Autoencoder techniques for survival analysis on renal cell carcinoma

Íñigo Sanz Ilundain[1]

[1]isanzi22@student.aau.dk

Renal cell carcinoma (RCC) is the most common type of kidney cancer, accounting for 90% of all cases. Understanding RCC's onset, progression, and response to treatments is critical to improving patient outcomes. Survival Analysis, a branch of statistics, provides essential tools for examining time-to-event data, which is pivotal in these efforts. However, the high-dimensional nature of medical data, particularly transcriptomic data, poses significant challenges for traditional regression methods.

In this study, we utilized data from the JAVELIN Renal 101 trial, a randomized phase 3 trial comparing the efficacy of two treatment arms: an immunotherapy regimen (avelumab + axitinib) and a tyrosine kinase inhibitor (TKI) named sunitinib. This trial aims to measure how long the cancer does not progress after treatment. Each patient's data includes detailed transcriptomic profiles, which quantify RNA transcripts produced from each gene, reflecting gene expression levels in tissues. This information is crucial because certain gene expressions are closely linked to cancer development.

Patients are assigned a Progression-Free Survival (PFS) value, indicating the duration they remain without cancer progression, and a censoring value, which notes if they left the study early for reasons unrelated to the primary endpoint. The high-dimensionality of this dataset, combined with a relatively small sample size, necessitates advanced techniques for dimensionality reduction. To address this, we employed autoencoders to compress the data into latent, meaningful features.

We utilized several metrics commonly employed in Survival Analysis to evaluate our models. These include the Concordance Index (C-index), which measures rank correlation between predicted risk scores and event times, and the time-dependent area under ROC, which evaluates the true positive rate against the true negative rate at various time points. To derive risk scores, we applied the COX Proportional Hazards model, a semi-parametric method that assumes the effect of features on the hazard is constant over time. However, as this model cannot directly estimate survival functions, we used Breslow's estimator, a non-parametric approach, to estimate these functions and predict PFS.

Given the complexity and perceived opacity of neural networks, especially autoencoders, we incorporated interpretability into our models by analyzing the mutual information between the original genes and the latent representations. This step helps to clarify which genes are most represented in the latent variables, making the models more transparent and their results more understandable.

Our preprocessing steps involved several key actions: removing genes expressed only in a small subset of patients, filtering out incomplete histological characteristics, and selecting genes specific for CCR. We normalized the transcriptomic data based on the highest expressed gene to ensure consistent scaling across the dataset.

We developed two types of autoencoders: the tabular autoencoder, which processes data in a tabular format and independently encodes and decodes features, and the graph autoencoder, which processes data through a graph where edges represent protein-to-protein interactions (PPI). For each autoencoder type, we tested multiple penalties:

- Denoising Penalty: Adds noise to input data to improve generalization.
- Sparse Penalty: Penalizes large activations to ensure meaningful feature extraction.
- Variational Penalty: Fits latent representations to a Gaussian distribution, making the model generative.
- Combined Penalties: Investigates the effects of combining denoising and sparse penalties.

The PPI network for the graph autoencoder was constructed using kidney-specific interactions, filtered to include only RCC-relevant genes. We performed thorough graph analysis to ensure a well-connected network without loose components.

Our results indicate that autoencoders can effectively compress high-dimensional data into latent features. The tabular autoencoder excels in data reconstruction because it treats genes independently. In contrast, the graph autoencoder, while constrained by the network's specifics, better captures the biological relevance of the data. Among the penalties, the denoising penalty was most effective for reconstruction, while the sparse penalty yielded the most meaningful latent features and the most accurate predictions. Variational autoencoders were less effective due to the constraints imposed by fitting to a specific distribution.

In terms of predictive performance, the graph autoencoder generally outperformed the tabular autoencoder, likely because it leverages significant biological interactions in the PPI network. The sparse penalty, in particular, identified meaningful representations, resulting in the most accurate predictions.

Our interpretability analysis highlighted the genes most frequently associated with the significant latent features identified by the COX model. Genes such as LRP2, NAT8, ACE2, CYP4A11, and EMX2 were found most frequently in our latent features, underscoring their potential importance in RCC.

This research demonstrates that autoencoders are powerful tools for handling high-dimensional medical data and can be effectively combined with statistical models to enhance data compression and regression tasks. The study also emphasizes the importance of selecting appropriate penalties based on the specific task and incorporating interpretability to better understand the models' results. While our focus was on RCC, the methodology can be extended to other medical fields, showcasing its versatility.

Future work could involve validating our results with a new cohort of patients, exploring other statistical methods such as parametric modeling, and enhancing interpretability and visualization techniques to gain deeper insights into the data. By continuing to refine these approaches, we can improve the accuracy and applicability of predictive models in various medical contexts, ultimately benefiting patient care and treatment outcomes.

Autoencoder techniques for survival analysis on renal cell carcinoma

Íñigo Sanz Ilundain[1]

[1]isanzi22@student.aau.dk

Abstract

Survival analysis heavily impacts the study of diseases by providing statistical methods and metrics to analyze time-to-event data, crucial in understanding disease progression and the effectiveness of treatments. However, in the medical domain, the data is often high-dimensional, complicating the regression of such methodologies. For this reason, in this work, we have focused on compressing the high-dimensionality found in the transcriptomic data of patients treated with an immunotherapy (avelumab + axitinib) and a TKI (sunitinib) into latent, meaningful features using autoencoders. We then applied a statistical methodology based on the COX Proportional Hazards model, a semi-parametric approach, combined with Breslow's estimator to determine the survival functions of the patients and predict each patient's Progression-Free Survival (PFS). We extensively analyzed different penalties as well as their combinations. Due to the nature of the transcriptomic data, we extended the model to accept not only tabular data but also its graph variant, where the edges represent protein-to-protein interactions between genes, which proved to be a more meaningful approach. Finally, since neural networks, and especially autoencoders, are often seen as black boxes, we worked on interpretability by identifying the mutual information between the genes in the original data and the representations of the latent features. This approach attempts to clarify which genes are most presented in which latent variables. Our results show that certain types of autoencoders are more relevant depending on the situation. To obtain accurate reconstruction, denoising autoencoders prove useful. To find meaningful representations of the data, the sparse variant is the best option. Moreover, these penalties can be combined to achieve both accurate representations and meaningful latent features. The interpretable models also suggested that genes such as LRP2 and ACE2 are highly related to renal cell carcinoma. We present this work as extensive research demonstrating the usefulness of autoencoders in high-dimensional problems.

1 - Introduction

Kidney cancer is the 12th most common type of cancer worldwide, accounting for 2.4% of all cancers, with over 330.000 new cases every year. Renal cell carcinoma (RCC) is the most common type of kidney cancer, representing over 90% of all cases [1, 2]. Early diagnosis of RCC is crucial for effective treatment and reducing mortality rates [3]. Unfortunately, the risk factors for the disease are not completely clear yet due to its complexity [4]. There exist several treatments for this type of cancer such as immunotherapies, which can be targeted towards certain cells of the immune system, and chemotherapies, which target all rapidly dividing cells.

In the treatment process, patients usually undergo clinical trials to evaluate the safety and efficacy of new therapies or drug combinations. These trials are critical for advancing medical knowledge and improving treatment options for future patients. During this procedure, doctors measure the Progression-Free Survival (PFS) of a patient, the amount of time for which the patient's condition has not reached a certain event, such as death. Moreover, when the patient's monitoring is cut short, they may get a censored PFS value, indicating that we do not know the exact time when the event occurred [5]. Survival analysis focuses on predicting the PFS of newly arrived patients based on available treatments.

