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga/history/policies

A MVAE Models

A.1 Mixture of Experts Model

```
Input shape Training Data : 5892, 3000 || Input shape Validation Data : 841, 3000 || Input shape Prediction Data : 1683, 3000
Input args : latent_dim=128, batch_size=256, epochs=100, learn_prior=False, llik_scaling=1.0, lr=0.0001, model='rna_gcn_dna',
num_hidden_layers=1, p_dim=3000, p_hidden_dim=256, r_dim=3000, r_hidden_dim=256, seed=1)

RNA_GCN_DNA(
  (vaes): ModuleList(
    (0): RNA(
      (enc): Enc(
        (enc): Sequential(
          (0): Sequential(
            (0): Linear(in_features=3000, out_features=256, bias=True)
            (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
          )
        )
        (fc_mu): Linear(in_features=256, out_features=128, bias=True)
        (fc_var): Linear(in_features=256, out_features=128, bias=True)
      )
      (dec): Dec(
        (dec): Sequential(
          (0): Sequential(
            (0): Linear(in_features=128, out_features=256, bias=True)
            (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
          )
        )
        (fc31): Linear(in_features=256, out_features=3000, bias=True)
        (final_layer): Sequential(
          (0): Linear(in_features=256, out_features=3000, bias=True)
          (1): Sigmoid()
        )
      )
      (_pz_params): ParameterList(
          (0): Parameter containing: [torch.FloatTensor of size 1x128]
          (1): Parameter containing: [torch.FloatTensor of size 1x128]
      )
    )
    (1): GCN(
      (enc): Enc(
        (enc): Sequential(
          (0): Sequential(
            (0): Linear(in_features=3000, out_features=256, bias=True)
            (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
          )
        )
        (fc_mu): Linear(in_features=256, out_features=128, bias=True)
        (fc_var): Linear(in_features=256, out_features=128, bias=True)
      )
      (dec): Dec(
        (dec): Sequential(
          (0): Sequential(
            (0): Linear(in_features=128, out_features=256, bias=True)
            (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
          )
        )
        (fc31): Linear(in_features=256, out_features=3000, bias=True)
        (final_layer): Sequential(
          (0): Linear(in_features=256, out_features=3000, bias=True)
          (1): Sigmoid()
        )
      )
      (_pz_params): ParameterList(
          (0): Parameter containing: [torch.FloatTensor of size 1x128]
          (1): Parameter containing: [torch.FloatTensor of size 1x128]
      )
    )
```

```
    (2): DNA(
      (enc): Enc(
        (enc): Sequential(
          (0): Sequential(
            (0): Linear(in_features=3000, out_features=256, bias=True)
            (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
          )
        )
        (fc_mu): Linear(in_features=256, out_features=128, bias=True)
        (fc_var): Linear(in_features=256, out_features=128, bias=True)
      )
      (dec): Dec(
        (dec): Sequential(
          (0): Sequential(
            (0): Linear(in_features=128, out_features=256, bias=True)
            (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (2): ReLU()
          )
        )
        (fc31): Linear(in_features=256, out_features=3000, bias=True)
        (final_layer): Sequential(
          (0): Linear(in_features=256, out_features=3000, bias=True)
          (1): Sigmoid()
        )
      )
      (_pz_params): ParameterList(
          (0): Parameter containing: [torch.FloatTensor of size 1x128]
          (1): Parameter containing: [torch.FloatTensor of size 1x128]
      )
    )
  )
  (_pz_params): ParameterList(
      (0): Parameter containing: [torch.FloatTensor of size 1x128]
      (1): Parameter containing: [torch.FloatTensor of size 1x128]
  )
)
```

A.2 Product of Experts Model

```
Input shape Training Data : 5892, 3000 || Input shape Validation Data : 841, 3000 || Input shape Prediction Data : 1683, 3000
Input args : (n_latents=128, batch_size=256, epochs=100, annealing_epochs=2, lr=0.0001, log_interval=10, cuda=False, seed=1,
num_hidden_layers=1, hidden_layer_dim=256)
MVAE(
  (rna_encoder): Encoder(
    (encoder): Sequential(
      (0): Sequential(
        (0): Linear(in_features=3000, out_features=256, bias=True)
        (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
    )
    (fc_mu): Linear(in_features=256, out_features=128, bias=True)
    (fc_var): Linear(in_features=256, out_features=128, bias=True)
  )
  (gcn_encoder): Encoder(
    (encoder): Sequential(
      (0): Sequential(
        (0): Linear(in_features=3000, out_features=256, bias=True)
        (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
    )
    (fc_mu): Linear(in_features=256, out_features=128, bias=True)
    (fc_var): Linear(in_features=256, out_features=128, bias=True)
  )
  (dna_encoder): Encoder(
    (encoder): Sequential(
      (0): Sequential(
        (0): Linear(in_features=3000, out_features=256, bias=True)
        (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
    )
    (fc_mu): Linear(in_features=256, out_features=128, bias=True)
    (fc_var): Linear(in_features=256, out_features=128, bias=True)
  )
  (rna_decoder): Decoder(
    (decoder): Sequential(
      (0): Linear(in_features=128, out_features=256, bias=True)
      (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
    )
    (final_layer): Sequential(
      (0): Linear(in_features=256, out_features=3000, bias=True)
      (1): Sigmoid()
    )
  )
  (gcn_decoder): Decoder(
    (decoder): Sequential(
      (0): Linear(in_features=128, out_features=256, bias=True)
      (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
    )
    (final_layer): Sequential(
      (0): Linear(in_features=256, out_features=3000, bias=True)
      (1): Sigmoid()
    )
  )
  (dna_decoder): Decoder(
    (decoder): Sequential(
      (0): Linear(in_features=128, out_features=256, bias=True)
      (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
    )
    (final_layer): Sequential(
      (0): Linear(in_features=256, out_features=3000, bias=True)
      (1): Sigmoid()
    )
  )
  (experts): ProductOfExperts()
)
```

