PCVAE: a Controlled deep Variational Autoencoder for Pancancer gene expressions clustering analysis

by

Yuxiang Fu

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Honours Bachelor of Science, Computer Science

in

THE FACULTY OF SCIENCE
(Department of Computer Science)

The University of British Columbia
(Vancouver)

April 2023

The following individuals certify that they have read, and recommend to the Faculty of Science for acceptance, the thesis entitled:

PCVAE: a Controlled deep Variational Autoencoder for Pancancer gene expressions clustering analysis

submitted by **Yuxiang Fu** in partial fulfillment of the requirements for the degree of **Honours Bachelor of Science, Computer Science** in **Department of Computer Science**.

Examining Committee:

Andrew Roth, Professor, Computer Science, Pathology and Laboratory Medicine, UBC
Supervisor

Abstract

The quest for longevity and healthy life is an essential human need. Cancer intercepts this need with great interference. Cancer exhibits high heterogeneity in both space and time. Its aggressive proliferation poses a challenge in searching for a cure for cancer as it is notoriously difficult to address recurrence, resistance and metastasis. This highlights the urgent demand to understand cancer better via developing a systematic classification framework.

Cancer subtyping and clustering methods have been studied frequently, while no method we know of unveils clusters in the Pancancer dataset by alleviating the tissue effect. The autoencoder architecture and its variants have the ability to retain tissue effect in the low-dimensional latent space. In this thesis, we present PCVAE, a modification of the variational autoencoder that controls the primary tissue effect of the bulk RNA sequencing data. PCVAE offers the ability to uncover novel connections across heterogeneous cancers in a site-effect-free environment. These connections can be expressed by identifying new clusters, which encapsulate abundant biological meaning. PCVAE demonstrates its proficiency in removing the tissue signal and additionally allows for clustering on less dominant features. One of the PCVAE models generates high-quality clusters compared to others under multiple measures. The survival analysis is performed on the novel cohorts of patients to provide valuable insights on the possible interpretations in terms of molecular oncology.

Mathematical Notation

Symbol	Meaning	Dimension
$\mathcal{M}$	a family of models	
$\mathcal{D}$	any generic dataset	
X	input, dataset	$\mathbb{R}^{m \times n}$
$\mathcal{X}$	sample space in probability theory	
x_i	i-th example in the data	$\mathbb{R}^n$
x	arbitrary example from the dataset	$\mathbb{R}^n$
z	example decoded in the latent space	$\mathbb{R}^k$
y	output, label	
θ, ϕ, ψ	an array of learnable parameters	
$\mathbf{\Sigma}$	a covariance matrix	$\mathbb{R}^{c \times c}$
$e_\theta(\cdot)$	an encoder network with a set of parameters θ in the autoencoder	
$d_\theta(\cdot)$	a decoder network with a set of parameters θ in the autoencoder	
$p(x)$	evidence probability	$[0, 1]$
$p(z)$	prior of the latent variable	
$p(z \mid x)$	posterior	
$p(x \mid z)$	likelihood	
$\mathcal{N}(\mu, \sigma^2)$	a normal distribution with mean μ and variance σ^2	
$D_{KL}(p \mid\mid q)$	the Kullback–Leibler divergence between two probability distributions $p(\cdot), q(\cdot)$	
$\mathcal{N}_c(\boldsymbol{\mu}, \mathbf{\Sigma})$	the multivariate normal of a c dimensional random vector	$\boldsymbol{\mu} \in \mathbb{R}^c$
$\mathbb{E}_p(X)$	the expectation of a random variable $X \sim p(\cdot)$	
∇_x	taking the gradient with respect to variable x	
$tr(A)$	the trace of matrix A	

Table of Contents

List of Tables

List of Figures

Glossary

PCA Principle Component Analysis

UMAP Uniform Manifold Approximation and Projection for Dimension Reduction

T-SNE t-Distributed Stochastic Neighbor Embedding

ICGC The International Cancer Genome Consortium

PCAWG The Pancancer Analysis of Whole Genome

CNV Copy Number Variation

SNV Single Nucleotide Variant

VAE Variational Autoencoder

ELBO Evidence Lower Bound

KL DIVERGENCE Kullback-Leibler divergence

SGD Stochastic Gradient Descent

RNA Ribonucleic acid

ARI Adjusted Rand Index

NMI Normalized Mutual Information

NGS Next-generation sequencing

ATAC-SEQ The assay for transposase-accessible chromatin with sequencing

Chapter 1

Introduction

Cancer is a group of diseases where abnormal cells bypass the apoptosis mechanism and proliferate to invade other normal tissues. Its heterogeneous nature exhibits on multiple granular levels, from molecular alterations to clinical outcomes. The evolutionary process of cancer comprises genetically distinct subclones across various sites and temporal molecular alterations in the tumor cells[7]. Determining and tackling these heterogeneities with high precision become instrumental in developing effective treatments since heterogeneities facilitate the progress of cancer resistance. These inherent heterogeneities of cancer pose a challenge in developing a systematic classification framework to categorize tumors with the appropriate abstraction based on their biological characteristics. The goal of hierarchical classifications for different cancers is to provide a more accurate diagnosis, predict the progression of the cancer subpopulations and improve prognosis along with informed, effective treatments.

Cancers are most commonly classified by the primary site and the tissues where the first cluster of tumor cells emerges i.e. histological types. An international standard has been established for the nomenclature of cancer types (ICD-O) which serves as a multi-faceted classification of the site, morphology and behavior. Beyond this standard, researchers have been actively investigating meaningful subtypes within each cancer category, with the objective of facilitating the prognosis and treatment prediction for chemotherapy or radiation. For example, the intrinsic subtypes (LumA, LumB, HER2-enriched and basal-like) of breast cancer have manifested their unique behaviours in the risk and survival analysis based on the PAM50 gene dataset[25]. Pan et al. [23] identified several Copy Number Variations (CNVS) associated with specific breast cancer subtypes. Moreover, they found that the expression of certain genes correlated with the occurrence of these CNVS, validating the potential use of CNVS as powerful biomarkers to identify subtypes of breast cancer. While gene expression profiling by microarrays underpins the advancement of breast cancer subtyping, the whole genome sequence has been employed to stratify liver cancer effectively. The genomic subtyping of liver cancers combined with genetic markers contributes to meaningful Single Nucleotide Variant (SNV) subgroups that demonstrate evident discrepancies in terms of patient survival duration[37].

However, instead of concentrating on identifying subtypes within individual cancer, we plan to expand and explore the deep connection across distinct cancers through Pancancer analysis. This holistic overview of genetic profiles across a spectrum of cancers gives us the opportunity to identify common mutated drivers regardless of the primary site and tissue type. The Pancancer Analysis of Whole Genome (PCAWG) project[1] calls for global collaboration on discovering similar mutational patterns across cancers, aiming to generate a comprehensive cancer catalogue by including donor clinical and histopathological data, subclonal reconstructions, purity and ploidy information, splice isoforms and mutational signatures. In 2020, The International Cancer Genome Consortium (ICGC)[32] performed whole genome sequencing over more than 2000 primary tumors and released the corresponding genetic data. Bulk RNA-sequencing data[5] provided by ICGC is an extensive resource encoding cancer molecular profiles. It plays a significant role in revealing commonly altered gene pathways across multiple tissue types that align with the goal of an unsupervised learning task. Cancer cells originating from heterogeneous tissue types could conceivably share a latent structure reflecting common features. A noticeable concern lies in the mixture of dominating tissue effects from different cancers that arise from scaling from subtyping tasks in single cancer to Pancancer classifications. It is often intractable to employ a direct unsupervised learning approach on sequencing-based gene expression from heterogeneous cancers as the resulting clusters constantly reflect the tissue of origin, which camouflage other interesting, potentially important, connections with peripheral signals. Our primary goal is to exclude the tissue-specific gene expression signals from the dataset while preserving the latent structural effect.

Combining all these factors, we propose a novel PCVAE autoencoder model to mitigate dominating tissue effect thereby allowing the model to cluster on latent structural effect. Ideally, the autoencoder segment of the model will learn insightful low-dimensional representations that summarize the connection among diverse types of cancers, whereas the label predictor segment extracts the prominent tissue effect for each cancer type. The hope is that the low-dimensional encodings from the bottleneck layer capture essential embeddings other than the known primary organs and histological types due to the targeted bifurcation process. The latent connections can enrich our understanding of the underlying mechanism of various types of tumors. Recent studies have found a promising connection between genomic alterations such as CNV, SNV and gene expression independent of cancer tissue types. Likewise, our results are expected to unravel the link between gene expression and some biological factor in the context of the pan-cancer relationship.

In this thesis, I divide the entire study into two major portions for clear structure. In the former part, we will formulate the question in a formal framework and define our objectives. Next, we will investigate a few autoencoder network architectures with rigor and assess their ability to disentangle the data with strong tissue effects via latent feature extraction. In the latter part, we propose several models to eliminate the prominent tissue effects from the original samples. In particular, we will introduce a controlled variable to alleviate the tissue effects and integrate it with variational autoencoder neural networks, leveraging the model's performance under different experimental settings. Eventually, the resulting clusters will be analyzed by computing the entropy respectively and visualized to evaluate the latent representation of the data. Biological significance will be assessed by conducting a survival analysis[19] for all patients with attainable data.

Chapter 2

Method

Complicated and unstructured data are ubiquitous in a wide array of disciplines, especially in the current big data era[11]. The dataset provided by ICGC[32] falls into the category above as it encodes a complete set of RNA molecules expressed by the cancerous cells (i.e. transcriptome).

To understand the given data in-depth, it is imperative to quest for a machine learning algorithm that is capable of acquiring effective and disentangled latent representations, while simultaneously preserving the intricate inter- and intra-cluster information. The ideal latent representations need to be both semantically meaningful and statistically independent. Furthermore, we are allowed to incept prior knowledge or extract certain properties from the architecture, which coerces the target machine learning model to be flexible and scalable. In Section 2.1, we provide an exhaustive discussion of the dataset we utilized. In Section 2.2, we introduce the definition of an autoencoder and its variants, focusing on how to take advantage of these architectures to better solve the predefined problem. The proposed novel models will be discussed in the last two sections.

2.1 ICGC Pancan Dataset

The dataset provided PCAWG consortium encompasses valuable embedding, which is a rich yet unwieldy resource to comprehend complex cancer biology. Next-generation sequencing (NGS) is a high-throughput sequencing technology that provides deep and massively parallel sequencing of gene or gene expressions. Bulk RNA sequencing (RNA-seq) is a popular application of NGS which enables the analysis of gene expression levels across cancer samples. It is a powerful technique that enables the quantification of gene expression levels across the entire transcriptome, providing a comprehensive view of the gene expression landscape. We select 6 different types of cancer from

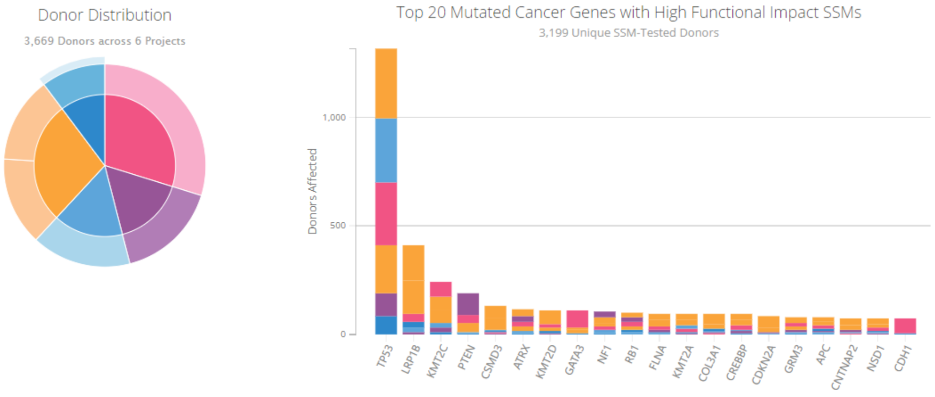

Figure 2.1: Donors distribution and mutated genes from the sampled ICGC dataset

separate primary locations in the ICGC data portal, specifically sequencing-based gene expression data that was produced and examined using the NGS platform. In particular, this RNA-seq dataset encodes dense, high-dimensional information across different types of cancer, yet it may incorporate multiple modalities which increases the difficulty to interpret. In the table, there are 2662 donors with available data type (EXP-S) which constitute the example entries in the dataset. To eliminate the potential regional bias and batch effect, we choose the projects offered by the United States only. In addition, we deliberately choose two subtypes of lung cancer where adenocarcinoma develops in an organ or gland, yet squamous cell carcinoma originates in the squamous epithelium. This choice is designed for examining to what extent can the model disentangle data in the latent space. In Figure 2.1, the orange portions indicate lung adenocarcinoma and lung squamous cell carcinoma. The purple rim represents brain cancer. The pink pie stands for breast cancer and the light blue and dark blue represent ovary and liver cancer respectively.