Patients are mainly described by transcriptomic data, which are positive continuous variables representing the frequency of gene expression in their system. Gene expression profiling is an effective technique to measure the expression of genes in the cells, especially during the onset of cancer [6].

An effective solution to predict the survival of a patient for a specific treatment is to use statistical methods, such as regression models. These models can integrate a series of covariates and learn coefficients based on the importance of the features. However, these methods are often employed on low-dimensional data [7].

Due to the amount of genes in the human body, however, the data becomes high-dimensional, especially compared to the low number of patients (or samples) in the trial. An efficient way to handle high-dimensional data is through the use of dimensionality reduction. In this regard, machine learning offers a variety of useful methodologies, from which we can highlight autoencoders. Autoencoders compress data into lower-dimensional representations while retaining essential features, which can prove an interesting method for transcriptomic data. While PCA is an effective method of dimensionality reduction, in a low sample setting with complex data, where non-linear relationships are present, non-linear approaches are more suitable.

An important characteristic in transcriptomics is the interaction between genes. These are called protein-to-protein interactions (PPI) and they give information on which genes are related to eachother. These interactions can be represented in a network where genes are no longer independent from eachother and information flows through these interactions. Graph autoencoders can be assimilated with this type of data to make meaningful representations.

Aside from making predictions, machine learning models are often seen as a black box. No matter how precise a regression model is, it is crucial to know how it interpreted the data and made a certain choice. Therefore, we decided to analyze the latent representations obtained by the autoencoder through the use of mutual information, an information theory technique.

Current works have been elaborated following a similar path. On the one hand, regression analysis was employed using a semi-parametric model called COX Proportional Hazards for survival analysis of pancancer patients. The number of genes chosen for the work was low, as statistical models are sensitive to high-dimensional data [8]. On the other hand, other works used machine learning tools such as autoencoders on lung cancer survival scenarios, however only the denoising variant was employed, which also lacked interpretability [9].

As such, the main goal of this work is an exploratory and interpretable analysis of different autoencoders in predicting an accurate PFS for new patients treated with either avelumab and axitinib, an immunotherapy; or sunitinib, a tyrosine kinase inhibitor (TKI), a pharmaceutical drug targeted towards angiogenesis.

The structure of the presented work is as follows: in Section 2, we will provide a formal description of the problem we intend to address. Following that, in Section 4, we will elaborate on the methodology utilized to solve the problem. Section 5 will detail the results achieved through the application of the methodology, and in Section 6, we will discuss these findings. Lastly, Section 7 will draw conclusions from this work and emphasize the highlights of the report.

2 - Problem Definition

Patients participating in the current study are characterized by transcriptomic, clinical and histologic data. On the one hand, patients are characterized by their whole exome sequencing, a technique that indicates how expressed, or how frequent, a gene is in the patient. On the other hand, some measurements obtained clinically and histologically can also indicate the status of a patient, such as the size of the tumor before any treatment is applied. As such, let $\mathbf{X} \in \mathbb{R}^{N \times M}$, the dataset representing renal-cell carcinoma patients, where N is the number of patients, and M is a mixture of genetic features: the expression of each gene in each patient and clinical features. While clinical features vary in what they measure, the expression level of a gene g is always such that $g \geq 0$. The dataset is inherently high-dimensional feature-wise since the amount of features that characterize each patient surpasses the number of patients that are included within. Moreover, let the matrix $\mathbf{Y} \in \mathbb{R}^{N \times 2}$ regarding survival data, where each patient $n_i \in [0, N]$ holds a pair of values (y_i, δ_i), representing the patient's Progression Free Survival (PFS) and censoring value, respectively.

Definition 2.1 (Censoring Event for a patient) *Censoring in a patient, denoted by $\delta_i \in \{0, 1\}$, is a binary indicator reflecting whether the endpoint (event) has been observed. In the context of renal-cell carcinoma, $\delta_i = 1$ indicates that the i-th patient did not show progression of cancer during treatment.*

Definition 2.2 (Progression-Free Survival (PFS)) *The PFS for a patient, represented as $y_i \in \mathbb{R}$, measures the duration in months during which no cancer progression is observed. This metric is influenced by the censoring status δ_i of the i-th patient, as it considers potential future events for those not reaching the endpoint during the study and can be interpreted as:*

$$y_i = \begin{cases} t & \textit{if } \delta_i = 1 \\ c & \textit{if } \delta_i = 0 \end{cases}$$

where t is the event time *when an event occurred, and c the* time of censoring.

These definitions are fundamental in survival analysis, as they provide insights into whether and when a patient experiences the key event of interest. The goal is that, given a new patient j with data X_j, we want to predict the PFS value in Y_j.

Our models will utilize these data points to forecast outcomes, assessed using various metrics including the *Concordance Index (C-index)* and the *Area Under the receiver operating characteristic Curve (AUC)*.

Definition 2.3 (Concordance Index (C-index)) *The C-index measures the rank correlation between predicted risk scores $\hat{f}$ and event times y. Let the comparable pairs C_A and the concordant pairs C_B for any pair of samples (i, j) be such that:*

$$C_A = \{(i,j) \mid y_j > y_i \wedge \delta_i = 1\}$$
$$C_B = \{(i,j) \in C_A \mid \hat{f}_i > \hat{f}_j \wedge y_j > y_i\}$$

where C_A represents the pairs for which we know at least there is one sample with a smaller observed event time, and C_B those pairs in C_A whose assigned risks correlate with the survival times of the samples. Then, the C-index is obtained by computing the ratio of their cardinalities:

$$C_{index} = \frac{|C_B|}{|C_A|}$$

Definition 2.4 (Time-dependent area under ROC) *The time-dependent ROC is an extended version of the original ROC, used to evaluate the discriminative capability of a model at a specific time t, given the predicted risks $\hat{f}$. Given their survival times y_i, we group patients into cumulative $y_i \leq t$ and dynamic $y_i > t$ cases. We then compute the true positive rate over the cumulative cases, and the true negative rate over the dynamic cases, using $\hat{f}_i \in \hat{f}$ as threshold. We obtain the time-dependent ROC, from which we can compute its area under the curve to obtain an overview of the model's performance at instant t.*

Both the C-index and the AUC are metrics for evaluating survival analysis models. The C-index gauges overall performance, while the AUC identifies specific time points of predictive weakness.

With these foundational concepts, we explore the following challenge.

Problem 2.1 (Accurate PFS prediction for RCC data) *Given a dataset $\mathcal{X}$ characterized by low sample size, censoring and high dimensionality with renal-cell carcinoma data, our objective is to derive meaningful low-dimensional embeddings through the use of autoencoders that report on accurate PFS predictions and satisfactory C-index and time-dependent area under ROC.*

3 - Background

In this section we go over a series of autoencoders, both differing in the type of data they compress, tabular and graph data, as well as different penalties such as the denoising penalty to improve the generalization, or the sparse penalty to obtain meaningful representations. We also introduce statistical methods such as COX Proportional Hazards and Breslow's estimator which were used to predict the risks and PFS of the patients.

