POLITECNICO DI TORINO
MASTER DEGREE COURSE IN BIOMEDICAL ENGINEERING

A SUPERVISED ADVERSARIAL AUTO-ENCODER BASED TECHNIQUE FOR DATA TRANSLATION: MRNA-MIRNA DATA TRANSLATION IN CANCER

Author:
NICOLA BARTOLINI

Supervisors:
PROF. ELISA FICARRA
PH.D. MARTA LOVINO
PH.D. GIANVITO URGESE

A thesis submitted in fulfillment of the requirements
for the degree of Master Doctor of Biomedical Engineering

in the

DAUIN
DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING

16-07-2021

"We are dwarves perched on the shoulders of giants.."

Bernard of Chartres

Abstract

Nicola BARTOLINI

A supervised Adversarial Auto-encoder based technique for data translation: mRNA-miRNA data translation in cancer

The crucial role of multi-omics analysis enables researchers to experiment with new deep learning techniques for the study of genomics and disease detection. A huge amount of several data types is available. However, due to time and cost limitations, the quantification of these data for a single sample is not always possible [1]. Therefore, researchers would benefit from the prediction of a new data type (e.g. miRNA expression) given available data (e.g. mRNA expression). This process can be considered similar to the translation between two domains. In this thesis, I focused on the translation between two different transcriptomics data: miRNA and gene expression. The proposed model was implemented on a kidney cancer dataset and extended on a lung cancer one. At first, I extracted miRNA and gene expression kidney cancer data for which the same patients were available. The model consists of two supervised adversarial autoencoders (sAAEs), one for each data type considered, i.e. miRNA and gene data. An sAAE is a deep generative model composed of 3 parts working simultaneously. It is based on an autoencoder (consisted of an encoder and a decoder) and introduces a discriminator network. The training of the model is based on two sequential phases, the reconstruction, and the regularization phases. The former exploits class labels to disentangling the original information. The latter provides a lower probability of overfitting, but also the possibility of generating new consistent samples from random variables of the same imposed distribution. Combining the encoder and the decoder of both sAAEs allows the translation between miRNA and gene expression data and vice versa. The model's architecture was defined through two different tuning types. The first one has been used to define the structural parameters of the networks, the number of layers, and the size of hidden layers, while the second one has been used to obtain the hyper-parameters that most affect the network learning (e.g., learning rates, batch size and test set size). Next, the method has been evaluated on a lung cancer dataset composed of adenocarcinoma (LUAD) and squamous cell lung (LUSC) classes. The obtained results aim to compare the original and translated data. First, principal Component Analysis (PCA) allowed a qualitative visualization of data variability between original and translated

data. Then, cluster analysis was performed to measure the distributions' differences and similarities by computing clustering performance metrics. The metrics used are the adjusted RAND index, the average silhouette value, and the Calinski-Harabasz value. These metrics report a similar behaviour in the case of the original and translated datasets. Therefore, an analysis of individual patients was performed and a scatter plot was generated for each patient (the original values are in the abscissa while the translated values are in the ordinate). I performed a differential analysis to investigate if the same genes/miRNAs that differentiate tumor subclasses in the original dataset are the same as those in the translated dataset. I generated the heatmaps representing the intensity of genes/miRNAs expression, which is differentially expressed for both the original and translated data. Thus, the biological coherence of the genes/miRNAs has been preserved. The correlation between original and translated genes/miRNAs was performed to demonstrate how well this translator is.

Acknowledgements

Here I am at the end of this university course at the Politecnico di Torino that lasted just under three years.
Three truly difficult and emotional years.
In this path, I had met fantastic people, both colleagues and professors, who have given me so much from a human viewpoint. Also, I was able to get to know other people more in-depth, which allowed me to reevaluate them.

First of all, I want to thank the most important person for me, my girlfriend Giulia. With her, I shared the crucial moments that characterized my life at Polytechnic. She was the first to celebrate with me in all moments of joy and victory and she was an incredible motivator for all those of discouragement and failure. Her presence made me grow as a man and as a professional.
You gave me so much, I love you.

I want to thank my mom and dad. They have always had a lot of faith, even when I didn't. I remember when I thought I couldn't do it, they were there, reminding me how strong I was. They always wanted the best for me and I can't thank them enough. They have always been there for me and I will always be there for them.