A.3 MOFA+ Model Settings

```
Following Code is repurposed from the Jupyter Notebook found in the "Training a model in Python" section
of the MOFA+ tutorials: https://biofam.github.io/MOFA2/tutorials.html

## (4) Set model options ##
# - factors: number of factors. Default is K=10
# - likelihoods: likelihoods per view (options are "gaussian","poisson","bernoulli").
#                 Default is None, and they are infered automatically
# - spikeslab_weights: use spike-slab sparsity prior in the weights? (recommended TRUE)
# - ard_factors: use ARD prior in the factors? (TRUE if using multiple groups)
# - ard_weights: use ARD prior in the weights? (TRUE if using multiple views)

# Model settings used:
# Simple (using default values)
ent.set_model_options()

# Advanced (using personalised values)
ent.set_model_options(
        factors = NUM_FACTORS,
        spikeslab_weights = False,
        ard_factors = False,
        ard_weights = True
)

## (5) Set training options ##
# - iter: number of iterations
# - convergence_mode: "fast", "medium", "slow".
#                For exploration, the fast mode is good enough.
# - startELBO: initial iteration to compute the ELBO (the objective function used to assess convergence)
# - freqELBO: frequency of computations of the ELBO (the objective function used to assess convergence)
# - dropR2: minimum variance explained criteria to drop factors while training.
#                 Default is None, inactive factors are not dropped during training
# - gpu_mode: use GPU mode? this needs cupy installed and a functional GPU, see https://cupy.chainer.org/
# - verbose: verbose mode?
# - seed: random seed

# Model training settings used:
# Simple (using default values)
ent.set_train_options()

# Advanced (using personalised values)
ent.set_train_options(
        iter = 100,
        convergence_mode = "medium",
        startELBO = 1,
        freqELBO = 1,
        dropR2 = None,
        gpu_mode = False,
        verbose = True,
        seed = 1
)
```

B UMAP Plots of Vanilla-VAE Latent Space

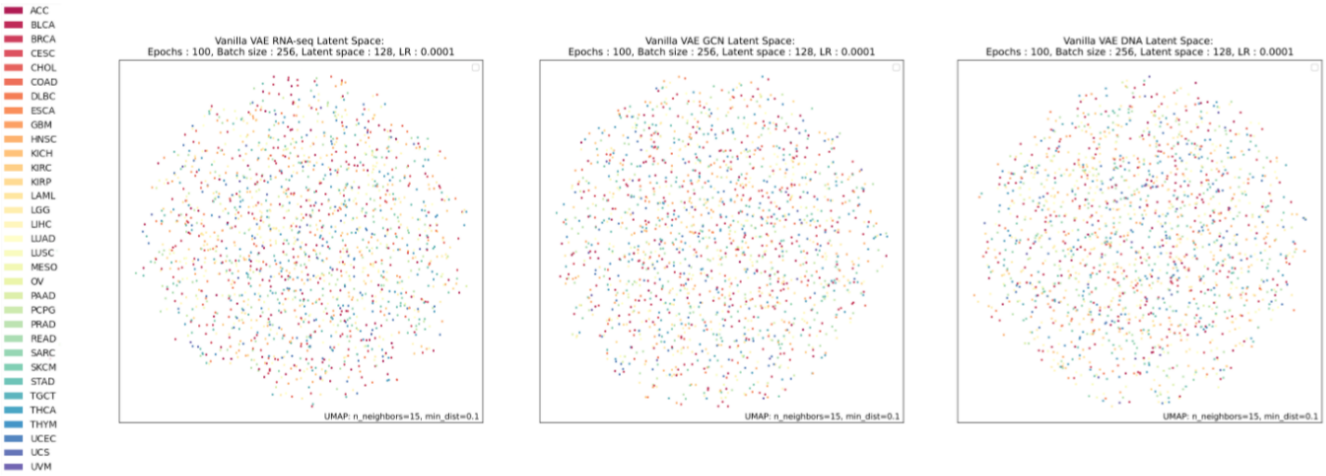

Figure 9: UMAP visuxalization of Vanilla-VAE's latent space (z), for each data type. Each sample's latent space representation is reduced to two dimensions using UMAP, and are then colored by the cancer type of the sample. Cancer type names in the figure are abbreviated and are found in the TCGA Study supplement [21]. Figure indicates Vanilla-VAE model is not learning a representation of the 33 cancer types present in the data.