3.1 Autoencoders

An autoencoder is a neural network designed to represent high-dimensional data in a lower-dimensional space. It operates under the assumption that high-dimensional data reside in or near a low-dimensional manifold within the input space [10]. Autoencoders consist of two components: an encoder f_e that compresses the data $\mathbf{x} \in \mathbb{R}^d$ into a low-dimensional representation $\mathbf{z} \in \mathbb{R}^p$ (where $d > p$), and a decoder f_d that performs a reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^d$ from $\mathbf{z}$. These components are trained to minimize a loss function that measures the quality of the reconstruction [11, p. 677]:

$$\mathcal{L}(f_e, f_d, \mathbf{x}) = (\hat{\mathbf{x}} - \mathbf{x})^2 = (f_d(f_e(\mathbf{x})) - \mathbf{x})^2$$

Standard autoencoders can be extended to regularized variants in order to achieve specific tasks such as a better generalization or more meaningful representations. [12]

The first such variant is the *denoising* autoencoder [13], which introduces a controlled amount of noise $\epsilon \sim \mathcal{N}(0,1)$ to the input, yielding a noisy version $\widetilde{\mathbf{x}} = \mathbf{x} + \epsilon$. The reconstruction loss is then adapted to accommodate the noisy data, in order to minimize:

$$\mathcal{L}(f_e, f_d, \mathbf{x}) = (f_d(f_e(\widetilde{\mathbf{x}})) - \mathbf{x})^2$$

This process enhances the autoencoder's ability to generalize from variations of the input data [13].

The second variant is the *sparse* autoencoder [14], which aims to reduce as much as possible the magnitude of the weights used in the neural network. This variant can be applied in two different ways: by using $\ell1/\ell2$ regularization or KL divergence. While the former are straightforward and drive the weights towards zero, the latter encourages the activation distribution to follow a certain distribution of our choice, becoming the better option. [15] The KL divergence penalizes the divergence between the activations of the hidden layers q_k and a target distribution ρ. This penalty can be written as:

$$\Omega(x, f_e) = \lambda \cdot \sum_k \mathrm{KL}(\rho \| q_k)$$

The third type of regularization is the *variational* autoencoder [16]. This unique variant regularizes the latent space, making it continuous and making the model generative. Instead of mapping the input data to a latent space, the variational autoencoder finds the underlying distribution of the data, and then samples from it using the reparameterization trick. [16] The penalty term that allows for this regularization is called the divergence loss and can be written as

$$\Omega(z, q, p) = \mathrm{KL}(q(z) \| p(z|x))$$

where $q(z)$ is the approximate posterior distribution and $p(z|x)$ is the true posterior distribution.

These variants can be combined to further enhance latent representations [17].

Additionally, autoencoders are applicable to graph-structured data. Graph Autoencoders (GAE) take a graph $G = (V, E)$, with V as nodes and E as edges, and $A \in \mathbb{R}^{N \times N}$ as the adjacency matrix. The aforementioned regularization techniques are also viable in GAEs, targeting either the graph's features, its adjacency matrix, or both [18].

3.2 Statistical approaches

Utilizing statistical methods allows us to make inferences and predictions regarding the survival of new patients. This survival is characterized by a survival function $S(t)$, which models the probability that an event has not occurred by time t, and a hazard function $h(t)$, the instantaneous likelihood of an event occurring at time t since no event has occurred before time t. The choice of method depends on the distribution of the data. Given the complex nature of RCC data and the lack of assumptions we can make about its distribution, semiparametric models, particularly the COX Proportional Hazards (PH) model, are suitable choices.

In the context of the COX PH model, the survival and hazard functions are defined as follows:

$$S(t) = S_0(t)^{\exp(X \cdot \beta)}$$
$$h(t|X) = h_0(t) \exp(X \cdot \beta)$$

where β is an array of coefficients associated with the covariates X, $S_0(t)$ is the baseline survival function, and $h_0(t)$ is the baseline hazard function.

A significant advantage of the COX PH model over other models is that to find the hazard ratio between patients, $\frac{h(t|p_A)}{h(t|p_B)}$, we can omit $h_0(t)$. This allows us to determine the risk between patients without making assumptions about $h_0(t)$.

Estimating the survival function $S(t)$ for a patient involves estimating $S_0(t) = \exp(-H_0(t))$, where $H_0(t)$ is the baseline cumulative hazard function. We can use Breslow's nonparametric estimator to estimate $H_0(t)$ and thereby $S(t)$ [19].

The COX PH model has a regularized variant that incorporates both Lasso (ℓ_1 norm regularizer) and Ridge (ℓ_2 norm regularizer) penalties, combined to form an Elastic Net, which is beneficial for feature selection. [20]

The Elastic Net formulation is given by:

$$\lambda \left[\alpha \sum_{p=1}^{P} |\beta_p| + \frac{1}{2}(1-\alpha) \sum_{p=1}^{P} \beta_p^2 \right]$$

where λ is the regularization parameter controlling the overall strength of the regularization, α gauges the balance between the ℓ_1 and ℓ_2 regularizations, P is the total number of features, and β are the coefficients obtained in the COX PH model.

4 - Materials and Methods

In this section we will go over the data we used and how we preprocessed it, as well as the creation of the PPI network that is to be integrated in the graph autoencoder. Finally, we will explain the design choices we have taken, as well as the implementation of the autoencoders and the statistical models.

4.1 Data Preparation

Data description The data used in this study are derived from the JAVELIN Renal 101 trial [21], a randomized phase 3 trial. The primary aim of this trial is to investigate and compare the efficacy between two treatment arms: an immunotherapy regimen involving avelumab and axitinib, and a tyrosine kinase inhibitor (TKI) named sunitinib. The trial included 886 eligible patients, all diagnosed with previously untreated renal cell carcinoma, over 18 years of age, and randomly assigned to the immunotherapy or sunitinib treatment arm. Of these 886 patients, 726 have transcriptomic measurements consisting of 22,955 genes, represented by positive real values.

Additionally, there are six histology measurements available for the patients. These measurements include metrics such as the percentage of cancer cells in the tumor area and the number of infected cells in the invasive margin. It is crucial to note that all these measurements are taken prior to treatment, ensuring they are not influenced by the treatment itself. Furthermore, demographic information, including the patient's age, sex, Progression-Free Survival (PFS), and censoring status, is also recorded for each patient.

Both treatment arms are similar in all respects: there are 354 patients in the avelumab+axitinib treatment arm, of which 157 (44.35%) are censored. In the sunitinib treatment arm, there are 372 patients, with 171 (45.96%) censored.

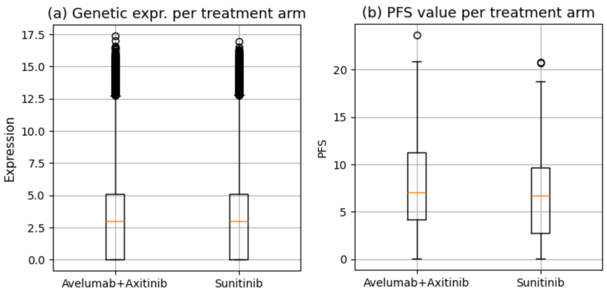

Figure 1: *(a) Genetic distribution and (b) PFS distribution among treatment arms*

Figure 1a displays the genetic data distribution for each treatment arm, showing that both distributions are very similar and should not introduce any bias in the model's predictions. Similarly, Figure 1b illustrates the PFS distribution for each treatment arm, which is also very similar, further indicating that the two groups are comparable.

Data filtering and preprocessing The JAVELIN Renal 101 trial dataset contains a total of 22,955 genes. Many of these genes are irrelevant to the development of the disease or are expressed only in a small subset of patients. This introduces noise that can complicate the model's inference based on the overall genetic expression of the patient.