To my brother who gives me hope for the future, a true and kind person who puts great passion and dedication in everything he does. He reminded me how important it was to put great determination into everything a person does. He is my hero.

I want to give a very heartfelt thank you to my newly acquired family, Massimo and Angela. You took me in as a son and supported me through difficult times. I have learned a lot from you. You are two beautiful people, so different, but very similar like all the best couples. For all this, I will always be grateful, and I will never forget it.

Special thanks to my grandmother Giannina and my new acquired grandmother Ada. You made my days lighter due to your lightness and carefreeness.

I thank all the people I have met on my university journey because for better or for worse they have been a part of it.

Contents

Chapter 1

Introduction

The history of medicine has been marked by developments in diagnosis that led to changing treatments and improved outcomes for patients. Although, when it comes to cancer, the diagnosis is mainly done by looking through the microscope at the appearance of the cancer cells, often the most accurate way of making a diagnosis is by defining the source organ. However, over the last couple of decades, researchers realized that a much better way to diagnose cancer is through the molecular abnormalities that distinguish the cancer cells from the normal ones in the body and these molecular abnormalities are often changes in the genome of the cancer cells.

In parallel, the evolution of artificial intelligence algorithms has driven enormous data processing capacity, for instance, through deep neural networks. In the biomedical field, machine learning algorithms have enabled the diagnosis of diseases. The main applications of deep learning algorithms in the biomedical field are omics analysis, biomedical imaging, and biomedical signals [2].

The crucial role of multi-omics analysis enables researchers to experiment with new deep learning techniques to study genomics and disease detection. A considerable amount of several data types is available. However, due to time and cost limitations, the quantification of these data for a single sample is not always possible [1]. Therefore, researchers would benefit from predicting a new data type (e.g., miRNA expression) given available data (e.g., mRNA expression). This process can be considered similar to the translation between two domains.

This thesis focused on the translation between two different transcriptomics data: miRNA and gene expression. The proposed model was implemented on a kidney cancer dataset and extended on a lung cancer one. In the first step, I extracted miRNA and gene expression kidney cancer data for which the same patients were available. Kidney tumor patients (samples) are divided into two classes: clear cell renal cell carcinoma (KIRC) and papillary renal cell carcinoma (KIRP). After pre-processing, these data resulted in 1544 miRNAs and

19373 genes for a total of 800 common patients (512 and 288 respectively for KIRP and KIRC classes). The model consists of two supervised adversarial autoencoders (sAAEs), one for each data type considered, i.e., miRNA and gene expression.

An sAAE is a deep generative model composed of 3 parts working simultaneously. It is based on an autoencoder (consisting of an encoder and a decoder), and it introduces a discriminator network. The training of the model is based on two sequential phases, the reconstruction and the regularization phase. The former exploits class labels to disentangling the original information. The latter provides a lower probability of overfitting, but also the possibility of generating new consistent samples from random variables of the same imposed distribution.

Combining the encoder and the decoder of both sAAEs allows the translation between miRNA and gene expression data and vice versa. The model's architecture was defined through two different tuning types. The first one has been used to define the structural parameters of the networks, the number of layers, and the size of hidden layers, while the second one has been used to obtain the hyper-parameters that most affect the network learning (e.g., learning rates, batch size and test set size).
Next, the method has been evaluated on a lung cancer dataset composed of adenocarcinoma (LUAD) and squamous cell lung (LUSC) classes.

The obtained results aim to compare the original and translated data. First, Principal Component Analysis (PCA) allowed a qualitative visualization of data variability between original and translated data. Then, cluster analysis was performed to measure the distributions' differences and similarities by computing clustering performance metrics. The metrics used are the adjusted RAND index, the average silhouette value, and the Calinski-Harabasz value. These metrics report a similar behavior in the case of the original and translated datasets.