As for gene features, there are 20502 distinct ones for each donor. We provide several options in the code to select favoured features. For instance, we can perform a logarithmic transformation on the dataset to render it normal-like. Since we only manipulate raw-counts RNA-seq data in the experiment, we need to normalize our data which counteracts the effect caused by varying read depths. We can standardize the data feature-wise if necessary. With respect to feature selection, `nanostring`[12] and `PAM50`[25] gene features have been proven to demonstrate valuable biological significance in a number of experiments. Besides, we can select top k gene features with high variance and high mean absolute deviation by `varmad`. Aggregated gene features are

Table 2.1: Detail of the selected ICGC dataset

Code	Name	Site	Donors	EXP-S
BRCA-US	Breast Cancer - TCGA, US	Breast	1,093	1041
GBM-US	Brain Glioblastoma Multiforme - TCGA, US	Brain	595	159
OV-US	Ovarian Serous Cystadenocarcinoma - TCGA, US	Ovary	584	262
LUAD-US	Lung Adenocarcinoma - TCGA, US	Lung	518	478
LUSC-US	Lung Squamous Cell Carcinoma - TCGA, US	Lung	502	428
LIHC-US	Liver Hepatocellular carcinoma - TCGA, US	Liver	377	294
Total			3669	2662

a viable option in the dataset object as well. Notice that in the table, the data points are imbalanced across cancer categories. Therefore, we can select the same number of donors uniformly at random for each cancer type. To visualize the data, we preliminarily invoke three dimension reduction techniques, i.e. Principle Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (T-SNE) and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP). The results will be presented in the next chapter, so as the experiment's configurations. These dimension-reduction approaches will be adopted to visualize the bottleneck layer of the autoencoder architecture (i.e. the latent representation).

2.2 Autoencoders

Autoencoders are a versatile family of neural networks that are trained to learn compressed representations of input data by minimizing the difference between the original input and its reconstructed output. Specifically, an autoencoder consists of an encoder and a decoder network, which broadly undertakes the corresponding tasks:

- **Encoder**: A function $e(\cdot)$ maps the input data to low-dimensional latent representations.
- **Decoder:** A function $d(\cdot)$ aims to reconstruct latent representations from the embedding back to the original input space.

Due to the non-linearity supported by the activation functions, these neural networks excel in tasks where linear methods such as Principle Component Analysis (PCA) or singular value decomposition are not sufficient to capture the underlying structure of the data. Substantial research have been done by applying autoencoder for the Pancancer data in terms of clinical outcome[31], multi-omics integration[40], somantic mutation[22] and gene expression[35] etc. which manifests the model's potential.

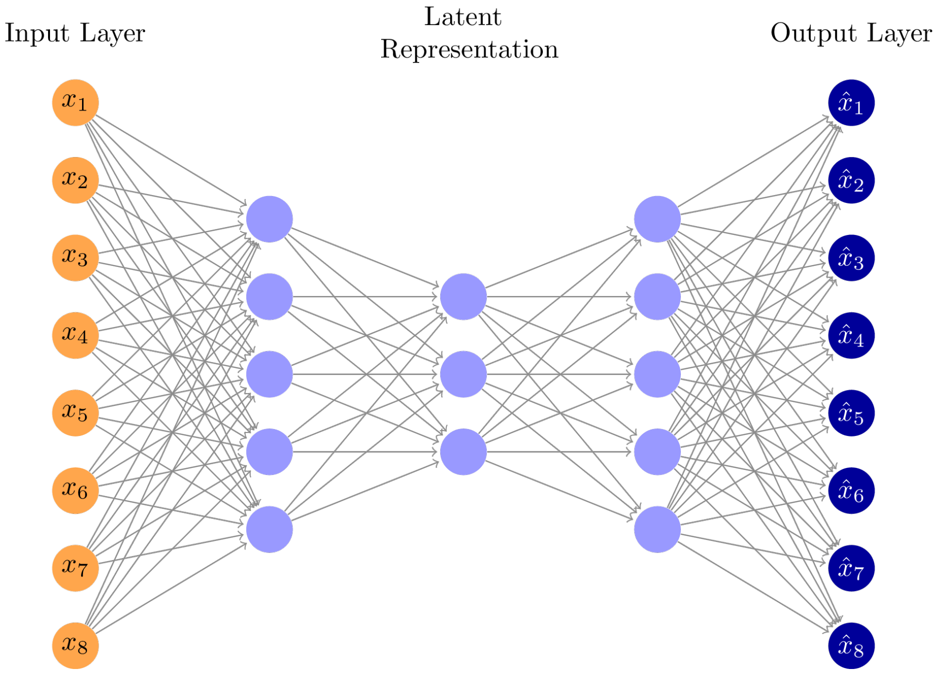

Figure 2.2: Schematic diagram of a standard autoencoder. In this example, the original dataset and the reconstructed one both have 8 dimensions. The bottleneck layer has 3 dimensions represented by 3 purple nodes in the middle.

There are several variants of autoencoders, which we will introduce in the following sections. Each type has its own specific architecture and training procedure, but all share the common goal of learning compressed representations of input data with high fidelity while discarding irrelevant or redundant information.

2.2.1 Vanilla Autoencoders

Mathematically, we assume any input dataset X has m examples and n features. A standard autoencoder contains an encoder $e_{\theta_1}(\cdot)$ and a decoder $d_{\theta_2}(\cdot)$, which are parametrized by both sequences of neural networks with activation functions. These functions with reversed domain and codomain $e : \mathbb{R}^n \to \mathbb{R}^k,\ d : \mathbb{R}^k \to \mathbb{R}^n$ where the parameters $\langle \theta_1, \theta_2 \rangle$ are trained in a network through backpropagation and optimization. Regarding the structure, the encoding layers and decoding layers denote the neural networks on the left-hand side and right-hand side in Figure 2.2. The bottleneck layer in an autoencoder refers to the hidden layer in the neural network that learns the latent representation of the input data. It is called a bottleneck layer because it often has the least number of neurons compared to the input and output layers, thus creating a bottleneck in the flow of information through the network. The dimension of the latent space is k which also stands for the number of neurons in the bottleneck layer. To learn a compact representation of the input data, an objective function has been established in a general form. It can be expressed as

$$L = \sum_{x \in \mathcal{X}} S(x, d_{\theta_2}(e_{\theta_1}(x))) \tag{2.1}$$

where S denotes a measure of the distance between two vectors. L is considered as the loss or objective function in machine learning which is required to be optimized during the training process.

A detailed reference of mathematical notation that we adopt in this thesis can be found in Chapter Mathematical Notation.

2.2.2 Variational Autoencoders

Deep generative models have been a heated research topic in the field and they have achieved monumental accomplishments in producing realistic-looking data from which human cannot differentiate (Deep Fakes)[29][21]. The past decade has witnessed a success in single-cell transcriptomics by employing deep generative models (scVI)[20]. These models are a family of probabilistic graphical models stemming from Bayesian inference, with the primary objective of generating samples $\hat{x} \sim p_\theta(x)$ similar to the ones $x \sim p_{d \in \mathcal{D}}(x)$ from the distribution embedded in the training set under a deep neural net framework. Alternatively, a deep generative model attempts to approximate the high-dimensional density in the observed dataset with certain assumptions imposed on the latent manifold [29]. In general, deep generative models are designed to solve the following optimization problem

$$\min_{m \in \mathcal{M}} S(p_m(x), p_{\mathcal{D}}(x))$$

where S is a measure of the distance between two probability distributions.

This model family consists of two essential elements, a generative phase and an inferential phase[2]. As a generative model, it can capture the joint probability over the entire set of variables by generating new data instances from learnt, tractable hidden distribution, whereas the discriminative model depicts the map only from observations to the predicted labels. It also enables us to encode rich and complex information in the latent space and provide reasoning about the data in a more sophisticated way than discriminative models trained by supervised methods[9]. In terms of inference, deep generative models provide the opportunity to assess three fundamental queries i.e. density estimation $p_\theta(x)$, sampling $\hat{x} \sim p_\theta(x)$ and representation learning. We pay close attention to the last query since it matches the interest of this thesis.

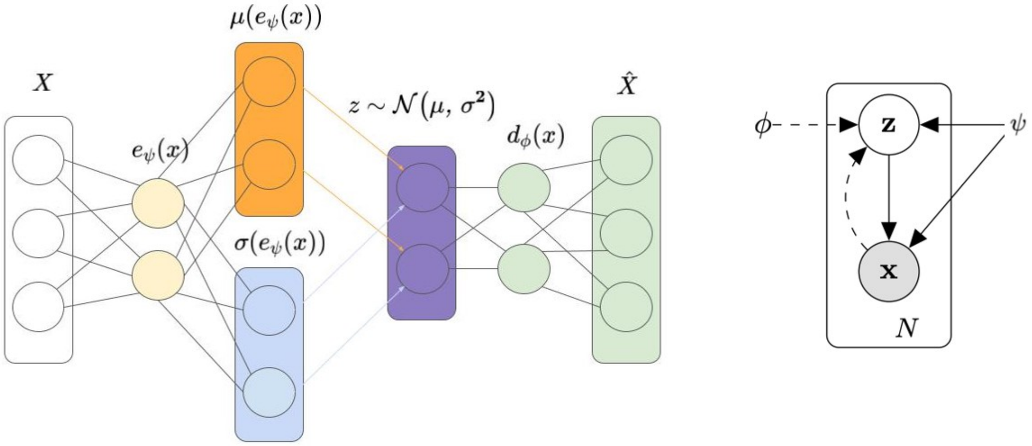

Figure 2.3: A schematic diagram of VAE and a graphical model representation [16]

The Variational Autoencoder (VAE) [16] is an important branch in the deep generative model family with a wide array of applications in computer vision [10][39], and linguistics [3]. In the context of Pancancer, biologically related latent space in cancer transcriptomes (RNA-seq data) from ICGC have been extensively studied through VAE, as evidenced by Way and Greene [36]. Similarly, VAE has been employeed for data integration on Pancancer dataset and heurstic design principles were proposed and evaluated [30]. The structure of VAE has been delineated in Figure 2.3 where the solid arrows represent the generative step and the dashed arrows imply the inferential step. In VAE setting, the encoder is often regarded as a recognition model and the decoder stands for a generative model. The model inherits the merits from the structure of normal autoencoders, yet biases on learning the embedded distribution in the input and how it projects the input to the latent space. In addition, this model can be regarded as a two-fold, independently parametrized architecture including a decoder $p_\phi(x_i \mid z)$ and an encoder $q_\psi(z \mid x_i)$. The encoder builds up an educated estimation about the posterior on z and transfers it to the decoder in the forward pass, while updating its parameters during learning iterations [17]. Considering the opposite direction, the decoder establishes a scaffolding for the encoder to learn meaningful representations. Therefore, the decoder and encoder mutually benefit each other, which results in refining the latent features and generating better samples [17]. In contrast to normal autoencoders, VAES incorporate the process of enforcing the latent representations to be meaningful by minimizing the 'statistical distance' or 'discrepancies' S between two distributions rooted in the model. Specifically, these two distributions are the posterior learnt from the encoder and the prior $p_\phi(z)$. The exact prior can be computed by using a series of fundamental probability rules such as Baye's theorem and marginalization upon the trained distribution $p_\phi(x_i \mid z)$ from the decoder. Mathematically speaking, the Kullback-Leibler divergence (KL DIVERGENCE) $D_{KL}(\cdot||\cdot)$ acts as a proxy for measuring how one probability distribution differs from another via calculating the relative entropy represented in information for both distributions. Note that this measure is not a metric since it is not symmetric in the sense that the order of probability distributions does matter as arguments in KL DIVERGENCE. The motivation behind VAES comes from a postulate on the mechanism of how data is produced in the physical world. The postulate relies on the cycle of scientific methods where we hypothesize the theories and verify the theories through observations [17]. Likewise, VAES delineate the input data by constructing an abstraction $p_\phi(x \mid z)$, i.e. the latent structure of the generating process, and perform downstream inference subsequently.