To reduce this noise, several filtering steps were applied:

1. **Gene Filtering**: Genes specific to certain individuals, characterized by low expression in most patients and higher expression in a small subset, were removed. Let G be the set of genes expressed in each patient, and X_g be the expression for a gene $g \in G$. The genes removed satisfied the following condition:

   $$\forall g \in G : \left(\text{median}(X_g) < 2 \;\;\&\;\; Q3(X_g) < 4\right)$$

   This means that genes with a median expression lower than 2 and an upper quartile lower than 4 were excluded.
2. **Integration of Histology Features**: Since the histology features available were all pre-treatment, they were integrated to enhance the statistical model's inference. Out of the seven histology features, only two were measured in all patients: *he_tumor_cell_content_in_tumor_area* and *PD-L1_total_immune_cells_per_tumor_area*, measuring the percentage of cancer cells and the number of PD-L1+ cells in the tumor area, respectively. Only these two features were included to avoid further reducing the sample size.
3. **Selection of RCC-Specific Genes**: To further refine the dataset, genes specific to renal cell carcinoma (RCC) were selected using DisGeNET [22],

a platform that aggregates information on human disease-associated genes and variants. Gene-disease associations were queried for RCC (CUI: C0007134) and three related conditions: conventional (clear cell) RCC (CUI: C0279702), hereditary RCC (CUI: C2608055), and familial RCC (CUI: C2931352). This resulted in a total of 4,447 gene-disease associations (GDAs), corresponding to 2,582 unique genes, represented in *symbol* format, a type representation for genes. When intersected with the original expression data, the total number of relevant genes was reduced to 2,403.

We also applied several pre-processing steps. Regarding the features, which consist of genetic and histology variables, we normalized the genetic expressions based on the maximal global expression value. This is because transcriptomics is based on counting, ensuring that variations in gene expression levels are accounted for. For the histology variables, traditional normalization was applied.

As for the labels, since the PFS value is measured in months, we converted it to trimesters to facilitate better clinical analysis.

As a result of preprocessing, we ended up with 726 patients, 354 in the avelumab+axitinib arm, and 372 in the sunitinib arm, where each patient has 2,403 genes.

4.2 Protein to protein interaction network

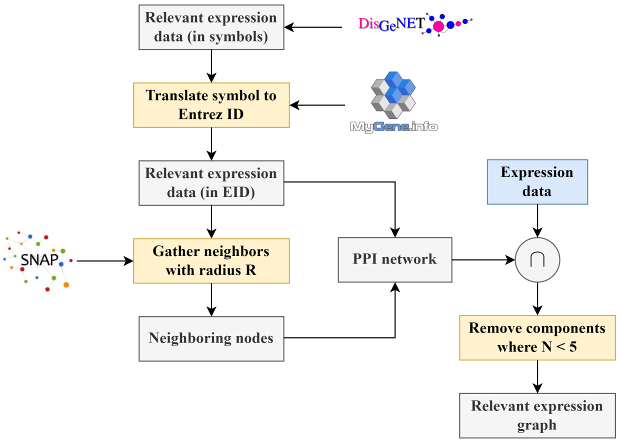
Relevant expression data (in symbols)
DisGeNET
Translate symbol to Entrez ID
MyGene.info
Relevant expression data (in EID)
Expression data
SNAP
Gather neighbors with radius R
PPI network
∩
Neighboring nodes
Remove components where N < 5
Relevant expression graph

Figure 2: *Visualization of the pipeline to create the PPI networks associated to each patient.*

Having obtained a relevant set of genes associated with the disease, we decided to build a PPI network, a graphical representation of the physical interactions between proteins within a biological system. This representation is intended to give context to the model on how information flows through the different genes in our dataset. We obtained the PPI data from PPT-Ohmnet, an interaction network from the Stanford Network Analysis Platform (SNAP) [23]. PPT-Ohmnet has a collection of physical PPI networks across different human tissues. In this network, proteins are represented as nodes, and the edges are the physical interactions between proteins.

Given the nature of RCC, we selected the sub-network that includes only kidney tissues. This subnetwork consists of 3,304 nodes and 52,126 edges. Due to the network being generic to the kidney, we still needed to perform feature selection to only obtain those genes relevant to RCC. As such, using the 2,403 genes obtained in Section 4.1, all relevant to RCC, we retrieved the related kidney-specific interactions, removing self-loops and disconnected components smaller than five nodes, as shown in Figure 2. Genes in PPT-Ohmnet are represented in *Entrez ID* format, a different form of representation than the genes queried from DisGeNET. For this reason, we translated between different gene representations using a library called mygene [24].

We retrieved 837 nodes. Given the limited number of nodes, we created a recursive neighbor lookup with a depth of 3 to add similar genes to the gene pool using breadth-first search. This resulted in a PPI network consisting of 2,870 nodes, where each node holds the expression level of a given gene.

We performed a graph analysis to ensure that no meaningful bridges existed within the graph. A bridge would indicate that the graph is comprised of two or more large components, affecting the flow of information. Additionally, we calculated the betweenness and closeness centralities of the graph, producing a mean and standard deviation of $5.90 \times 10^{-4} \pm 0.02$ and 0.37 ± 0.04, respectively. A betweenness of 1 would suggest that a path in the network has a large influence on the flow of information, and a closeness of 1 would indicate that nodes are, on average, very close to each other. These values indicate that we have an interconnected graph with multiple paths where information can flow and where all nodes are not adjacent to eachother.

With the PPI network ready, we created one for each of the 726 different patients. The networks between patients are identical except for the node attributes, which hold different expression levels for each gene, depending on the patient. In addition, the graphs are attributed with the histology features aforementioned for each patient.

4.3 Design

The models developed for this work all follow the same pipeline, seen in Figure 3. They compress the data using an autoencoder, and once the representation is learned, the latent representation is fed to the COX PH model, which computes the hazard ratios of the patients and returns the coefficients assigned to each feature. This model is combined with Breslow's estimator [19] to estimate survival functions for prediction.

The key difference between the models lies in the type of autoencoder used. Two different autoencoders were developed for this work: (i) a tabular autoencoder, which considers only tabular data where the genes are not placed within a context and can be freely combined, and (ii) a graph autoencoder, which considers the previously mentioned PPI network and reconstructs the features based on the context provided by the graph. Both autoen-

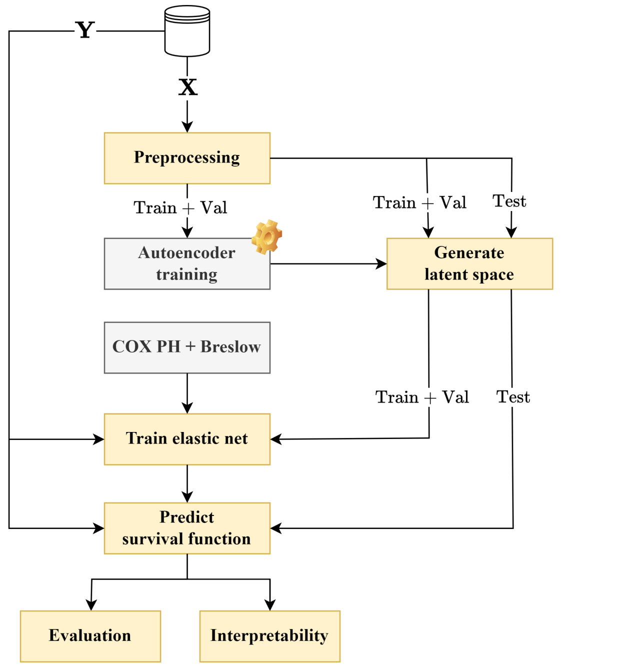

Figure 3: *Pipeline for training and evaluating the models.* **X** *represents the genetic and histology data, while* **Y** *represents the set of PFS and censoring values for the patients.*

coders compare the Mean Squared Error (MSE) between the original and reconstructed features.