However, the values for the translated data are generally higher than the corresponding original data. Therefore, an analysis of individual patients was performed. First, a random population was examined, and both original and translated data were considered. Next, a scatter plot was generated for each patient (the original values are in the abscissa while the translated values are in the ordinate). A predicted regression line was superimposed on each plot. In detail, both translators are less reliable with translations of low expression values. Gene-to-miRNA translation improves as the expression value increases while it is less visible in miRNA-to-gene translation.
In order to evaluate the goodness of predicted data, I performed a differential analysis to investigate if the genes/miRNAs that differentiate tumor subclasses in the original dataset are the same as those in the translated dataset. In

addition, I generated the heatmaps representing the intensity of genes/miRNAs expression, which is differentially expressed for both the original and translated data. Thus, the biological coherence of the genes/miRNAs has been preserved. Indeed, most of the genes/miRNAs reported in the heatmaps are among the biological markers for these specific tumor classes. In particular, I found the gene ENTPD1 alias CD39 with a high expression is a powerful prognostic marker of clear cell RCC (KIRC) patients [3], the hsa-mir-200b, hsa-mir-424 and more markers between normal kidney and different renal cell carcinoma subtypes [4]. The model has identified hsa-miR-375, hsa-miR-205, and hsa-miR-196b, which are valuable molecular markers for the classification of non-small cell lung carcinoma (NSCLC) histologic subtypes [5]. The correlation between original and translated genes/miRNAs was performed to demonstrate how well this translator works.

Chapter 2

Biological Background

This chapter will overview the fundamental mechanisms that regulate the interaction between miRNAs and messenger RNAs.

In the first part, we will initially look at what generally happens after the generation of miRNAs and how they interact with mRNAs. Then we will focus on how these interactions affect the processes of tumorigenesis.

Then, in the second part, we will explain the process of obtaining the expression levels of both genes and miRNAs. Finally, we will say what they are and why they are crucial in the diagnostic process of some types of diseases.

2.1 Introduction of microRNA

The cells of every multicellular living organism are of many types, and every one of them has a particular purpose. However, in each of them, there is a part common to all, the nucleus. It contains the DNA that encodes all the information about the organism.
The cell can differentiate itself by using or not using this information. This information resides in the genes. Genes are nucleotide sequences contained in DNA. The information used is different for each type of cell.
It means that genes are exploited differently by each of the different cells.

2.1.1 Biosintesis of miRNA

The ability to use only the information that the specific cell needs is referred to as gene silencing. The molecules that allow these genes to be used or not are miRNAs.

miRNAs are a small (22 nts) endogenous non-coding RNAs (ncRNAs), and in the following lines, we will see how they can inhibit the action of genes. Due to their regulatory function, they are the leading players in gene silencing.

Before explaining gene silencing, I want to introduce how miRNA biogenesis occurs.

MicroRNA (miRNA) is formed within the cell by the action of RNA polymerase II. The ladder produces a miRNA as a nucleotide segment that forms a hairpin ring structure. This first stage of miRNA is called primer-miRNA. In a second step, the pri-miRNA is identified and associated by the Drosha-DGCR8 complex, which cuts off a portion. The pri-mirna thus becomes the precursor of the miRNA (pre-miRNA). The pre-miRNA is thus ready to be carried out of the nucleus and Exportin 5 is the protein assigned to this task. In the cytoplasm, pre-miRNA is released and recognized by the Dicer protein. Dicer is responsible for cutting the steam loop of the pre-miRNA. Now the pre-miRNA is composed of a double-wrapped nucleotide strand.

Subsequent interaction of AGO2 with the Dicer/pre-miRNA complex allows passage of the pre-miRNA from Dicer to AGO2.
AGO2 unwinds the pre-miRNA and releases one of the two strands. The

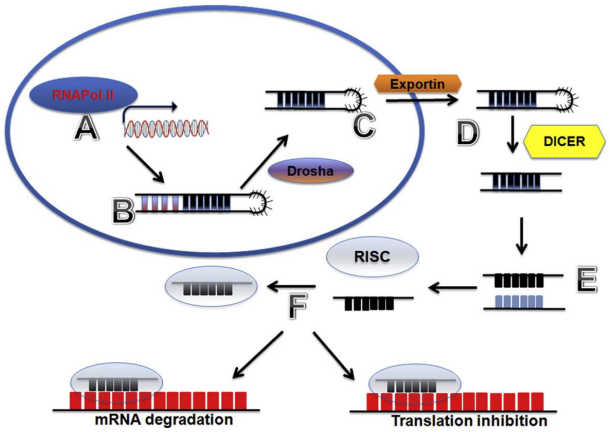


FIGURE 2.1: Biogenesis and functionality of the microRNA: the RNA polymerase II (RNA Pol II) (A) produces the primary microRNA (pri-miRNA) (B) that is sliced by Drosha-DGCR8 complex (C). In this way, the pri-miRNA becomes a small precursor hairpin microRNA (pre-miR). The Exportin5 carries out the pre-miR to the cytoplasm (D). The DICER binds the pre-miR and cuts off the hairpin steam loop. Next, the pre-miR moves from DICER to AGO2, which splits it into two single strands. One remains bound to AGO2 while the second is degraded (E). Then the RISC complex can be formed (F)