In Appendix A, I will elaborate on the loss for VAES and provide a mathematical derivation of Evidence Lower Bound (ELBO), which is a lower bound for $\log(p(x_i))$. Hence, the variational lower bound for log marginal likelihood of a datapoint x_i can be expressed as

$$\mathcal{L}(x_i, \psi, \phi) = \mathbb{E}_{q_\psi(z|x_i)}\left[\log(p_\phi(x_i|z))\right] - D_{KL}(q_\psi(z|x_i) \mid\mid p_\phi(z))$$

where the first component can be approximated by the reconstruction loss between the input and the output and the second component represents the regularization via KL DIVERGENCE. In terms of optimizing, we desire to maximize the evidence probability which means we can take the negative of ELBO to obtain a loss for VAE. Canonically, we will perform Stochastic Gradient Descent (SGD) on this objective function as an optimization criterion. Nevertheless, there is an issue that arises from the backward pass when the gradient backpropagates through the sampling process

$\hat{z} \sim p_\phi(z|x)$ [17]. It is difficult to acquire an unbiased gradient estimator since the encoder itself is a function of ψ.

$$\nabla_\psi \mathcal{L}(x_i, \psi, \phi) = \nabla_\psi \mathbb{E}_{q_\psi(z|x_i)} \left[\log \frac{p_\phi(x_i, z)}{q_\psi(z|x_i)} \right] \neq \mathbb{E}_{q_\psi(z|x_i)} \left[\nabla_\psi \log \frac{p_\phi(x_i, z)}{q_\psi(z|x_i)} \right]$$

Now, the reparameterization trick comes into play and circumvents this issue [16][27]. The idea is that we propose a differentiable function h to represent the random variable by transforming another random variable ε which is easy to tackle and manipulate. This function can be written as

$$z_i = h(\varepsilon, \psi, x_i)$$

where the distribution of ε is totally independent from x_i, ϕ. As a result, we are allowed to exchange the position of expectation and gradient operator which yields

$$\begin{aligned} \nabla_\psi \mathcal{L}(x_i, \psi, \phi) &= \nabla_\psi \mathbb{E}_{q_\psi(z|x_i)} \left[\log \frac{p_\phi(x_i, z)}{q_\psi(z|x_i)} \right] \\ &= \nabla_\psi \mathbb{E}_{p(\varepsilon)} \left[\log \frac{p_\phi(x_i, z)}{q_\psi(z|x_i)} \right] \\ &= \mathbb{E}_{p(\varepsilon)} \left[\nabla_\psi \log \frac{p_\phi(x_i, z)}{q_\psi(z|x_i)} \right] \\ &\approx \nabla_\psi \log \frac{p_\phi(x_i, z)}{q_\psi(z|x_i)}. \end{aligned}$$

In this case, we build up a simple Monte Carlo estimator for the ELBO which retains unbiasness. To simplify the computation and obtain a closed form for ELBO, we invoke the Gaussian assumption on the latent variables and we eventually achieve our VAE model customized for the given ICGC dataset in the first stage. Another reason for using this assumption is that the input data is real-valued which matches the support of the Gaussians. This assumption results in a posterior that takes the form of a Gaussian distribution with diagonal covariance, allowing for the use of variational inference to effectively approximate the true posterior distribution[30]. Here is a list of random variables that are inherent in our VAE model under the Gaussian assumptions.

$$\varepsilon \sim \mathcal{N}(0, \boldsymbol{I}_k) \tag{2.2}$$

$$z = h(\psi, \varepsilon, x) = \mu_\psi + \sigma_\psi \odot \varepsilon \tag{2.3}$$

$$q_\psi(z_i \mid x_i) = \mathcal{N}(z_i; \mu_\psi(x_i), \sigma^2_\psi(x_i)) \tag{2.4}$$

$$p_\phi(x_i \mid z_i) = \mathcal{N}(x_i; \mu_\phi(z_i), \sigma^2_\phi(z_i)) \tag{2.5}$$

where $\mu_\psi, \log \sigma_\psi$ are parameters learned from the encoder segment of the VAE model and $\odot$ is the notation for the element-wise product. To maximize the ELBO, we desire to minimize the loss function which is proportional to the negative of ELBO. Overall, we denote B to be the batch of samples generated from the encoder and the explicit form of the VAE loss function can be expressed as

$$L = -\frac{1}{2}\sum_{t=1}^{k}(1 + \log(\sigma^2_t(\psi, x)) - \sigma^2_t(\psi, x) - \mu^2_t(\psi, x)) - \frac{1}{|B|}\sum_{b \in B} \mathbb{E}_{q_\psi(z|x_i)}[\log p_\phi(x_i|z_b)] \tag{2.6}$$

and the minima of such loss function can be reached through multiple optimizers provided by the machine learning framework. (full derivation is in Appendix A).

2.2.3 β-VAEs

Beta-VAE [15][4] is a modification of VAE framework that introduces an additional hyperparameter, called beta, to the standard VAE objective function. The beta hyperparameter controls the trade-off between the quality of the generated samples and the disentanglement of the learned latent representation. The β-VAE objective can be written as

$$\mathcal{L}(x_i, z; \psi, \phi, \beta) = \mathbb{E}_{q_\psi(z|x_i)}\left[\log(p_\phi(x_i|z))\right] - \beta D_{KL}(q_\psi(z|x_i) \mid\mid p_\phi(z)) \tag{2.7}$$

where $\beta > 1$ imposes a stronger constrain for the posterior $q_\psi(z|x)$ to match the factorised Gaussian prior $p(z)$[15]. Observed that this objective is equivalent to a standard VAE model when $\beta = 1$. A refined β-VAE training objective has been proposed by Burgess et al. which can be expressed as

$$\mathcal{L}(x_i, z; \psi, \phi, \gamma, C) = \mathbb{E}_{q_\psi(z|x_i)}\left[\log(p_\phi(x_i|z))\right] - \gamma|D_{KL}(q_\psi(z|x_i) \mid\mid p_\phi(z)) - C| \tag{2.8}$$

where C increases gradually from zero to a value that is sufficient for decent reconstruction.

The disentanglement property refers to the ability of the autoencoder to learn a representation where each dimension of the latent space corresponds to a semantically meaningful feature of the input data. For instance, in an image dataset, the latent dimensions could represent attributes such as the pose, shape, and color of an object.

β-VAE encourages disentanglement by adding a penalty term to the VAE objective function, which encourages each dimension of the latent space to be used for a separate and meaningful attribute. Applying a higher coefficient for KL DIVERGENCE can result in a compromise between the fidelity of the reconstructed data and the disentangled characteristics of the latent embeddings[4]. In general, the value of beta determines the strength of this penalty term, and a larger value of beta results in a more disentangled latent space[15].

2.3 PCAE model

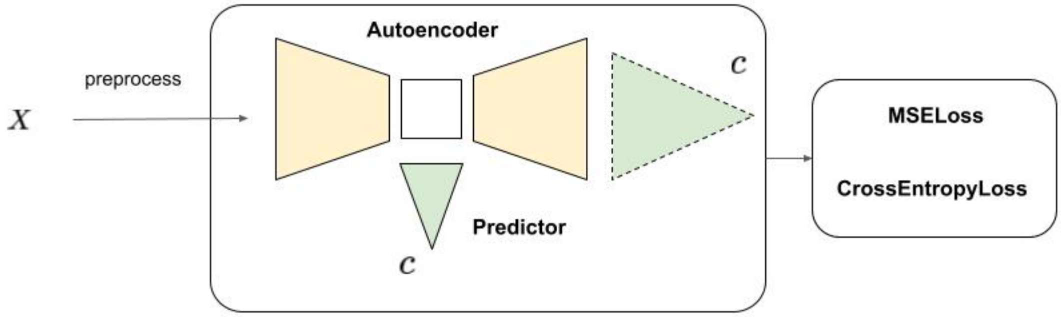

Figure 2.4: PCAE model. The dataset was preprocessed initially and fed into the autoencoder. There are two possible positions of the predictor, one attached to the bottleneck layer and the other one connected with the output layer. The final loss was evaluated as the sum of MSEloss and CrossEntropyloss. The parameters were refined by backpropogation (in the backward pass).

To quantify and eradicate the tissue effect, we require an adaptive structure that can extract this site information from the sequencing-based gene expression data. Intuitively, one possible solution is to use a multi-class predictor to bifurcate the information flow and sift out the tissue effect. As the diagram depicts above, we append the predictor to either the bottleneck layer or the end of the decoding layers (c_i is the tissue label for each patient x_i). The final loss for our model will be the combination of the reconstruction loss (between X and $\hat{X}$) and the prediction loss (between c and the ground truth i.e. Cross Entropy Loss[41]). By incorporating the loss of the predictor, there exhibits a trade-off between the fidelity of reconstructions and the alignment between each patient tumor sample and the corresponding primary site. Consequently, the proposed multi-class classifier is able to effectively capture the tissue effect, which in turn relieves the autoencoder architecture from this burden and allows it to focus on generating meaningful latent feature encodings independent of primary sites. By isolating the tissue effect from the encoding process, this approach has the potential to facilitate the identification of novel biological clusters.

2.4 PCVAE model

We propose a novel genre of PCVAE models which serves as a modification of the standard VAE model. Due to the high flexibility and interpretability of VAE, we can incept a learnable parameter ν in the VAE which controls the tissue effect. According to the current ICGC dataset we pick, there are six cancers from distinct primary sites which implies that ν incorporates 6 rows in total. Regarding a single row of ν, we initialized it as a standard normal vector with varying size which relies on the position we insert ν in the VAE network. We speculate that this variable will extract the prominent tissue signal from our dataset, which granted permission to cluster on the latent representations with ν set to zero (frozen). In the first model, we place ν to the reparametrization process which indicates the size of ν ought to match the dimension of the latent layer k.

$$\nu_c \sim \mathcal{N}(0, \boldsymbol{I}_k) \tag{2.9}$$

$$\varepsilon \sim \mathcal{N}(0, \boldsymbol{I}_k) \tag{2.10}$$

$$z = h(\psi, \varepsilon, x) = \nu_c + \mu_\psi + \sigma_\psi \odot \varepsilon \tag{2.11}$$

$$q_\psi(z_i \mid x_i) = \mathcal{N}(z_i; \mu_\psi(x_i), \sigma^2_\psi(x_i)) \tag{2.12}$$

$$p_\phi(x_i \mid z_i) = \mathcal{N}(x_i; \mu_\phi(z_i), \sigma^2_\phi(z_i)) \tag{2.13}$$

Note that the subscript c denotes the numerical encoding of the corresponding primary site of a patient x_i. In particular, c represents the set containing sites such as Brain, Lung-AD, Lung-SC, Liver, Ovaries, Breast where the detail can be found in Table 2.1.

As for the second model, we attach ν to the last layer of the decoding layer which means the size of ν has to adapt to the dimension of original data space n.

$$\nu_c \sim \mathcal{N}(0, \boldsymbol{I}_n) \tag{2.14}$$

$$\varepsilon \sim \mathcal{N}(0, \boldsymbol{I}_k) \tag{2.15}$$

$$z = h(\psi, \varepsilon, x) = \mu_\psi + \sigma_\psi \odot \varepsilon \tag{2.16}$$

$$q_\psi(z_i \mid x_i) = \mathcal{N}(z_i; \mu_\psi(x_i), \sigma_\psi^2(x_i)) \tag{2.17}$$

$$p_\phi(x_i \mid z_i) = \mathcal{N}(x_i; \nu_c + \mu_\phi(z_i), \sigma_\phi^2(z_i)) \tag{2.18}$$

Notice that one advantage of the design choice for these models is that there is no post-processing step required for visualizing the clusters. Since the encoding step is independent of the effect of ν, we can encode the test data and run the clustering algorithms upon the output without the cumbersome procedure to isolate ν from the latent embeddings.

We intend to discover the new clusters by clustering on the μ layer after the tissue effect removal. In particular, we will employ a graph-based clustering algorithm called Leiden, which has been shown to be highly effective in clustering large-scale networks[14][34]. The algorithm starts by initializing each node as its own cluster. Subsequently, it iteratively merges clusters that lead to a decrease in a quality function that measures the modularity of the network.

Chapter 3

Result

To simplify the process of developing scalable machine learning models, we adopt `Pytorch` which is a fully featured machine learning framework. The code used for this thesis is available at https://github.com/Roth-Lab/PCVAE and all results are reproducible. In the first three sections, we will show that variants of autoencoder structure can cluster according to the tissue labels effortlessly. In the last two sections, we verify if the tissue effect has been eliminated and evaluate the novel clusters with the latent representations from the model which effectively extracts the tissue signal.

3.1 Preliminary

We begin by attempting to cluster upon the raw dataset to ensure that the tissue signal is strong enough. After we normalize the data for each patient, we visualize data directly by running UMAP to reduce the dimension down to two. In Figure 3.1, we randomly select 150 patients in each cancer type to balance the data for the first row of plots and the entire dataset for the second row. Comparing the plots column-wise, it is evident that the more the number of examples, the better quality of the clusters is in the experiment. Notice that the patients with Lung-AD and Lung-SC displayed in the figure are intertwined with each other. A reasonable claim would be the histological signal from cancers is not as prominent as the signal from the primary organ. Based on the quality of the clusters, the tissue effect exhibits a moderately pronounced signal dominance. This observation underscores the importance and necessity of developing a machine-learning model for processing this signal.

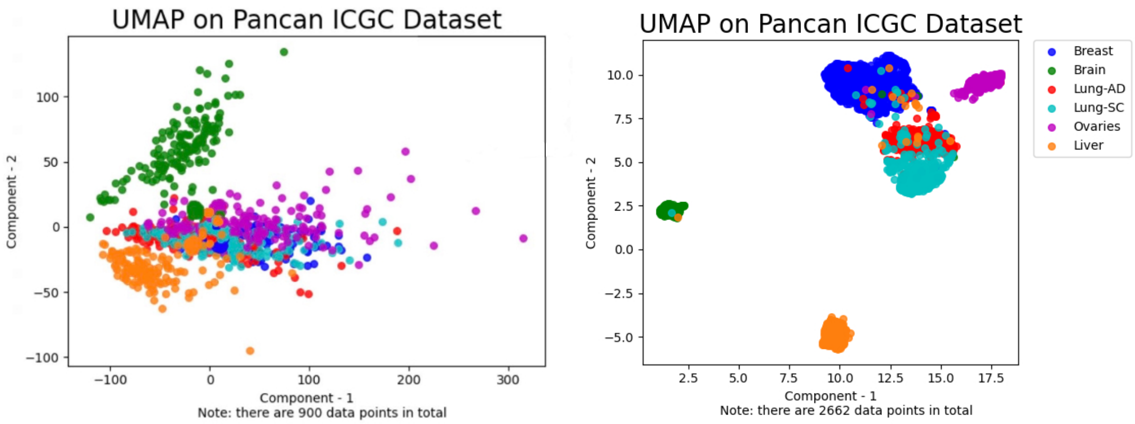

Figure 3.1: Preliminary visualization of the dataset

3.2 Autoencoder

In the normal autoencoder setting, we configure the layers' dimension as $[20501, 2048, 1024, 256, 16, 256, 1024, 2048, 20501]$. During training, we split the data randomly into a training set and a validation set. The loss on the validation set (validation error) determines whether the model saves the current parameters. In the experiment, we aim to compare the performance of various optimizers and determine which one may contribute to the fast and robust convergence of clusters. As for the left-most plot in Figure 3.2, the optimizer invoked is `Adam` with a learning rate 1e-3 and weight decay 1e-5. We apply the SGD optimizer with `momentum` set to 0.9 for the middle plot. An addition `Nesterov` momentum for SGD is imposed on a standard SGD for the right-most figure. The batch size is the length of the entire training dataset to achieve a complete gradient. We set the number of epochs to 20000. In the visualization after UMAP, there is a clear discrepancy between the quality of clusters with latent features optimized by `Adam` and features optimized by other gradient-based optimizations. We obtain two significant observations from this figure. Firstly, a standard autoencoder can produce good clustering on the tissue effect. The second one is that `Adam` optimizer outperforms other optimizers in the current experiment configuration. For the sake of simplicity and efficiency, we only adopt `Adam` optimizer in the following autoencoder architectures.