We incorporated cross-validation (CV) in our model generation process. During CV, we further divided the data of the training folds into 85% for training and 15% for validation. The autoencoders were trained using the training set, and to prevent overfitting, we evaluated the models on the validation set. Finally, the remaining fold was used for testing.

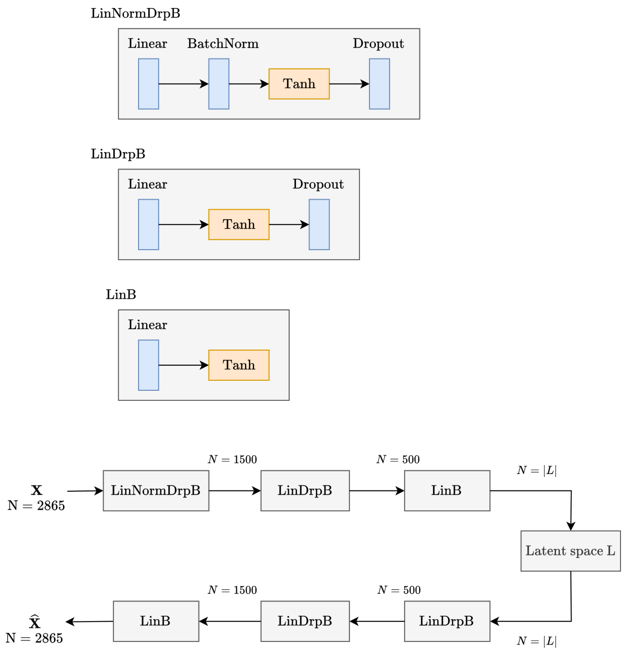

Figure 4: *Architecture of the tabular autoencoder.*

The architecture of the tabular autoencoder can be seen in Figure 4. All blocks used combine linear layers in order to reduce the dimensionality of the input, along activation functions such as tanh and sigmoid. The sigmoid activation function was employed in order to obtain a latent representation with values between 0 and 1, which could then be concatenated with the histology features, which are normalized as well.

Additionally, a BatchNorm layer [25] was added to normalize the data to a mean of 0 and unit variance. This step significantly improved the generalization process. Finally, dropout layers, where 20% of weights are set to zero, were added to prevent the model from overfitting to the training data.

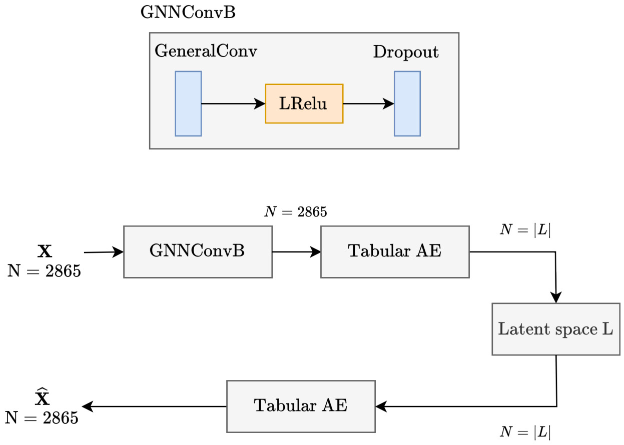

Figure 5: *Architecture of the graph autoencoder.*

The graph autoencoder is slightly more complex than the tabular one. The input graphs, obtained using the NetworkX [26] library, are processed through a GeneralConv layer consisting of several message passing layers, which reportedly yield better results than the default GCN layer [27].

Similarly to the tabular autoencoder, we add a dropout layer with a rate of 10%. Once the graphs are processed through the layer, the information flowed through the connections present in the PPI network. We then achieve a compression of the data by using the tabular autoencoder defined in Fig. 4. As mentioned in subsection 3.1, among the different reconstruction options, we chose to reconstruct the features, as we are particularly interested in obtaining a representation of the gene expression levels rather than the connections in the PPI network.

As for the stadistical model, we train the COX PH model with the latent representation of both the training and validation sets concatenated, since its implementation does not include a validation process [28]. We explained in Section 3 that the elastic net incorporated to the COX PH model uses a hyperparameter α for regularization. To find the optimal α, we performed a grid search in the set $[10^{-4}; 10^{-2}]$, with a step of 2×10^{-4}. We keep the α for which the Concordance Index IPCW is maximal with the training and validation sets combined. To improve the regression of the statistical model, we standardize the data so that it has a mean of zero and unit variance. Once the COX model is trained, we use the test set to obtain the survival functions and the area under the ROC curve, utilizing the optimal α value identified earlier.

4.4 Implementation

Both autoencoders were implemented with a batch size of 16 samples, for a total of 100 epochs, with a learning rate of 1×10^{-4}, which decays by a factor of 0.5 every 25 epochs, and latent dimensionalities of $16, 64, 128$. Regarding the penalties of the autoencoders, the denoising autoencoder applies a noise input of 0.1% to the normalized data. Adding more noise deteriorates the quality of the encoding due to the low amount of samples. The sparse autoencoder uses a regularization strength of 10% and a parameter ρ of 1×10^{-3}. To speed up the training process, GPU compatibility was added to the autoencoders.

For the COX model with elastic net, we used the implementation in the scikit-survival 0.22.2 [28] library, selecting a regularization ratio of $\alpha = 0.5$ to balance the ℓ_1 and ℓ_2 regularizations. A grid search was performed to find the best regularization term λ, within the interval $[10^{-4}; 10^{-2}]$, with a step of 2×10^{-4}. The search focused on the best concordance index IPCW, a modification of the original concordance index that provides a better estimate when censored data is present [29]. Additionally, a stratified K-Fold with $K = 7$ was used to ensure an equal number of observed samples in both the train and test sets. Once the COX model returned the coefficients associated with each covariate, Breslow's estimator was used to obtain the survival functions. The PFS prediction was obtained using the area under the survival function.

Finally, we interpreted the results obtained by the model. We evaluated the mutual information between the five latent features with the highest weights, as determined by the semi-parametric model, and the original transcriptomic data. This evaluation allowed us to identify the genes most represented within the latent representations, which were subsequently chosen by the statistical model.

5 - Experimental Results

5.1 Experiments

We studied how different autoencoder types and penalties performed in predicting PFS and risk scores of new patients. For both tabular and graph autoencoders, we evaluated the performance of the sparse, denoising, and variational penalties, as well as the combination of sparse and denoising penalties.

For each model, we obtained a series of metrics:

- **Autoencoder Reconstruction**: Measures how well the autoencoder can reconstruct the input data.
- **MSE between Actual and Predicted PFS**: Evaluates the accuracy of PFS predictions.
- **Mean of the Area Under ROC**: Indicates the overall quality of risk assignment by the model.
- **Overestimation of PFS Predictions**: The percentage of times the predicted PFS is higher than the actual PFS of the patient, where 0

The first two metrics are values in the range $[0; \infty)$, where a value of 0 represents a perfect match in either the reconstruction or the predictions. The mean of the Area Under ROC is a value in the range $[0; 1]$, with a higher value representing better risk assignment by the model. The overestimation metric indicates how often the model overestimates the patient's survival, and thus a percentage of 0% would indicate that the model never overestimates the patient's survival.