newly created complex formed by AGO2 and the miRNA strand is referred to as RISC (RNA Induced Silencing Complex). This complex allows inactivating the action of genes contained in mRNA. The binding of a miRNA to its target gene transcript contained in mRNA often relies on a 6-8 nucleotide pair region of almost perfect complementarity between the 5 end (the "seed

region") of the miRNA and its target mRNA sequence. The targeting is precise because it is determined by basic pairing between the RISC complex and the targeting mRNA. There are two ways to silence a gene: the mRNA degradation or the inhibition of the translation expression, also called translational repression (Figure 2.1. Degradation of mRNA is induced when the complementarity between the miRNA and the target mRNA sequence is high.

2.1.2 Role of miRNAs in tumor suppression and proliferation

The role of miRNAs is central in the regulation of gene expression, and this is also confirmed in cancer cells. Therefore, the action of miRNAs during the life stage of cancer cells is of great interest, especially for those miRNAs for which cancer is highly "addicted". Those miRNAs are defined "oncomiRs". In contrast, some miRNAs act as tumor suppressors. Therefore, understanding which miRNAs are oncogenic or tumor suppressors is a hot topic in research. Two lists of some of the tumor suppressors and oncomiRs validated in literature are reported in Table 2.1 and Table 2.2 respectively.

miRNA	Cancer Type	Function
miR-34b/c	Lung Cancer	A positive feedback between p53 and miR-34 mediates tumor suppression in human lung cancer
miR-126	Lung, breast and colon cancer	Plays a critical tumor-suppressor role in tumor initiation and metastasis
miR-155	Breast cancer	Downregulates RAD51 and sensitizes cancer cells to irradiation
miR-494	Lung cancer	Regulated by ERK1/2 it modulates proliferation and apoptosis response

TABLE 2.1: List of some miRNAs that are tumor suppressors validated in literature [6]

miRNA	Cancer Type	Function
miR-9	AML	Specifically overexpressed in MLL-rearranged AML and promotes leukemia progression
miR-181a/b	Liver, breast and colon cancers	Promote tumorigenesis and tumor progression
miR-21	Breast cancer	Overexpression of miR-21 contributes to proliferation and metastasis
miR-421	Gastric cancer	Marker of circulating tumor cells

TABLE 2.2: List of some oncomiRs validated in literature [6]

The following section will introduce the data types used in this thesis, what sequencing techniques are used, i.e., how they are obtained and what they represent.

2.2 Brief introduction of RNA-Seq

Gene expression and miRNA data are the product of sequencing processes. RNA-seq or RNA sequencing is a developed approach to transcriptome profiling. It is a high-throughput sequencing technique that is included among the next-generation sequencing. In addition to whole RNA (mRNA) sequencing, RNA-seq also includes RNA (mRNA) sequencing as well as small RNAs (miRNA, piRNA, siRNA) and tRNA [7]. These sequencing techniques are