3.3 VAE

Moving on towards VAE, we tune the hyperparameters carefully to achieve evident clustering results. By ensuring the quality of clusters that correspond to tissue effects in the bottleneck layer

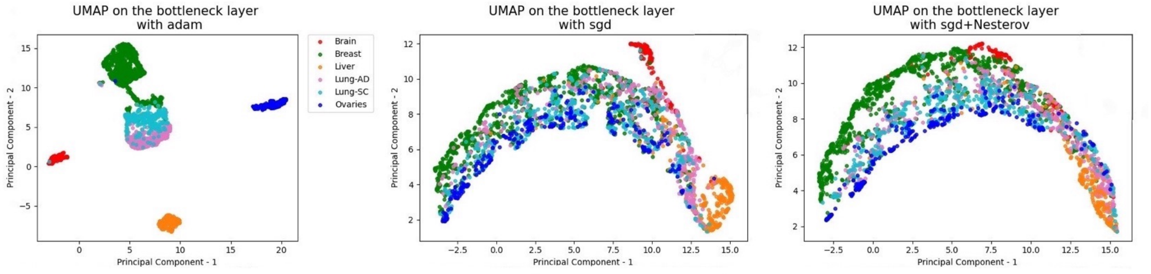

Figure 3.2: Visualization of the latent layer of vanilla autoencoder using distinct optimizers

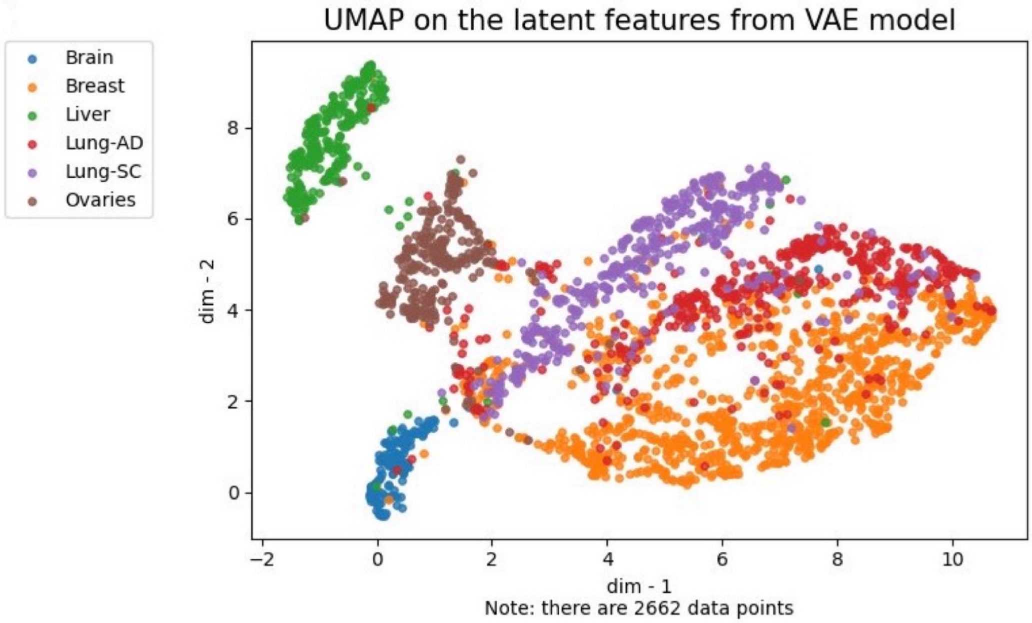

Figure 3.3: Visualization of the bottleneck layer of VAE

with the current configuration, the groundwork is laid for the proposed model to effectively remove the tissue signal. The optimizer has the same configuration as the `Adam` in the previous section. The dimensions for the hidden layers are assigned with $[1024, 512]$ in the encoder and $[512, 1024]$ for the symmetric purpose. In terms of initialization, we deploy `Xavier` initialization[13] to avoid local minima in the objective function. To prevent possible overfitting of the training data, we incorporate the early stopping mechanism where the training process will be terminated if the validation error does not improve up to a number of iterations (i.e. patience). We also add `Dropout` layer in the neural networks for a similar purpose. In the following analysis, we visualize the μ_ψ from equation (2.3) instead of the latent reparametrization z to further mitigate the noise incurred by the variance. As for the preprocessing, we normalize and log-transform the dataset. The features that we select are the top 2000 under `varmad` criterion and union with `nanostring` and `PAM50` where the detail can be referred in Chapter 2. The resulting clusters in Figure 3.3 demonstrate clear dominating tissue effects which indicate that such effects have been preserved and refined in this architecture.

In the β-VAE model, the β coefficient plays a significant role in terms of enforcing the latent space to be more disentangled. In the experiment, we set $\beta = 100$ and achieve the following clustering results illustrated in Figure 3.4. Observed from the UMAP plot, the cohesion for each cluster largely increases comparing the plot generated from the VAE model. This observation from the experiment validates that an appropriate β can result in more disentangled characteristics of the latent embeddings.

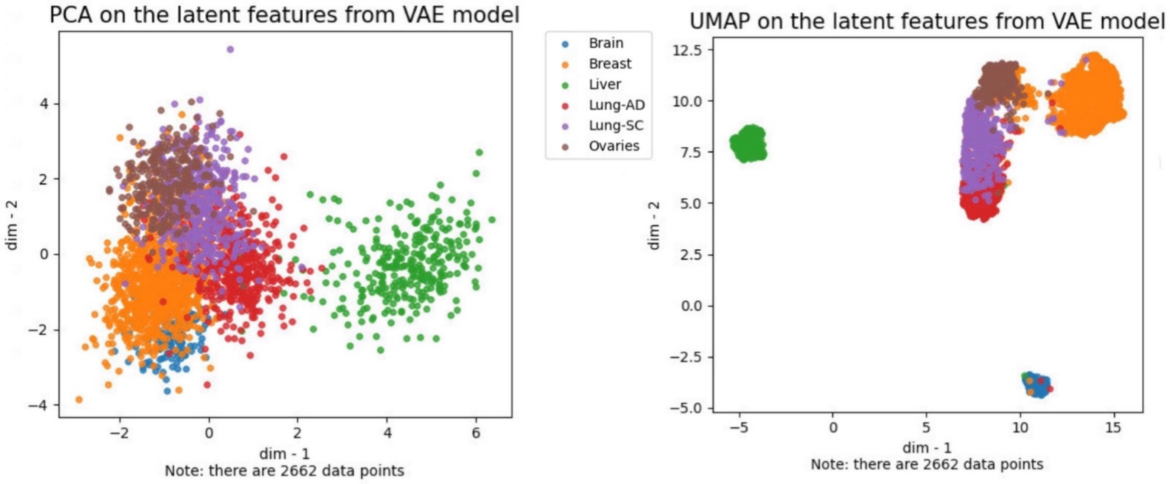

Figure 3.4: Visualization of the bottleneck layer of beta VAE

3.4 PCAE

Before this experiment, we first establish a promising multi-class predictor with the following configuration. The structure of the predictor can be expressed as $[20501, 7000, 2048, 512]$ with `ReLU` as the activation function. This predictor can achieve 99% accuracy during the testing phase. We perform two experiments depending on the position of the classifier (i.e a specific layer in the autoencoder where we retrieve the data). See Figure 2.4. The plot **a** in Figure 3.5 corresponds to

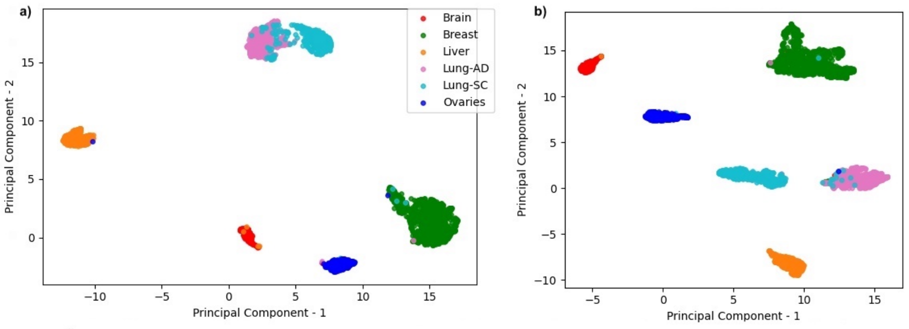

Figure 3.5: Visualization of the bottleneck layer of the PCAE model using UMAP. The autoencoder structure inherits the most performant setting from the preliminary analysis. Visualization was created on the standardized and normalized dataset with aggregated gene features. The dimension of the autoencoder is [2443, 512, 256, 32, 256, 512, 2443] and the shape of predictor is [32, 16, 8] in **a** and [2443, 256, 16] in **b**.

the model with predictor in solid lines. In this scenario, the predictor inputs the value produced from the bottleneck layer of the standard autoencoder. On the other hand, the predictor takes the output of the reconstructed X which results in the plot **b** (the model with predictor in dotted lines in Figure 2.4). Unfortunately, PCAE instead amplifies the tissue effect from heterogeneous cancers which does not satisfy our expectations. One plausible explanation may be that the prediction loss term in the objective function enhances the tissue effect in the latent space due to the additional penalty.

3.5 PCVAE

Table 3.1: Experiment setup for PCVAE

Model Name	Position of ν	Beta
VAEb	bottleneck layer during reparametrization	No
VAEd	after decoding layers	No
betaVAEHb	bottleneck layer during reparametrization	Yes
betaVAEHd	after decoding layers	Yes
betaVAEBb	bottleneck layer during reparametrization	Yes*
betaVAEBd	after decoding layers	Yes*

We inherit the hyperparameters in the previous sections, especially from section 3.3. In terms of initialization, we utilize `kaiming` initialization for models based on beta VAE. In Table 3.1, the model betaVAEB applies the objective function (2.8) while the model betaVAEH uses the objective function (2.7). The asterisk in the beta column suggests that beta is equal to the value of γ in β-VAE objective (2.8).

According to the visualization of the latent embeddings Figure 3.6, we see that the tissue signal is diluted and extracted by ν reasonably well in **a-b, e-f** while the signal remains prominent in the results in the middle row **c-d**. This implies that our betaVAEBb and betaVAEBd models do not remove the tissue effect by incorporating ν. Thus, we shall run the clustering algorithm on the latent representations provided by other models. During the training phase, the validation error drops violently after 50-100 epochs so as the reconstruction loss and the KL DIVERGENCE, as shown in Figure A.1. Note that the last column of plots refers to the VAE base model with the reduction method set to 'sum'. This explains the absurdly large loss compared to the first column since we employ an average reduction method for the betaVAEB models in lieu of the 'sum' method.

Finally, we run the Leiden clustering algorithm[34] upon the latent μ, which encompasses a minimal amount of tissue effect. In models VAEb and VAEd, the algorithm detects 12 new clusters and uncovers 9 new clusters respectively. Meanwhile, the algorithm has identified 12 and 14 new clusters from models betaVAEHb and betaVAEHd respectively, see Figure 3.8. As this study is exploratory in nature, there are no ground truth labels available for each data point in novel clusters. Consequently, popular evaluation measures such as the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) cannot be applied to assess the quality of clustering. Instead, we report the Silhouette score[28], the Calinski Harabasz score[6] and the Davies-Bouldin index[8] to evaluate the internal validity of these novel clusters in Table 3.2. In terms of the assessment of the clusters, betaVAEHb and betaVAEHd models produce clusters with the lowest Davies-Bouldin index which means the intra-class similarity and inter-class differences are reasonably prominent. These models also dominate under the evaluation of Calinski Harabasz measure which implies the novel clusters are dense and well separated. During the survival analysis[19], we tackle the survival time for each patient in days and new cluster arrays. In the Kaplan-Meier curve in Figure 3.10, we see the patients' survival rate within each novel cluster. This curve estimates the survival function from data and shows the probability that the current cohort will survive up to t days.