Since we are working with autoencoders, we experimented on different latent dimensionalities, specifically 16, 64, 128.

To ensure the validity of the results, we performed a 10-fold cross-validation. We also conducted an analysis of variance (ANOVA) between the results obtained by the tabular and graph autoencoders and the different autoencoder penalties used, considering a p-value under 0.05 as representing a significant difference. Additionally, we ran the analysis on the different treatment arms to determine whether the model performed better for avelumab+axitinib or sunitinib.

5.2 Results

We obtained results for different latent dimensionalities, specifically $L = \{16, 64, 128\}$, to determine how much we could compress the data without losing valuable information. Out of these three dimensionalities, we chose to elaborate on 64 latent features, not only because of the significant compression of the original data but also because the results were very similar to those with higher dimensionalities.

In Figure 6, we can see the overall results obtained for $L = 64$ over 10 folds. We gathered the best results obtained for each autoencoder type and treatment arm in Table 1.

Firstly, Figure 6(a) shows the reconstruction loss of the autoencoder relative to the original transcriptomic data. Regarding the tabular autoencoders, the reconstructions over the different folds are very similar, with the best performance by the denoising autoencoder, having a mean and standard deviation of 43.52 ± 3.51 in the sunitinib arm. Meanwhile, the variational autoencoder performed the worst, with a loss of 86.05 ± 8.00 in the avelumab+axitinib arm. The reconstruction performed by the variational autoencoder was significantly different from the autoencoder with no penalty for both the avelumab+axitinib ($p = 1.88e^{-10}$) and the sunitinib arm ($p = 1.7e^{-15}$).

The graph autoencoders did not reconstruct the data as well as their tabular counterparts, with the best reconstruction obtained by the denoising variation, achieving a loss of 50.43 ± 5.30 in the sunitinib arm, and the worst reconstruction by the variational autoencoder, with a loss of 87.97 ± 8.05 in the combination arm. As previously noted, the variational autoencoder was significantly different in both the immunotherapy ($p = 1.8e^{-7}$) and the

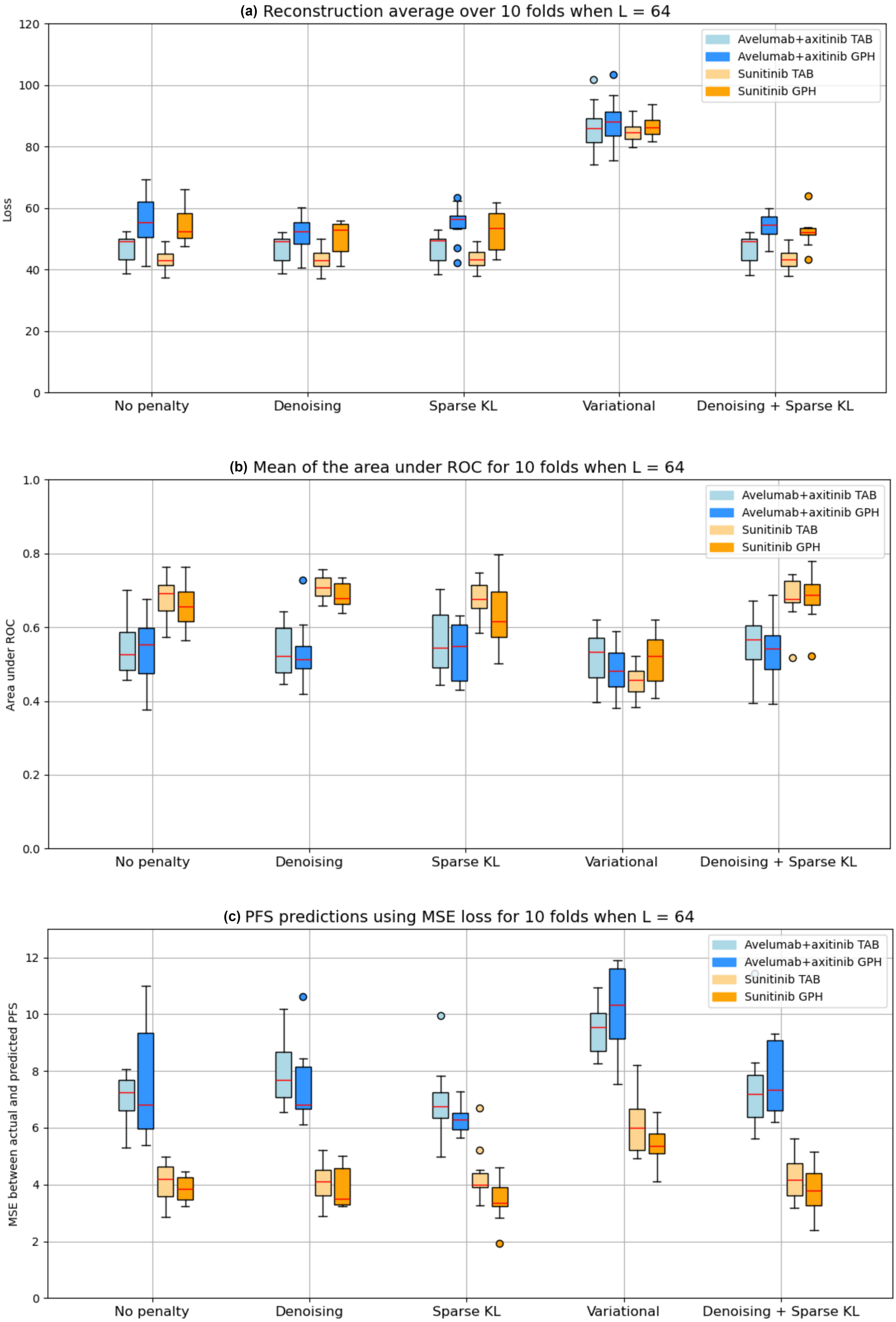

Figure 6: *Results for (a) reconstruction, (b) mean of the area under ROC, and (c) PFS prediction for each type of autoencoder over 10 folds when $L = 64$.*

Table 1: *Best results obtained for each combination arm in the reconstruction of the data, the area under curve and PFS loss. We specify the penalty they were obtained with, and underlined values represent the best results among the different models.*

Type	Arm	Reconstruction	Area under ROC	PFS
Tabular	Avelumab+axitinib	46.91 ± 4.55 Denoising	0.56 ± 0.08 Sparse	6.90 ± 1.26 Sparse
	Sunitinib	$\underline{43.52 \pm 3.51}$ Denoising	$\underline{0.708 \pm 0.03}$ Denoising	4.30 ± 0.95 Sparse
Graph	Avelumab+axitinib	51.67 ± 5.29 Denoising	0.53 ± 0.07 Sparse	6.29 ± 0.45 Sparse
	Sunitinib	50.43 ± 5.30 Denoising	0.68 ± 0.06 Denoising	$\underline{3.45 \pm 0.78}$ Sparse

TKI ($p = 8.7e^{-11}$) arms. There was a significant difference between the tabular autoencoder with no penalty and the graph autoencoder with no penalty for both the avelumab+axitinib ($p = 1.1e^{-2}$) and sunitinib ($p = 1.0e^{-4}$) treatment arms, indicating that the tabular autoencoder is indeed better at reconstructing the original data. No differences were observed between treatment arms.