C Vanilla-VAE KL Weighting

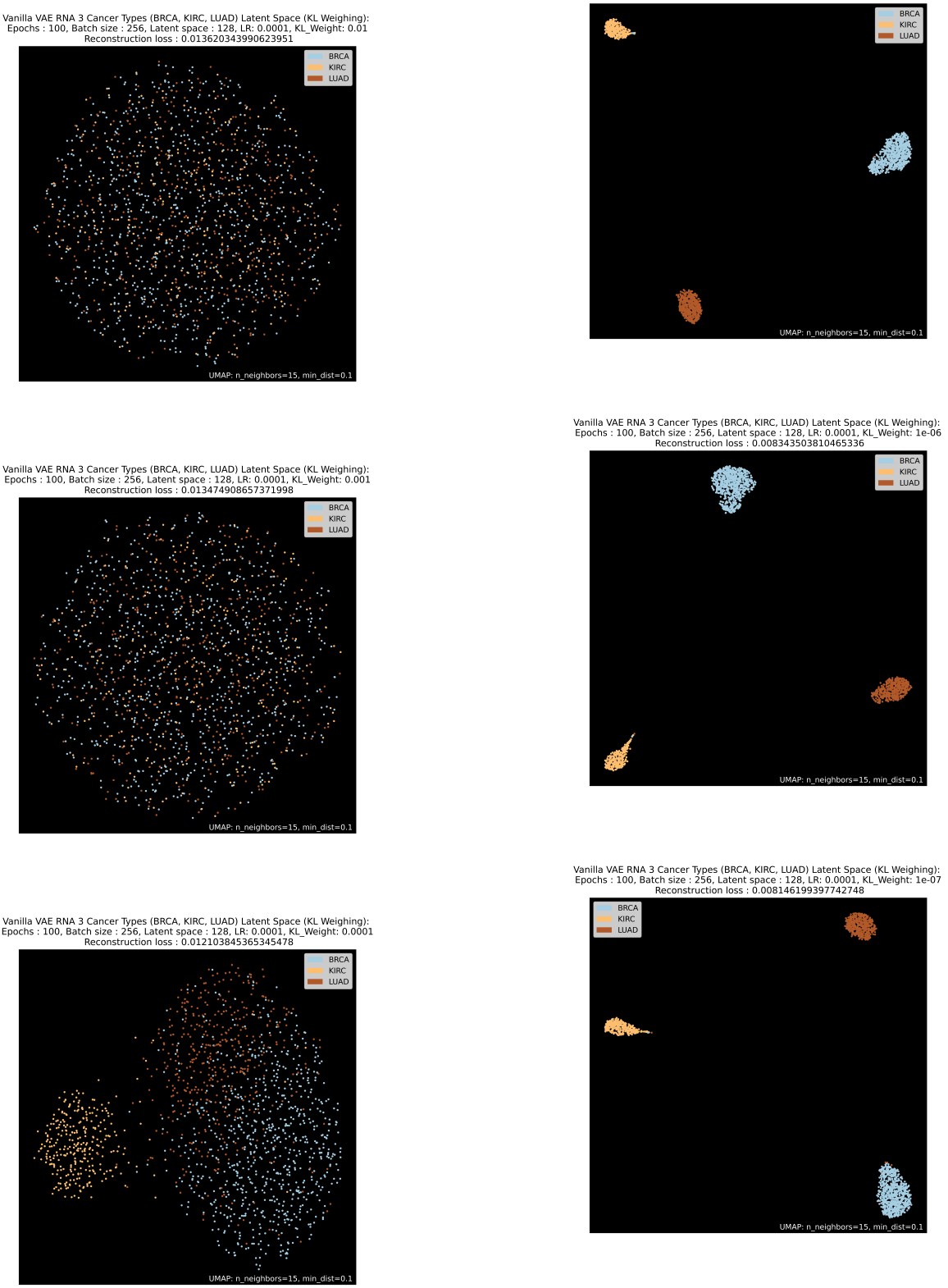

Figure 10: UMAP of Vanilla-VAE latent space (z). Model gradually increases its capability of differentiating three cancer types as the KL term in the loss function is scaled down.