Subsequently, the entropy H of each novel cluster i found from the Leiden algorithm can be written as

$$H_i = -\sum_c p_{ic} \log(p_{ic}) \tag{3.1}$$

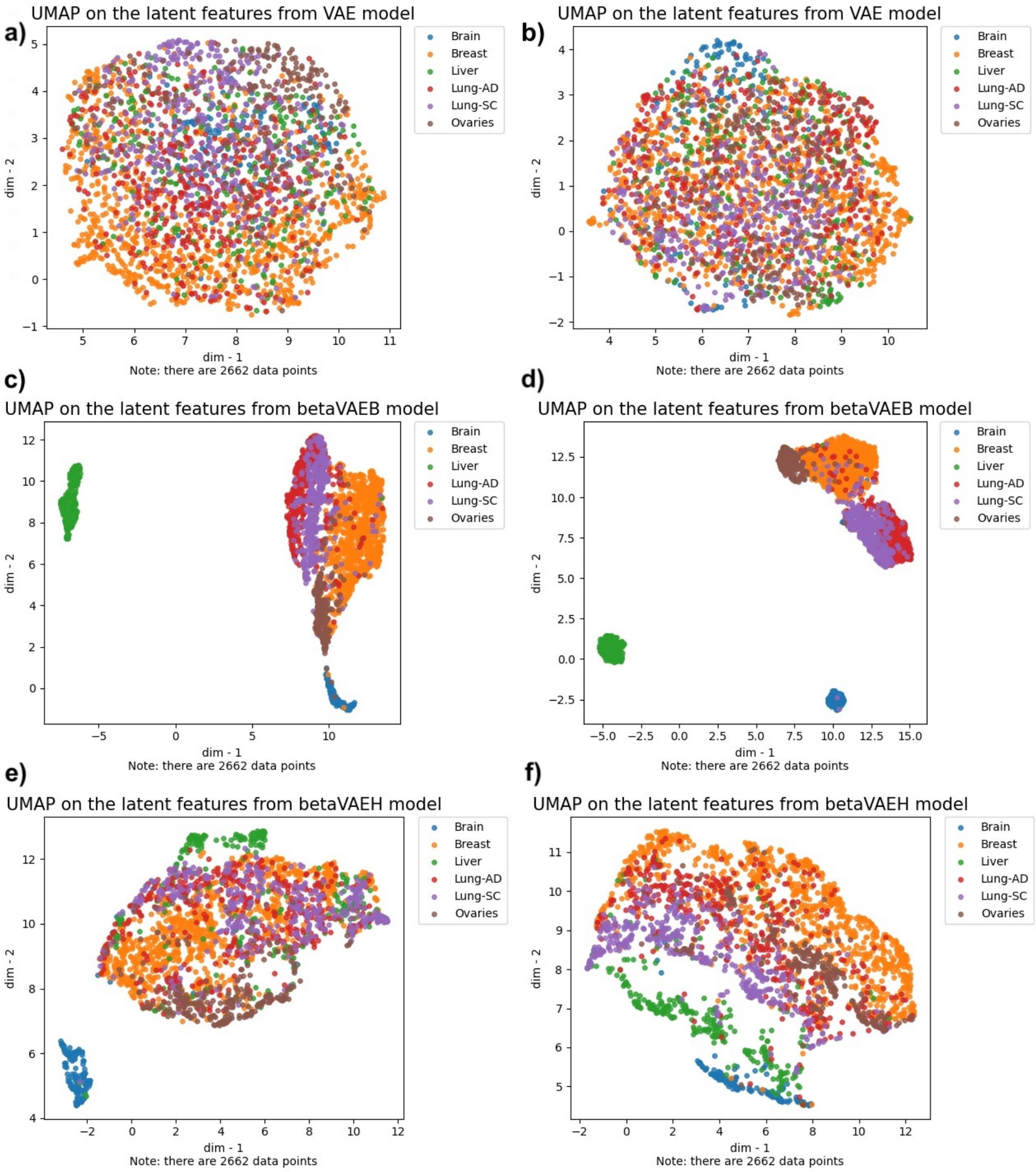

Figure 3.6: Visualization of latent embeddings from PCVAE models. **a,c,e** correspond to the models where ν is at the bottleneck layer while **b,d,f** refer to the model where ν locates after the decoding layers. In **a-b**, there are no obvious clusters observed yet Brain tissue effect still dominates in **e-f**.

where c is the tissue label. Note that p_{ic} stands for the proportion of the number of patients diagnosed as c cancer within cluster i with respect to the total number of patients in cluster i. According to Figure 3.7, we conclude that the entropy of clusters from VAEb and VAEd models is distributed relatively uniformly and sustains at around 1.0. This indicates the tissue signal spreads evenly within clusters from VAE base PCVAE models while tissue signal varies moderately from betaVAE base PCVAE models. Notice in subplots **c,d**, Cluster 11 from **c** has only 118 donors with Liver cancer and Cluster 13 from **d** encompasses 56 donors with Liver cancer as well. To assess the attributes of the novel clusters, we alternatively visualize the donors' age for each cluster, see Figure 3.9. The plot depicts that the median age of all clusters hovers around 60-70 years old. The number of outliers within each batch of clusters constructed from different models exhibits a notable divergence in age, both the age at diagnosis and age at last follow-up. However, the spread of age and distribution is approximately the same across different clusters which indicates age attribute shows a minimal batch effect.

In the resulting KM curves, we observe all the novel clusters follow a similar survival trend in panel **a,b**. Cluster 1 in panel **b** exhibits high survival probability during timeline 500 to 2000, which exemplifies that patients with Breast cancer in the US have a high survival rate. This is because the number of Breast cancer patients (273) dominates in such cluster. In panels **c,d**, the curves are more dispersed so that the donors in these clusters follow distinguishable survival trends. Notice that cluster 10 in subplot **c** shows a notably diminished survival rate. Cluster 10 is constituted of 153 patients with Brain tumor, 2 patients with Liver tumor and 2 Lung-SC cancer patients. This result matches the properties of brain cancer, which are highly aggressive and fast transition to malignancy. According to subplot **d**, Cluster 7 demonstrates a relatively high survival rate in the long tail. By scrutinizing such cluster, the number of Breast cancer patients dominates in terms of tissue type. Hence, patients with breast cancer tend to have longer periods of survival, aligning with the current knowledge of this cancer. Finally, we fit Cox's proportional hazard regression model with a L1 penalizer. The fitted coefficients with confidence interval are illustrated in Figure A.2. By varying the value of the L1 penalizer, we obtained a branching plot for various clusters and tissue types (coefficients v.s. L1 penalty), see Figure 3.11. Based on the control group **c**, the clusters in **a,b** are similar since they all converge to 0.0 when L1 increases to approximately 0.1. Nevertheless, the coefficients for Liver, Lung-AD and Lung-SC cancers converge at that identical point. This strongly implies that Cluster 10 found from betaVAEHd model and Cluster 11,6,7 demonstrate equivalently pronounced signal intensity as Ovary, Brain and Breast cancers (for the detail statistics of these special clusters, please refer to Table A.1).

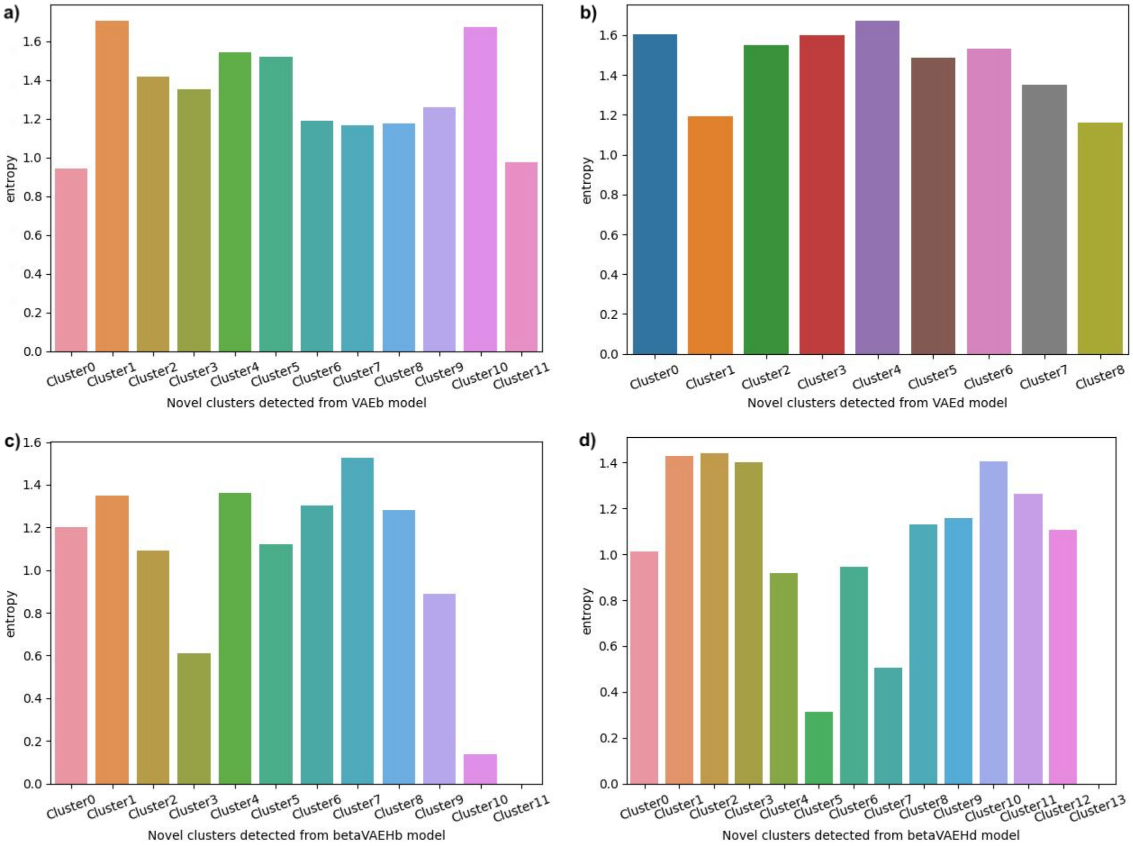

Figure 3.7: Entropy values for the novel clusters of donors from VAEb (**a**), VAEd (**b**), betaVAEHb (**c**), betaVAEHd (**d**) base PCVAE models.

Table 3.2: Evalution scores for novel clusters

Model Name	Silhouette score	Calinski Harabasz score	Davies-Bouldin index
VAEb	-0.06074	7.95771	9.45565
VAEd	-0.05967	5.46833	11.59724
betaVAEHb	-0.04762	65.90446	7.55795
betaVAEHd	-0.07778	54.78236	7.70724

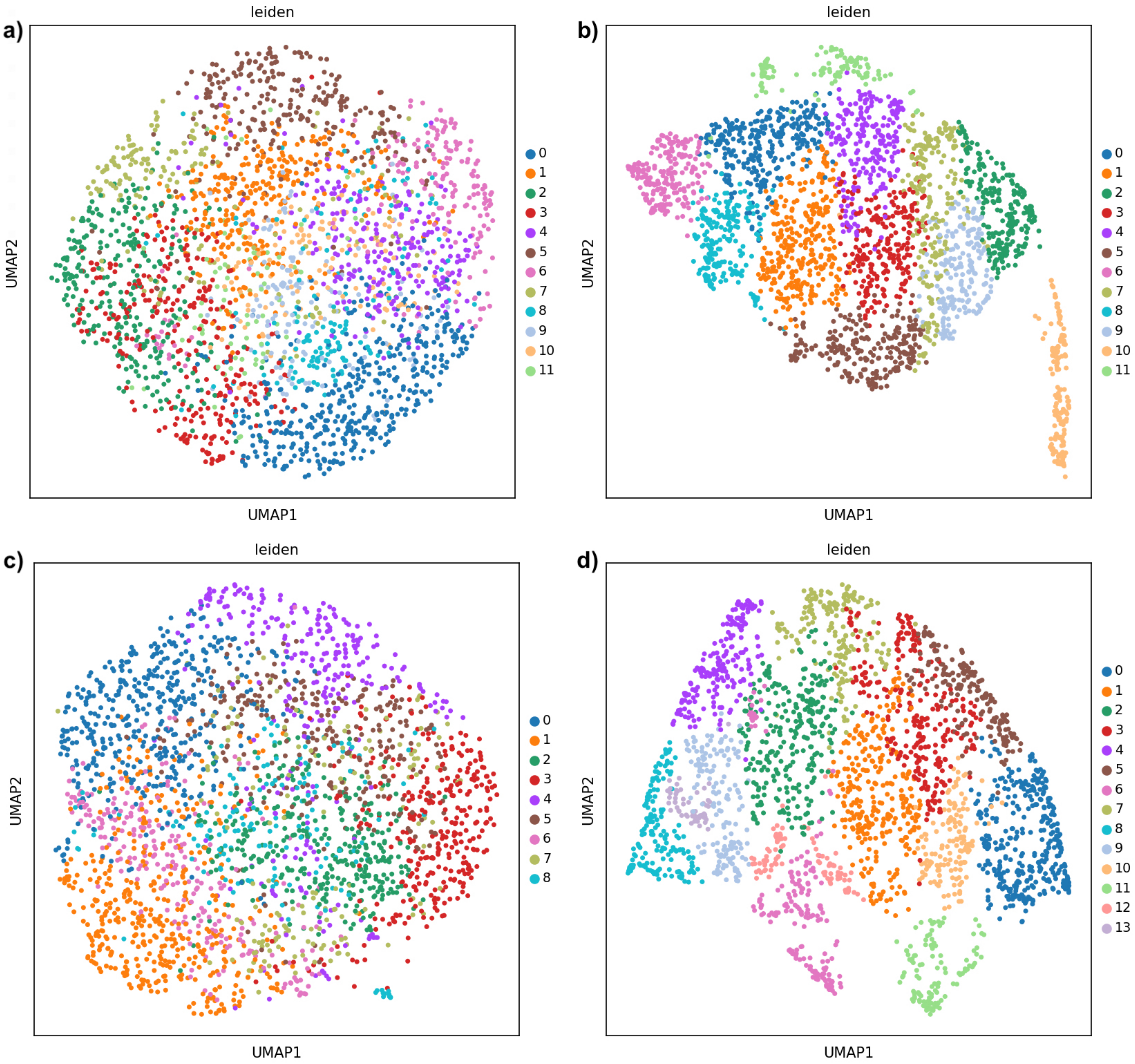

Figure 3.8: Novel clusters found from models with position b (**a**. VAEb, **b**. betaVAEHb) and position d (**c**. VAEd, **d**. betaVAEHd) by executing Leiden clustering algorithm upon the latent embeddings. By observation, the quality of the clusters found in PCVAE models with betaVAE base is better (more cohesive) than the clusters identified in models with vanilla VAE architecture.