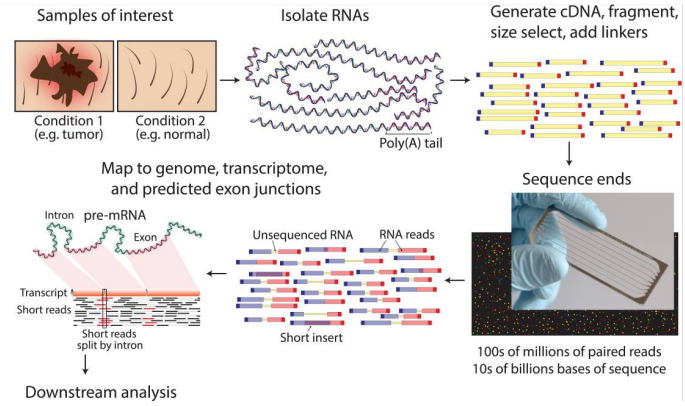


FIGURE 2.2: RNA-Seq workflow

based on complementary DNA. During the transcription step, the DNA is copied, and after removal of the "non-protein-coding" (introns), we obtain RNA, which will be composed of nucleotide bases complementary to DNA. The most popular method of RNA sequencing is done indirectly. However, direct RNA sequencing methods are growing that could lead to new gold standards [8].
After the transcription phase, there is a reverse-phase called "reverse transcription" in which RNA is converted back into DNA, in particular into complementary DNA (cDNA). In this way, it is possible to sequence RNA indirectly because the cDNAs are more stable than RNAs.
Given a sample with RNA to be sequenced: the RNA size selection is important for the types of molecules that will be sequenced. Indeed the total RNA sequencing and small non-coding RNA sequencing (microRNA-Seq) follow different pathways:

1 - the microRNA-Seq isolate the RNA through size selection. After isolation and purification, the small RNAs are converted to cDNA through reverse transcription.

2 - the mRNA-Seq convert the RNA into cDNA before the size selection. Each cDNA molecule corresponds to one RNA strand. The cDNA sequences are then randomly cut to form large fragments of equal size. In this way, the miRNAs, as the rest of small RNAs are lost [9](Figure 2.2).

In any case, the fragments or reads are aligned to the reference genome through an alignment algorithm. A gene or a miRNA expression corresponds to the number of reads that map that specific gene or miRNA.

Chapter 3

Deep Learning

In this section, the main technical instruments used for the development of the project are described. Then, those tools will be explained in order to give a clearer view of the model. The following paragraphs will introduce complex topics, starting with the general discussion of deep neural networks and ending with a discussion on generative models such as *GAN*s, Variational Auto-encoders, and adversarial Auto-encoders.

3.1 Deep Feedforward Networks

Deep neural networks are mathematical models that aim to approximate functions of interest. We can define a deep neural network with a function *f*. If we consider x the input of a neural network and f the function of the network, then *f(x)* is the output of the network. f is typically a function composed of multiple functions. Each function represents a layer of the network. A deep neural network is composed of at least 3 layer's types:

1- **Input layer**

2- **Hidden layer**

3- **Output layer**

There may be more than one hidden layer but only one input and output layer. The input and the output layers are the *visible layers*. A deep neural network composed by two hidden layers is shown in Figure 3.1 Each layer contains neurons. Each neuron is part of a layer. The neuron is the smallest part that makes up the neural network, and they are connected to the adjacent neurons of the following and the previous layer. The connection between one neuron and another is weighted: the value assumed by the next neuron will be a linear combination of the weighted values of the neurons of the previous layer to which a bias will be added. An **activation function** is applied to this linear combination, which changes according to the type of problem. For example,