Regarding Figure 6(b), we see the mean of the time-dependent area under ROC for each type of autoencoder. As defined in Section 2, this metric is an overall estimate of how well the statistical model assigns risks to each sample. A clear distinction in the tabular autoencoders is the difference in performance between treatment arms. The tabular model that obtained the best area is the denoising variant, with an area of 0.71 ± 0.03 in the sunitinib arm. Meanwhile, the variational autoencoder achieved the worst score with 0.45 ± 0.04 in the sunitinib arm. There was a significant difference when comparing the default autoencoder with the variational variant in the sunitinib arm ($p = 9.01e^{-9}$).

A similar observation can be made for the graph autoencoders. The combination of sparse and denoising obtained the best score of 0.68 ± 0.06 in the TKI, whereas the variational autoencoder performed the worst with 0.48 ± 0.07 with the immunotherapy. Similarly, the variational autoencoder was significantly different from the default one ($p = 1.6e^{-4}$). There is no significant difference between the tabular and graph autoencoders when no penalty is considered, meaning both models assign similar risks. However, there are differences between the avelumab+axitinib and sunitinib arms for all penalties except the variational one for both tabular and graph autoencoders.

Finally, Figure 6(c) shows the loss in PFS predictions among autoencoders. A similar observation to the previous figure can be made: overall, patients on sunitinib are predicted more accurately than those on avelumab+axitinib. The best variant within the tabular autoencoders is the sparse one, with a loss of 4.30 ± 0.95 in the sunitinib arm. Meanwhile, the variational autoencoder performs the worst, with a loss of 9.49 ± 0.85 in the avelumab+axitinib arm. Regarding graph autoencoders, they perform slightly better when working with the immunotherapy than tabular autoencoders do. The penalty that worked best on these autoencoders is the sparse penalty, with a loss of 3.45 ± 0.78, while the penalty that performed the worst is the variational one, with a loss of 10.16 ± 1.48. There were significant differences between the tabular and graph autoencoders in the sparse variant with the sunitinib arm ($p = 4.4e^{-2}$), indicating that the graph model performs better than the tabular one in the TKI. Furthermore, there are differences between treatment arms for all penalties in both the tabular and graph autoencoders. One notable aspect regarding the PFS prediction is that our models tend to overestimate the survival of the patients, with all autoencoders showing an overestimation of at least 70%.

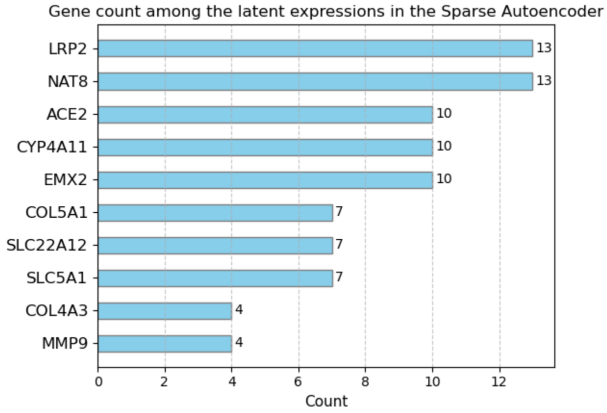

Figure 7: *Frequency of each gene appearing in the top five latent features chosen by the statistical model over 10 folds for the sparse autoencoder.*

Finally, we integrated interpretability into the models to identify which genes are most represented by the latent features. Considering the sparse autoencoder, which is among the models that obtained the best results and is known for finding meaningful representations, we obtained the five latent features with the highest coefficients assigned by the COX model. By finding the mutual information of each of these five latent features with the original data, we queried the top five genes most correlated to each representation. Figure 7 shows how many times each gene appeared in the most important latent features, with genes LRP2, NAT8, ACE2, CYP4A11, and EMX2 being the most frequent.

6 - Discussion

We can draw multiple discussion points from the previously presented results.

Autoencoder performance First and foremost, autoencoders accurately recreated the original transcriptomic data, despite the limited number of samples and the data's complexity. These models are, therefore, suitable for compressing this type of data. The tabular autoencoder generally outperformed its graph variant in reconstruction accuracy, even though both used the same hyperparameters and had very similar architectures.

However, while the tabular autoencoder can establish non-linear relationships between all genes, the graph autoencoder is constrained by the edges present in the PPI network. Additionally, the creation of the PPI network depends on the genes selected in DisGeNet and the neighbor gathering radius used to create the network. Consequently, the graph autoencoder's reconstruction performance is highly dependent on the structure of the PPI network, which can translate in a disparate reconstruction.

Regarding the penalties used, the denoising penalty provided the best reconstruction, particularly for the graph autoencoder. This aligns with the nature of the penalty, as the added noise aids in better generalization [30]. The sparse autoencoder did not reconstruct as well as the denoising penalty when working with graph data because its primary aim is to obtain meaningful representations by deactivating weights in the neural network [31]. Nonetheless, combining sparse and denoising penalties improved reconstruction quality. Unfortunately, the variational autoencoder neither reconstructed nor predicted as well as the other penalties. Variational autoencoders impose constraints on the latent space to follow a specific distribution (in our case, Gaussian), which can lead to information loss in the latent representation [32].

Statistical approach The statistical model we used consists of two parts: the COX model to find hazard ratios (and thus, risks), and Breslow's estimator, a non-parametric estimator, to find the survival functions. While the PFS predictions are accurate, especially for the graph autoencoder with sparse penalty, there is still a significant overestimation. This overestimation is due to Breslow's estimator's tendency to overestimate survival probabilities, particularly for small sample sizes [33]. Increasing the sample size or using a different estimator, could mitigate this issue. Despite this, the semi-parametric model worked well, as evidenced by the areas under ROC, although this was only the case for the sunitinib arm. The avelumab+axitinib arm's performance was close to that of a random model.

Interpretability Aside from performing predictions, we identified the genes most represented in the latent representations of the sparse autoencoder, as seen in Figure 7. Among these genes, LRP2 and EMX2 are closely related to renal cancer [34, 35].

Clinical discussion We observed a notable difference in performance between the avelumab+axitinib and sunitinib treatment arms, despite both arms having similar transcriptomic data, PFS, and censoring. One possible explanation for this phenomenon is that the immunotherapy consists of two different mechanisms (avelumab and axitinib), whereas sunitinib is not mixed with any other component. Consequently, predicting the response to one unique mechanism is more straightforward than predicting the response to a combination. An interesting future direction could be to analyze the difference in response between sunitinib and one component of the immunotherapy.

Novelties and limitations This work demonstrates the feasibility of creating meaningful representations of complex data in a low-dimensional space. Although further compression of the data is possible, doing so would likely degrade the statistical model's predictive performance.

Moreover, our work presents some limitations in its design which would prove interesting to tackle in the future.

First, gathering more data can not only improve the reconstruction capabilities of the autoencoder but also enhance the inference performed by the statistical model. Non-parametric and semi-parametric models heavily rely on the observed data, so increasing the sample size is crucial for more accurate predictions.

Moreover, the preprocessing of the data, particularly in creating the PPI graphs, can be further refined. When we selected the genes from DisGeNet to be part of the graph, the resulting network was disconnected and sparse, necessitating a neighbor search. However, some of these genes might not be highly relevant to the disease. Therefore, a more curated filtering process can be employed to ensure the inclusion of only the most meaningful genes.