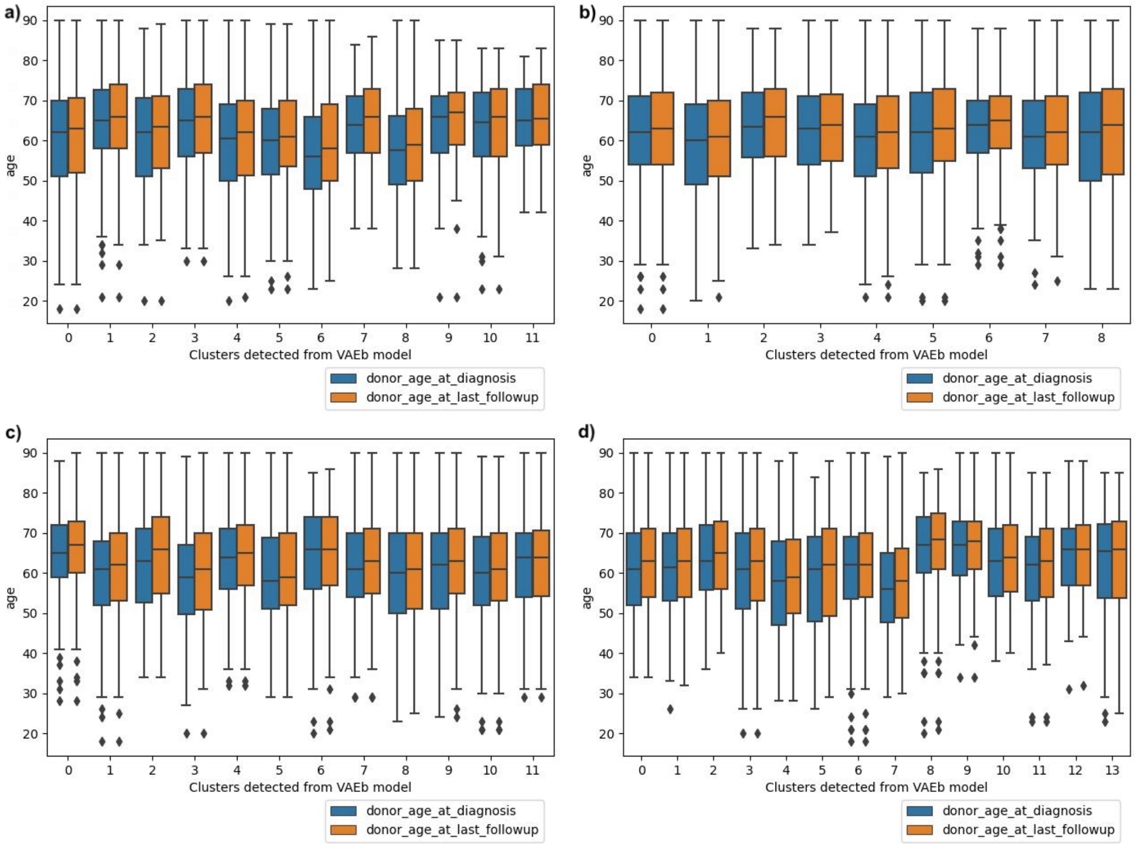

Figure 3.9: The box plot of donors' age at diagnosis and age at last follow-up for the novel clusters of donors from VAEb (**a**), VAEd (**b**), betaVAEHb (**c**), betaVAEHd (**d**) base PCVAE models.

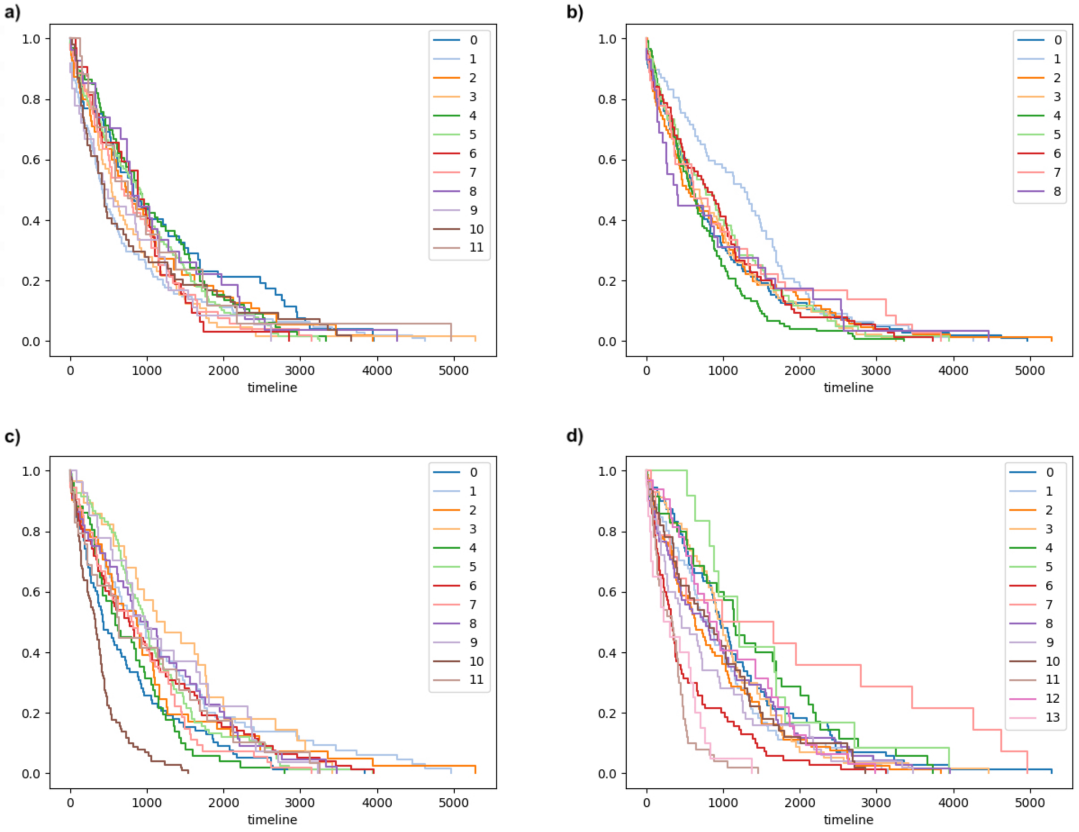

Figure 3.10: The Kaplan-Meier curves for the novel clusters of donors from VAEb (**a**), VAEd (**b**), betaVAEHb (**c**), betaVAEHd (**d**) base PCVAE models.

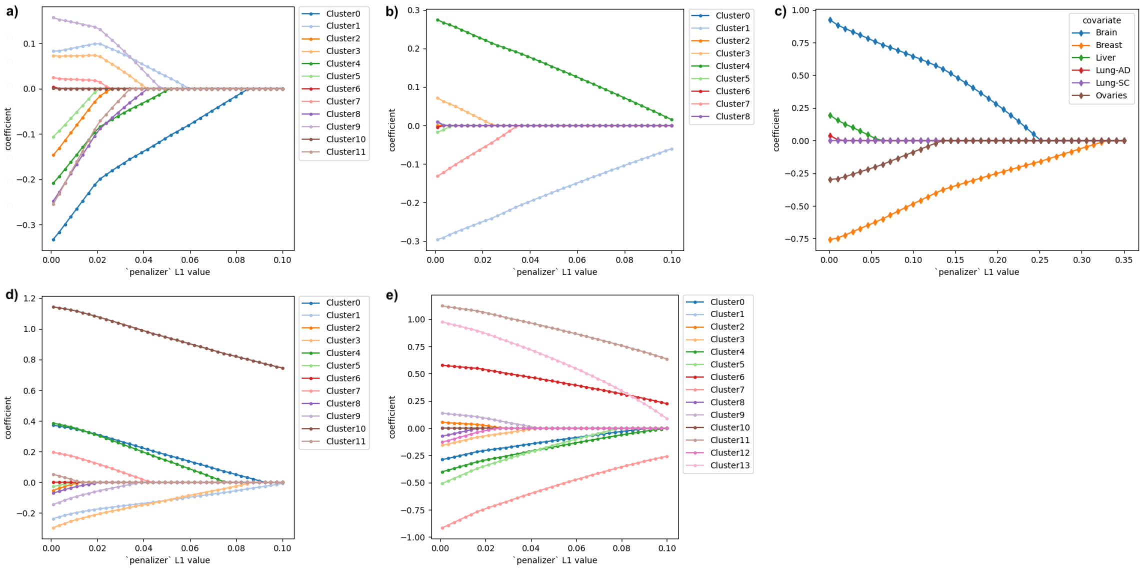

Figure 3.11: The coefficients versus L1 penalty generated by Python package `lifelines` from Penalised Cox-proportional hazards model. Clusters in the legend align with the ones detected respectively from VAEb (**a**), VAEd (**b**), betaVAEHb (**d**), betaVAEHd (**e**) base PCVAE models. X-axis indicates L1 penalization and y-axis represents the coefficient size. Fitted for 40 values of L1 lambda. Notably, panel **c** is a baseline of tissue covariate and markers are in a diamond shape.

Chapter 4

Conclusion

The problem we attempt to resolve is investigating the in-depth connections amongst heterogeneous cancers while controlling the dominant tissue effect. In the method section, we first verify that the signal is pronounced and difficult to eliminate. Next, we construct different autoencoder architectures to concentrate the genetic information from the high-dimensional space via latent feature extraction. In particular, we ensure that the signal persists with the latent representations for these base models. Ideally, the prominent tissue effect along with other signals that we are interested are indistinguishably encoded in the latent space through autoencoder variations. According to the visualization of the latent representations, we conclude that the signal of tissue remains dominant with the proper configurations and hyperparameters of the model. This validates the effectiveness of autoencoders and provides opportunities to extract and eradicate the tissue-related effect. By comparing the UMAP plots Figure 3.2, we observe that `Adam` optimizer displays proficiency in achieving rapid loss convergence and generating cohesive clusters. In addition, we conclude that a larger β coefficient yields a more disentangled space illustrated by Figure 3.3 and Figure 3.4. In terms of the PCAE model, the tissue effect has been reinforced instead of being mitigated in the latent space. We postulate this is caused by the regulation imposed by the cross entropy loss associated with the predictor. The tissue information is further condensed in the latent space instead of capturing such effect by the predictor. Due to the flexibility and high interpretability of VAE model, the PCVAE models demonstrate competence in eliminating the tissue effect by introducing a control variable ν. The location where we incept ν in the model does not affect the clustering much while the form of objective functions does. Both PCVAE models, which utilize a VAE base and a betaVAEH base, exhibit significant efficacy in eliminating the tissue signal. betaVAEH models produce high-quality novel clusters according to Calinski Harabasz score and Davies-Bouldin index. Overall, among all measures, betaVAEHb outperforms other models in terms of internal index criteria. In terms of entropy, the models with VAE base exterminate the tissue signal more thoroughly than the ones with the betaVAE base. We also conclude that Cluster 10 found from the betaVAEHd model and Cluster 11,6,7 exhibit equivalently signal intensity as Overay, Brain and Breast cancers underpinned by the Cox's proportional hazards model.

Chapter 5

Discussion

While in this thesis we only investigate a standard normal prior for VAE, there are a number of more sophisticated posteriors or priors that satisfy a variety of objectives. Specifically, Dilokthanakul et al. proposed a VAE model with a mixture of Gaussians prior to result in better disentanglement in the latent space, which has been implemented and improved by estimating the prior with a mixture of posteriors in a subsequent study[33]. This technique has been successfully deployed on single cell ATAC-SEQ RNA sequencing data[38]. On the other hand, there exist methodologies for devising adaptable posterior distributions that have gained significant popularity. They are also categorized within the family of deep generative model, comprising normalizing flows[26], auto-regressive flows[24][39] and an inverse version[18]. In addition, there are other variants of integrative VAES such as Mix-Modal VAE and Hierarchical VAE that have been utilized in the integration of heterogeneous cancer data types (multi-omics analysis)[30]. Nevertheless, these variants place weights on the refined orchestration amongst various VAE architectures instead of improving individual ones. Another observation is that the effectiveness of VAE model can be severely affected by the minor perturbation of the parameters. This inspires us to attempt multiple initialization strategies before the training stage of the PCVAE models and select the parameters that lead to the highest ELBO value.