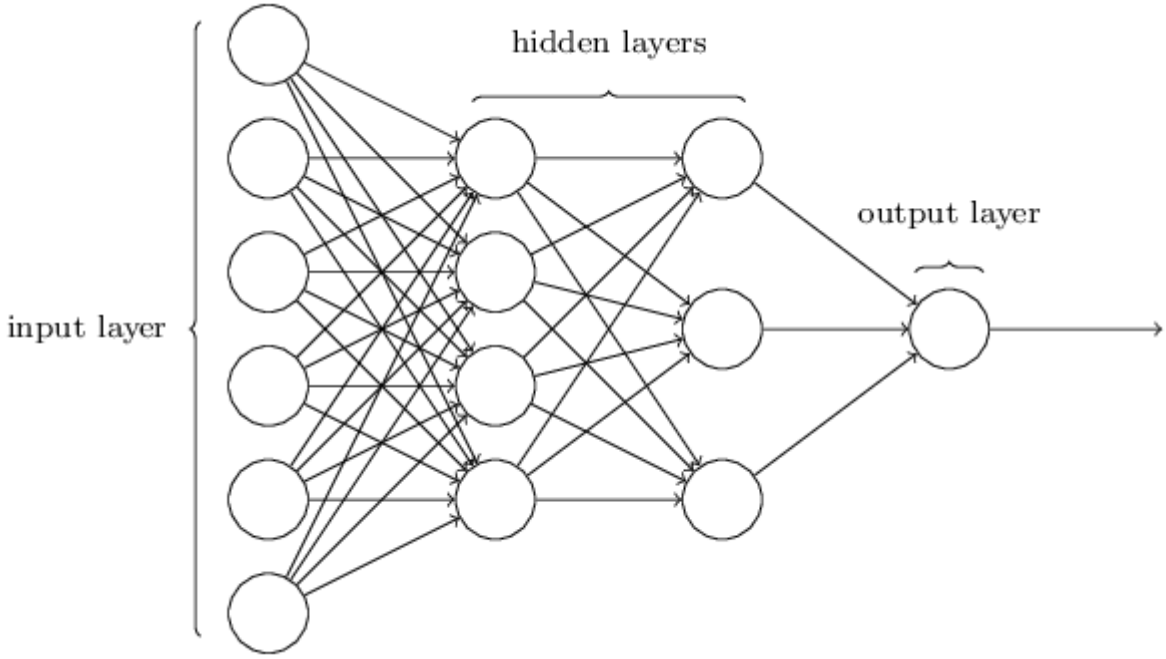


FIGURE 3.1: Neural Network with two hidden layers and single neuron output

let us consider a classification problem: given a sample input x, the classifier must understand what x is. In other words, if x has y as its class, I want the classifier to understand that x belongs to class y. If class y is a binary value, 0 or 1, then the network's output consists of a single sigmoid neuron, to which the sigmoid function is applied. A sigmoid function (Figure 3.2) is a logistic function that maps any real value on a small range, usually between 0 and 1. It can convert the output into probabilities between 0, and 1 [10]. In general, if a network consists of two layers (with input layer excluded), the network will be composed of two generic functions f_1 and f_2. If $\hat{y}$ is the predicted value of the network, the formulation will be:

$$\hat{y} = f(x) = f_2\,(f_1\,(x)) \tag{3.1}$$

The deep neural networks also called deep feed-forward networks and they are characterized by feed-forward connections between neurons: the graph formed is acyclic, i.e. the information flows along through the function f that maps the input with the output $f(x)$ [11]. The neural network architecture must be designed according to the type of application and the complexity of the problem. Defining the input and output dimensions is often straightforward. The input is defined by what data I have and the output is defined by what I want to achieve. The number of hidden layers and the number of neurons per layer are parameters that strongly affect the network's performance. Tuning

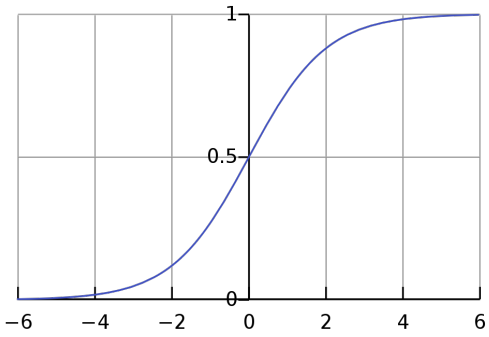


FIGURE 3.2: Sigmoid function

these parameters is necessary to ensure the best combination. It will only be necessary to utilize a **search algorithms** such as Grid Search, Random Search [12] to search for them.

One of the essential things that characterize neural networks but extendable to any machine learning problem is the training and testing phase. During the training phase, the network learns specific patterns based on data to be autonomous in the testing phase. The training phase generally consists of several iterations. We will also need to split our data into a training set and a test set. If we consider the previous classification problem, at each iteration, data from the training set is given input to the network, The learning algorithm of the neural network is the **backpropagation** algorithm. After the forward propagation through layer by layer, the network generates the output. The memory of the network consisting of the weights and biases is updated at each iteration.

3.1.1 Hyper-parameters

The performance of Neural Networks is affected by several parameters. The best way to choose those parameters is to use an optimization algorithm for optimal searching. Machine learning developers, as deep learning ones, must know which hyper-parameters affect the network the most. The parameters that are generally optimized in a deep neural network are:

- **Depth of the model**

  The depth of a deep neural network is the number of hidden layers that determine the overall length of the network chain.

- **Width of the model**

  It correspond to the dimension of each hidden layer.

- **Learning Rate**

  The learning rate is a parameter that determines how fast the learning algorithm should do its work. The optimization of the learning rate is crucial in neural network training.

- **Dropout**

  The dropout is a regularization technique used by neural networks during the training phase. The idea is that some neurons were inactivated randomly during the training phase. The parameter that the machine learning developer sets is the probability of a neuron to be zeroed. The optimization of this parameter can improve the performance of a neural network in many fields [13].