Regarding the statistical model, while the COX elastic net proved to be an efficient tool, further research could explore the use of parametric models. However, this would require a thorough understanding of the distribution of transcriptomic data to select the appropriate model. Parametric models could potentially provide more precise predictions if the correct distribution assumptions are made.

Finally, for interpretability, we attempted to give meaning to the latent representations based on the genes we analyzed. Methods other than mutual information, such as alternative interpretability techniques and visualizations of genes in the latent space, could be explored to provide deeper insights into the data and the models' decision-making processes.

Overall, this work shows that autoencoders can be effectively combined with statistical models to yield meaningful results. Although models like the COX elastic net can take a long time to produce results, compressing the data beforehand can be very useful, provided the model remains interpretable.

7 - Conclusion

We have conducted extensive research on different autoencoder types and variations to analyze their efficacy in predicting the response of two different treatment arms used to combat renal cell carcinoma. Despite the complexity of the data, autoencoders have proven to be excellent tools for compressing the dimensionality of the data into meaningful features, which can then be utilized by statistical approaches.

Our analyses have shown that the denoising penalty is optimal for reconstructing data, the sparse penalty is effective for obtaining latent features, and combining these penalties can yield both characteristics. The variational penalty, while crucial in generative scenarios, assumes a distribution that does not always suit the data.

We believe this methodology can be applied in medical fields beyond renal cancer, offering a robust approach to handling high-dimensional data and extracting meaningful insights for various diseases. While not apt for clinical practices, the possibility of using this tool for screening patients who are candidates for therapy has a very high clinical impact, both for the optimization of processes and costs, as well as for patient survival.

This work can be further extended in multiple ways. First, a new cohort of patients can be obtained to confirm the results we found. Secondly, a more extensive fine-tuning, as well as model implementation such as β-VAE can be done. Finally, expanding upon our statistical approach by adding parametric estimators can help the predictions made by the model.

References

[1] Tomoyuki Makino et al. "Epidemiology and prevention of renal cell carcinoma". In: *Cancers* 14.16 (2022), p. 4059.

[2] Laura Bukavina et al. "Epidemiology of renal cell carcinoma: 2022 update". In: *European urology* 82.5 (2022), pp. 529–542.

[3] Sara Bahadoram et al. "Renal cell carcinoma: an overview of the epidemiology, diagnosis, and treatment". In: *G Ital Nefrol* 39.3 (2022), p. 1.

[4] Juliet Usher-Smith et al. "Current evidence on screening for renal cancer". In: *Nature Reviews Urology* 17.11 (2020), pp. 637–642.

[5] Frank Emmert-Streib and Matthias Dehmer. "Introduction to survival analysis in practice". In: *Machine Learning and Knowledge Extraction* 1.3 (2019), pp. 1013–1038.

[6] Jean Fan, Kamil Slowikowski, and Fan Zhang. "Single-cell transcriptomics in cancer: computational challenges and opportunities". In: *Experimental & Molecular Medicine* 52.9 (2020), pp. 1452–1465.

[7] Ping Wang, Yan Li, and Chandan K Reddy. "Machine learning for survival analysis: A survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), pp. 1–36.

[8] Ádám Nagy, Gyöngyi Munkácsy, and Balázs Győrffy. "Pan-cancer survival analysis of cancer hallmark genes". In: *Scientific reports* 11.1 (2021), p. 6047.

[9] Jacob G Ellen et al. "Autoencoder-based multimodal prediction of non-small cell lung cancer survival". In: *Scientific Reports* 13.1 (2023), p. 15761.

[10] Salah Rifai et al. "Higher order contractive auto-encoder". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*. Springer. 2011, pp. 645–660.

[11] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL: `probml.ai`.

[12] Pengzhi Li, Yan Pei, and Jianqiang Li. "A comprehensive survey on design and application of autoencoder in deep learning". In: *Applied Soft Computing* 138 (2023), p. 110176.

[13] Pascal Vincent et al. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.

[14] Andrew Ng et al. "Sparse autoencoder". In: *CS294A Lecture notes* 72.2011 (2011), pp. 1–19.

[15] Nan Jiang et al. "An empirical analysis of different sparse penalties for autoencoder in unsupervised feature learning". In: *2015 international joint conference on neural networks (IJCNN)*. IEEE. 2015, pp. 1–8.

[16] Diederik P Kingma, Max Welling, et al. "An introduction to variational autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[17] Yang Qiu et al. "Denoising sparse autoencoder-based ictal EEG classification". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.9 (2018), pp. 1717–1726.

[18] Zhenyu Hou et al. "Graphmae: Self-supervised masked graph autoencoders". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 594–604.

[19] DY Lin. "On the Breslow estimator". In: *Lifetime data analysis* 13 (2007), pp. 471–480.

[20] Vahid Ebrahimi et al. "Predictive determinants of overall survival among re-infected COVID-19 patients using the elastic-net regularized Cox proportional hazards model: a machine-learning algorithm". In: *BMC public health* 22 (2022), pp. 1–10.

[21] Robert J Motzer et al. "Avelumab plus axitinib versus sunitinib in advanced renal cell carcinoma: biomarker analysis of the phase 3 JAVELIN Renal 101 trial". In: *Nature medicine* 26.11 (2020), pp. 1733–1741.

[22] Janet Piñero et al. "DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants". In: *Nucleic acids research* (2016), gkw943.

[23] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. `http://snap.stanford.edu/data`. June 2014.

[24] Chunlei Wu, Ian MacLeod, and Andrew I Su. "BioGPS and MyGene. info: organizing online, gene-centric information". In: *Nucleic acids research* 41.D1 (2013), pp. D561–D565.

[25] Nils Bjorck et al. "Understanding batch normalization". In: *Advances in neural information processing systems* 31 (2018).

[26] Aric Hagberg, Pieter J Swart, and Daniel A Schult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

[27] Jiaxuan You, Zhitao Ying, and Jure Leskovec. "Design space for graph neural networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17009–17021.

[28] Sebastian Pölsterl. "scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn". In: *Journal of Machine Learning Research* 21.212 (2020), pp. 1–6.

[29] Thomas A Gerds et al. "Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring". In: *Statistics in medicine* 32.13 (2013), pp. 2173–2184.

[30] Lovedeep Gondara. "Medical image denoising using convolutional denoising autoencoders". In: *2016 IEEE 16th international conference on data mining workshops (ICDMW)*. IEEE. 2016, pp. 241–246.

[31] Binghao Yan and Guodong Han. "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system". In: *IEEE Access* 6 (2018), pp. 41238–41248.

[32] Shengjia Zhao, Jiaming Song, and Stefano Ermon. "Infovae: Information maximizing variational autoencoders". In: *arXiv preprint arXiv:1706.02262* (2017).

[33] Fang Xia, Jing Ning, and Xuelin Huang. "Empirical comparison of the breslow estimator and the kalbfleisch prentice estimator for survival functions". In: *Journal of biometrics & biostatistics* 9.2 (2018).

[34] Martin Q Rasmussen et al. "Epigenetic Silencing of LRP2 Is Associated with Dedifferentiation and Poor Survival in Multiple Solid Tumor Types". In: *Cancers* 15.6 (2023), p. 1830.

[35] Xinhao Niu et al. "ACE2 is a prognostic biomarker and associated with immune infiltration in kidney renal clear cell carcinoma: implication for COVID-19". In: *Journal of Oncology* 2021 (2021).