An immediate continuation of the current work presented is incorporating the entire RNA-seq data from all available cancer types in ICGC portal and checking if the novel clusters and the assessment scores (Silhouette score and the Variance Ratio Criterion) alter. Moreover, inspired by transfer learning [42], we could extend and generalize our model in the future by feeding cancer data with heterogeneous types to our model (i.e. CNV, somatic mutation data, DNA methylation, structural somatic mutations, protein expression, etc). Alternatively, the model can be trained as a time series where we train the model with the initial type of data and use another type of data to feed the pre-trained model sequentially so on and forth. If the training data encompasses a comprehensive range of cancer-related aspects, the resulting model has the potential to effectively extract tissue effects on multiple granular levels, thereby enhancing its competence as a target effect extractor. According to the central dogma, genetic information will only flow in the direction from DNA, to RNA, to protein. Therefore, a compact dataset incorporating all these genetic expressions is likely to form a complete gene feature signal extractor since these molecules above encrypt the entire human genome information. Assuming that a generalized tissue effect filter can be established for any genetic profile, it will provide a fertile field for cancer research and may eventually unravel a deeper connection among heterogeneous cancers. It is worth noting that the tissue effect can be extended beyond primary sites to any semantically meaningful biomarker or biological effect. The survival analysis of patients could be inaccurate since the bulk RNA-seq data for each patient incorporates several confounding factors including batch effect. To analyze the biological significance of each cluster, it is vital to devise a controlled experiment to monitor cancer's status at a molecular scale.

Bibliography

[1] L. A. Aaltonen, F. Abascal, A. Abeshouse, H. Aburatani, D. J. Adams, N. Agrawal, K. S. Ahn, S.-M. Ahn, H. Aikata, R. Akbani, K. C. Akdemir, H. Al-Ahmadie, S. T. Al-Sedairy, F. Al-Shahrour, M. Alawi, M. Albert, K. Aldape, L. B. Alexandrov, et al. Pan-cancer analysis of whole genomes. *Nature*, Jan. 2023. doi:10.1038/s41586-022-05598-w. URL https://doi.org/10.1038/s41586-022-05598-w. → page 2

[2] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. 2021. doi:10.48550/ARXIV.2103.04922. URL https://arxiv.org/abs/2103.04922. → page 8

[3] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. 2015. doi:10.48550/ARXIV.1511.06349. URL https://arxiv.org/abs/1511.06349. → page 9

[4] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae, 2018. URL https://arxiv.org/abs/1804.03599. → pages 12, 13

[5] C. Calabrese, N. R. Davidson, D. Demircioğlu, N. A. Fonseca, et al. Genomic basis for RNA alterations in cancer. *Nature*, Jan. 2023. doi:10.1038/s41586-022-05596-y. URL https://doi.org/10.1038/s41586-022-05596-y. → page 2

[6] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 1974. doi:10.1080/03610927408827101. URL https://doi.org/10.1080/03610927408827101. → page 21

[7] I. Dagogo-Jack and A. T. Shaw. Tumour heterogeneity and resistance to cancer therapies. *Nature Reviews Clinical Oncology*, 15(2):81–94, Nov. 2017. doi:10.1038/nrclinonc.2017.166. URL https://doi.org/10.1038/nrclinonc.2017.166. → page 1

[8] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi:10.1109/TPAMI.1979.4766909. → page 21

[9] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders, 2016. URL https://arxiv.org/abs/1611.02648. → pages 9, 31

[10] F. Duffhauss, N. A. Vien, H. Ziesche, and G. Neumann. Fusionvae: A deep hierarchical variational autoencoder for rgb image fusion, 2022. URL https://arxiv.org/abs/2209.11277. → page 9

[11] J. Fan, F. Han, and H. Liu. Challenges of big data analysis. *National Science Review*, 1(2): 293–314, Feb. 2014. doi:10.1093/nsr/nwt032. URL https://doi.org/10.1093/nsr/nwt032. → page 4

[12] G. K. Geiss, R. E. Bumgarner, B. Birditt, T. Dahl, N. Dowidar, D. L. Dunaway, H. P. Fell, S. Ferree, R. D. George, T. Grogan, J. J. James, M. Maysuria, J. D. Mitton, P. Oliveri, J. L. Osborn, T. Peng, A. L. Ratcliffe, P. J. Webster, E. H. Davidson, L. Hood, and K. Dimitrov. Direct multiplexed measurement of gene expression with color-coded probe pairs. *Nature Biotechnology*, 26(3):317–325, Feb. 2008. doi:10.1038/nbt1385. URL https://doi.org/10.1038/nbt1385. → page 5

[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html. → page 18

[14] I. N. Grabski, K. Street, and R. A. Irizarry. Significance analysis for clustering with single-cell RNA-sequencing data. Aug. 2022. doi:10.1101/2022.08.01.502383. URL https://doi.org/10.1101/2022.08.01.502383. → page 15

[15] 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 *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl. → pages 12, 13

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114. → pages viii, 9, 11

[17] D. P. Kingma and M. Welling. An introduction to variational autoencoders. 2019. doi:10.48550/ARXIV.1906.02691. URL https://arxiv.org/abs/1906.02691. → pages 10, 11, 41

[18] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improving variational inference with inverse autoregressive flow, 2016. URL https://arxiv.org/abs/1606.04934. → page 31

[19] J. Kishore, M. Goel, and P. Khanna. Understanding survival analysis: Kaplan-meier estimate. *International Journal of Ayurveda Research*, 1(4):274, 2010. doi:10.4103/0974-7788.76794. URL https://doi.org/10.4103/0974-7788.76794. → pages 3, 21

[20] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, Nov. 2018. doi:10.1038/s41592-018-0229-2. URL https://doi.org/10.1038/s41592-018-0229-2. → page 8

[21] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen. Deep learning for deepfakes creation and detection: A survey. 2019. doi:10.48550/ARXIV.1909.11573. URL https://arxiv.org/abs/1909.11573. → page 8

[22] M. Palazzo, P. Beauseroy, and P. Yankilevich. A pan-cancer somatic mutation embedding using autoencoders. *BMC Bioinformatics*, 20(1), Dec. 2019. doi:10.1186/s12859-019-3298-z. URL https://doi.org/10.1186/s12859-019-3298-z. → page 7

[23] X. Pan, X. Hu, Y.-H. Zhang, L. Chen, L. Zhu, S. Wan, T. Huang, and Y.-D. Cai. Identification of the copy number variant biomarkers for breast cancer subtypes. *Mol. Genet. Genomics*, 294(1):95–110, Feb. 2019. → page 1

[24] G. Papamakarios, T. Pavlakou, and I. Murray. Masked autoregressive flow for density estimation, 2017. URL https://arxiv.org/abs/1705.07057. → page 31

[25] J. S. Parker, M. Mullins, M. C. U. Cheang, S. Leung, D. Voduc, T. Vickery, S. Davies, C. Fauron, X. He, Z. Hu, J. F. Quackenbush, I. J. Stijleman, J. Palazzo, J. S. Marron, A. B. Nobel, E. Mardis, T. O. Nielsen, M. J. Ellis, C. M. Perou, and P. S. Bernard. Supervised risk predictor of breast cancer based on intrinsic subtypes. *J. Clin. Oncol.*, 27(8):1160–1167, Mar. 2009. → pages 1, 5

[26] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows, 2015. URL https://arxiv.org/abs/1505.05770. → page 31

[27] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models, 2014. URL https://arxiv.org/abs/1401.4082. → page 11

[28] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi:https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257. → page 21

[29] L. Ruthotto and E. Haber. An introduction to deep generative modeling, 2021. URL https://arxiv.org/abs/2103.05180. → page 8

[30] N. Simidjievski, C. Bodnar, I. Tariq, P. Scherer, H. A. Terre, Z. Shams, M. Jamnik, and P. Liò. Variational autoencoders for cancer data integration: Design principles and computational practice. *Frontiers in Genetics*, 10, Dec. 2019. doi:10.3389/fgene.2019.01205. URL https://doi.org/10.3389/fgene.2019.01205. → pages 9, 11, 31

[31] K. Tan, W. Huang, J. Hu, and S. Dong. A multi-omics supervised autoencoder for pan-cancer clinical outcome endpoints prediction. *BMC Medical Informatics and Decision Making*, 20(S3), July 2020. doi:10.1186/s12911-020-1114-3. URL https://doi.org/10.1186/s12911-020-1114-3. → page 7

[32] The International Cancer Genome Consortium. International network of cancer genome projects. *Nature*, 464(7291):993–998, Apr. 2010. doi:10.1038/nature08987. URL https://doi.org/10.1038/nature08987. → pages 2, 4

[33] J. M. Tomczak and M. Welling. Vae with a vampprior, 2017. URL https://arxiv.org/abs/1705.07120. → page 31

[34] V. A. Traag, L. Waltman, and N. J. van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), Mar. 2019. doi:10.1038/s41598-019-41695-z. URL https://doi.org/10.1038/s41598-019-41695-z. → pages 15, 21

[35] G. P. Way and C. S. Greene. Evaluating deep variational autoencoders trained on pan-cancer gene expression, 2017. URL https://arxiv.org/abs/1711.04828. → page 7

[36] G. P. Way and C. S. Greene. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pac. Symp. Biocomput.*, 23:80–91, 2018. → page 9

[37] Z. Wu, X. Long, S. Y. Tsang, T. Hu, J.-F. Yang, W. K. Mat, H. Wang, and H. Xue. Genomic subtyping of liver cancers with prognostic application. *BMC Cancer*, 20(1):84, Jan. 2020. → page 2

[38] L. Xiong, K. Xu, K. Tian, Y. Shao, L. Tang, G. Gao, M. Zhang, T. Jiang, and Q. C. Zhang. SCALE method for single-cell ATAC-seq analysis via latent feature extraction. *Nature*

Communications, 10(1), Oct. 2019. doi:10.1038/s41467-019-12630-7. URL https://doi.org/10.1038/s41467-019-12630-7. → page 31

[39] F. Zhan, Y. Yu, R. Wu, J. Zhang, K. Cui, C. Zhang, and S. Lu. Auto-regressive image synthesis with integrated quantization, 2022. URL https://arxiv.org/abs/2207.10776. → pages 9, 31

[40] 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)*. IEEE, Nov. 2019. doi:10.1109/bibm47256.2019.8983228. URL https://doi.org/10.1109/bibm47256.2019.8983228. → page 7

[41] Z. Zhang and M. R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018. URL https://arxiv.org/abs/1805.07836. → page 14

[42] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning, 2019. URL https://arxiv.org/abs/1911.02685. → page 31

Appendix A

Supplementary Materials

A.1 KL DIVERGENCE and its properties

Let's define the Kullback-Leibler divergence (KL DIVERGENCE) formally. Given two probability distributions p, q defined on the same sample space $\mathcal{X}$,

$$D_{KL}(p(x) \mid\mid q(x)) = \sum_{x \in \mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

if p, q are discrete and

$$D_{KL}(p(x) \mid\mid q(x)) = \int_{\mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

if p, q are continuous.

First, KL DIVERGENCE is not a symmetric measure. This is straight-forward to show. Let's equate $D_{KL}(p(x) \mid\mid q(x))$ with $D_{KL}(q(x) \mid\mid p(x))$ and we obtain

$$p(x)(\log p(x) - \log q(x)) = -q(x)(\log p(x) - \log q(x))$$

in either scenarios above. Based on the axiom of probability measure $0 \leq p(x) \leq 1$, we have shown that this is not a symmetric measure unless $p = q$.

Second, KL DIVERGENCE is always non-negative. Gibbs inequality is a direct proof for this lemma. Here, we employ Jensen's inequality to do a quick proof.

$$\begin{aligned}
D_{KL}(p(x) \mid\mid q(x)) &= \mathbb{E}_p\left[\log\left(\frac{p(x)}{q(x)}\right)\right] && \text{(A.1)}\\
&= \mathbb{E}_p\left[-\log\left(\frac{q(x)}{p(x)}\right)\right] && \text{(A.2)}\\
&\geq -\log\left(\mathbb{E}_p\left[\frac{q(x)}{p(x)}\right]\right) && \text{(A.3)}\\
&\geq 0 \quad \square && \text{(A.4)}
\end{aligned}$$

Equation (A.3) is the Jensen's inequality and we use a fact that $f(x) = -\log(x)$ is a convex function.

A.2 Derivation of ELBO

Provided the second property in A.2, we evaluate the following KL DIVERGENCE. In this context, we assume that both distributions p, q are continuous.

$$\begin{aligned}
D_{KL}(q_\psi(z \mid x) \mid\mid p_\phi(z \mid x)) &= -\int q_\psi(z \mid x) \log \frac{p_\phi(z \mid x)}{q_\psi(z \mid x)} dz && \text{(A.5)}\\
&= -\int q_\psi(z \mid x) \log \frac{p_\phi(x \mid z) p_\phi(z)}{q_\psi(z \mid x) p_\phi(x)} dz && \text{(A.6)}\\
&= -\int q_\psi(z \mid x) \log \frac{p_\phi(x \mid z) p_\phi(z)}{q_\psi(z \mid x)} dz + \int q_\psi(z \mid x) \log p_\phi(x) dz && \text{(A.7)}\\
&= -\int q_\psi(z \mid x) \log \frac{p_\phi(x \mid z) p_\phi(z)}{q_\psi(z \mid x)} dz + \log p_\phi(x) \int q_\psi(z \mid x) dz && \text{(A.8)}\\
&= -\int q_\psi(z \mid x) \log \frac{p_\phi(x \mid z) p_\phi(z)}{q_\psi(z \mid x)} dz + \log p_\phi(x) && \text{(A.9)}\\
&\geq 0 && \text{(A.10)}
\end{aligned}$$

Equation (A.6) holds because of Baye's rule and (A.9) is the non-negative property of KL DIVERGENCE. After rearranging the terms, we arrive at the ELBO in the following form.

$$\begin{aligned}
\log p_\phi(x) \geq \int q_\psi(z \mid x) \log \frac{p_\phi(x \mid z) p_\phi(z)}{q_\psi(z \mid x)} dz && \text{(A.11)} \\
= \mathbb{E}_{q_\psi(z|x)} \left[\log \frac{p_\phi(x, z)}{q_\psi(z \mid x)} \right] && \text{(A.12)} \\
= - \int q_\psi(z \mid x) \log \frac{q_\psi(z \mid x)}{p_\phi(z)} dz + \int q_\psi(z \mid x) \log p_\phi(x \mid z) dz && \text{(A.13)} \\
= \mathbb{E}_{q_\psi(z|x)} \left[\log(p_\phi(x \mid z)) \right] - D_{KL}(q_\psi(z \mid x) \mid\mid p_\phi(z)) && \text{(A.14)} \\
= \mathcal{L}(x, \psi, \phi) && \text{(A.15)}
\end{aligned}$$

Notice that expressions (A.12) and (A.14) are essentially ELBO in different forms.

A.3 KL DIVERGENCE of Gaussians

Let's consider the general case where we want to compute the KL DIVERGENCE of two multivariate gaussians

$$p(x) = \mathcal{N}_k(x; \boldsymbol{\mu}, \boldsymbol{\Sigma}),\ q(x) = \mathcal{N}(x; \mathbf{0}, \boldsymbol{I}_k).$$

In the explicit form, the multivariate Gaussian distribution has the following density $p(x)$

$$p(x) = \frac{1}{(2\pi)^{k/2} [\det(\Sigma)]^{1/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \boldsymbol{\Sigma}^{-1} (x - \mu) \right)$$

if the symmetric covariance matrix $\mathbf{\Sigma}$ is positive definite.
According to the expectation form (A.1) of KL DIVERGENCE, we can write it as

$$
\begin{aligned}
D_{KL}(p(x) \mid\mid q(x)) &= \mathbb{E}_p \left[\log p(x) - \log q(x)\right] && \text{(A.16)}\\
&= \frac{1}{2}\mathbb{E}_p \left[-\log\det\mathbf{\Sigma} - (x-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(x-\boldsymbol{\mu}) + x^T x\right] && \text{(A.17)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} + \frac{1}{2}\mathbb{E}_p \left[-(x-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(x-\boldsymbol{\mu}) + x^T x\right] && \text{(A.18)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} + \frac{1}{2}\mathbb{E}_p \left[-tr((x-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(x-\boldsymbol{\mu})) + tr(x^T x)\right] && \text{(A.19)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} + \frac{1}{2}\mathbb{E}_p \left[-tr(\mathbf{\Sigma}^{-1}(x-\boldsymbol{\mu})(x-\boldsymbol{\mu})^T) + tr(xx^T)\right] && \text{(A.20)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} - \frac{k}{2} + \frac{1}{2}\mathbb{E}_p \left[tr(xx^T)\right] && \text{(A.21)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} - \frac{k}{2} + \frac{1}{2}\mathbb{E}_p \left[tr((x-\boldsymbol{\mu})(x-\boldsymbol{\mu})^T + x\boldsymbol{\mu}^T + \boldsymbol{\mu}x^T - \boldsymbol{\mu}\boldsymbol{\mu}^T)\right] \\
& && \text{(A.22)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} - \frac{k}{2} + \frac{1}{2}tr(\mathbf{\Sigma} + 2\boldsymbol{\mu}\boldsymbol{\mu}^T - \boldsymbol{\mu}\boldsymbol{\mu}^T) && \text{(A.23)}\\
&= -\frac{1}{2}\log\det\mathbf{\Sigma} - \frac{1}{2}k + \frac{1}{2}tr(\mathbf{\Sigma}) + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} && \text{(A.24)}
\end{aligned}
$$

Note that the trace of a value in $\mathbb{R}$ remains the same value (the identity map). Here, I take advantage of the property of trace for the quadratic form where we can commute matrices within the trace and it will generate the identical outcome, i.e.

$$tr(AB) = tr(BA)$$

holds for any matrices A, B. Another fact I use is derived from the linearity of expectation. The trace operator and the expectation operator are exchangeable, i.e.

$$tr(\mathbb{E}(A)) = \mathbb{E}(tr(A))$$

for an arbitrary matrix A comprised of random variables. A possible proof provided online.

In the Section 2.2.2, we select the encoder to be simple factorized Gaussian [17]. Hence, the covariance matrix becomes diagonal $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_k^2)$. Therefore, we derive the KL DIVERGENCE written in that section.

$$D_{KL}(p(x) \mid\mid q(x)) = -\frac{1}{2}\log\det\boldsymbol{\Sigma} - \frac{1}{2}k + \frac{1}{2}tr(\boldsymbol{\Sigma}) + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} \tag{A.25}$$

$$= -\frac{1}{2}\log\prod_{i=1}^{k}\sigma_i^2 - \frac{1}{2}k + \frac{1}{2}\sum_{i=1}^{k}\sigma_i^2 + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} \tag{A.26}$$

$$= -\frac{1}{2}\sum_{t=1}^{k}(1 + \log(\sigma_t^2(\psi, x)) - \sigma_t^2(\psi, x) - \mu_t^2(\psi, x)) \quad \square \tag{A.27}$$

A.4 Plots and table

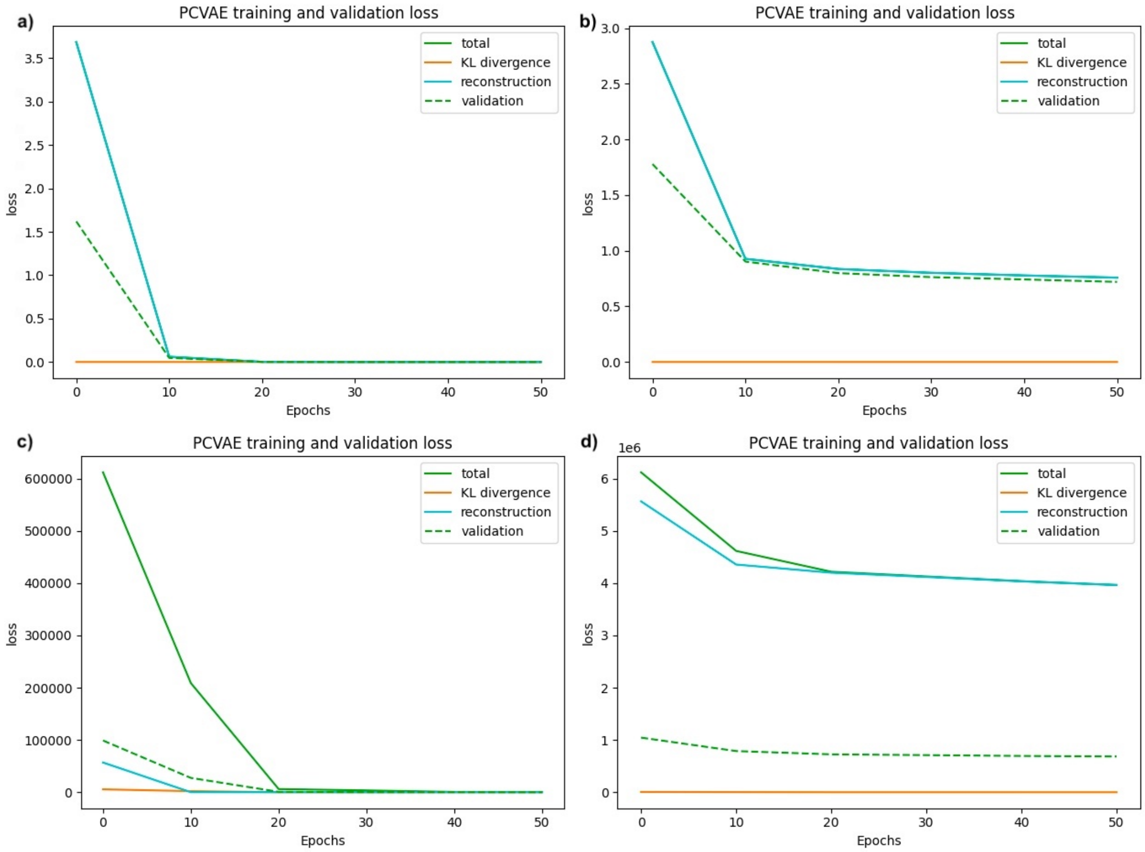

Figure A.1: Trace plots of loss evaluated from PCVAE models with VAEb (**c**), VAEd (**d**), betaVAEHb (**a**), betaVAEHd (**b**) base.

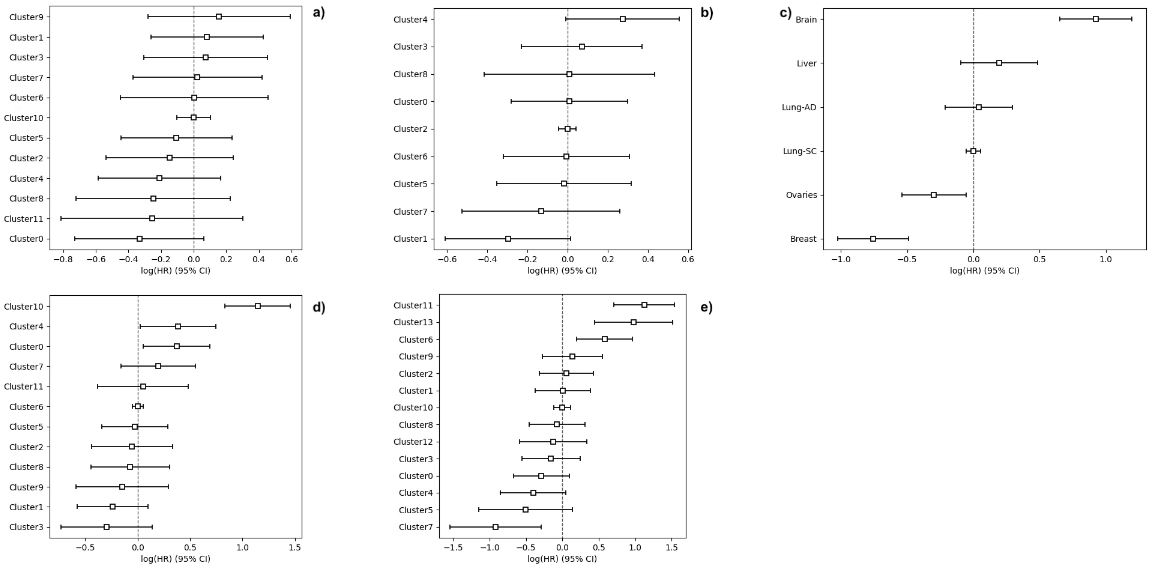

Figure A.2: A visual representation of the log hazard ratios, including their standard errors and magnitudes. Using `lifelines` package, fitted coefficients are against Cox-proportional harzards model with L1 penalty. Set `penalizer=0.001`. The error bar represents the 95% confidence interval for each fitted coefficient within each category. The cluster labels are represented as one-hot encoding matrix derived from PCVAE models with VAEb (**a**), VAEd (**b**), betaVAEHb (**d**), betaVAEHd (**e**) base. The plot **e** delineates the log hazard ratios against ground truth primary site labels, which acts as a baseline.

Table A.1: Count of some evident clusters and their primary site

Model Name	Primary Site	Cluster	Count
betaVAEHb	Brain	Cluster10	153
betaVAEHb	Lung-SC	Cluster10	2
betaVAEHb	Liver	Cluster10	2
betaVAEHd	Brain	Cluster6	74
betaVAEHd	Breast	Cluster6	2
betaVAEHd	Liver	Cluster6	97
betaVAEHd	Lung-AD	Cluster6	4
betaVAEHd	Lung-SC	Cluster6	6
betaVAEHd	Brain	Cluster7	4
betaVAEHd	Breast	Cluster7	156
betaVAEHd	Liver	Cluster7	2
betaVAEHd	Lung-AD	Cluster7	11
betaVAEHd	Ovaries	Cluster7	4
betaVAEHd	Brain	Cluster11	63
betaVAEHd	Breast	Cluster11	7
betaVAEHd	Liver	Cluster11	34
betaVAEHd	Lung-AD	Cluster11	6
betaVAEHd	Lung-SC	Cluster11	11
betaVAEHd	Ovaries	Cluster11	1