The data used to build the final model is split in two datasets:

- **Training dataset**: the part of the dataset that is used for training the network. For one (or more) deep neural networks, this coincides with training the weights and biases of the various nodes and links.
- **Test set**: is used to test the final models' performance. A test dataset is provided and consists of data that the network has never seen during the training phase. It is smaller than the training dataset.

Machine learning models can be of two categories:

- **Generative models**
- **Discriminative models**

3.2 Generative Models

This section will discuss generative models, starting from the introduction of classical Auto-encoders up to the explanation of the operation of complex generative models such as *GAN*, Variational Auto-encoders, and Adversarial Auto-encoders.

3.2.1 Auto-encoder

The Auto-encoder (*AE*) is a simple generative model that in general is formed by two deep neural networks, the encoder and the decoder, that work together. The input of this model is mapped by the encoder to a low dimensional latent space z. After that, the output of the encoder is the input of the decoder that

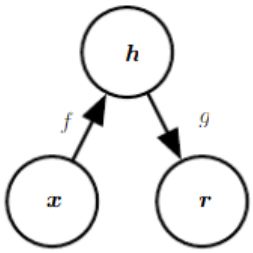


FIGURE 3.3: Summary structure of the Auto-encoder: the function f maps x to the encoded space h and the function g maps the space h to the reconstruction r

reconstruct the original data (Figure 3.3). The Auto-encoder is formally defined as the functions A: $\mathbb{R}^m \rightarrow \mathbb{R}^z$ (encoder) and B: $\mathbb{R}^z \rightarrow \mathbb{R}^m$ (decoder) where m is the input dimension and z is the latent space dimension. Input reconstruction is accomplished by minimizing a distance measure between the decoder output and the encoder input e.g. ℓ_2-norm. The ℓ_2 cost function for the reconstruction task by the Auto-encoder is

$$L(\phi, \theta) = \frac{1}{n} \sum_{i=1}^{n} [x^{(i)} - g_\phi(f_\theta(x^{(i)}))]^2 \tag{3.2}$$

where the ϕ and θ are the parameters that define the encoder and the decoder, respectively. This parameters represent the weight and the bias of the networks. n is the number of data contained in the training set and the $x^{(i)}$ correnspod to the i^{th} sample. The $g_\phi(f_\theta(x^{(i)}))$ term is the reconstructed data.

The simple Auto-encoder cannot generate new coherent samples but only generate a low-dimensional representation of the input.

The approach for the training of this network is unsupervised. Indeed, the input will be reconstructed without any label. This type of encoding is called **deterministic encoding** that allows reproducing the input as best as possible.

The Auto-encoder is widely used in the field of feature selection and anomaly detection because the latent space is a compression of the input data in which the highest level of variability of the data is concentrated. In some cases, the latent space coincides with the PCA. The Auto-encoders can be trained with all the same techniques as feed-forward networks. Indeed, it is necessary to set the same hyperparameters such as numbers of hidden layers, the hidden size, learning rate, decay of weights, momentum, and optimization strategy for backpropagation. However, the dimensionality of the latent space must also be considered. The dimensionality of the latent space consists of a parameter that affects the results of the Auto-encoder reconstruction. The latent space that the Auto-encoder would like to train is a bottleneck of the input

data. The smaller the size of the latent space, the greater the loss of information relative to the input that will occur, and this leads to an inevitable loss of quality at the output of the Auto-encoder [11] [14] [15].

3.2.2 Variational Auto-encoder

This particular type of Auto-encoder was introduced by Kingma et al. [16]. The variational Auto-encoder architecture is similar to the one of the Auto-encoder, but there are many differences in terms of purpose and operation. First of all, the variational Auto-encoder works with probability, and the encoder and the decoder become conditional probability functions.

Before dealing with the variational Auto-encoder, it is necessary to introduce some concepts of variational inference.

- **Kullback-Leibler divergence**

  The Kullback-Leibler divergence (KL) is a measure of dissimilarity of two distribution. Given two distribution $p(x)$ and $q(x)$, the KL divergence of $p(x)$ with respect $q(x)$ is

  $$KL(p(x)||q(x)) = -\sum_x p(x) log \frac{q(x)}{p(x)}. \tag{3.3}$$

  Two important properties of KL divergence are:

  - $KL \geq 0$ (always positive)
  - $KL(p(x)||q(x)) \neq KL(q(x)||p(x))$ (asymmetric function)

- **Bayes's theorem**

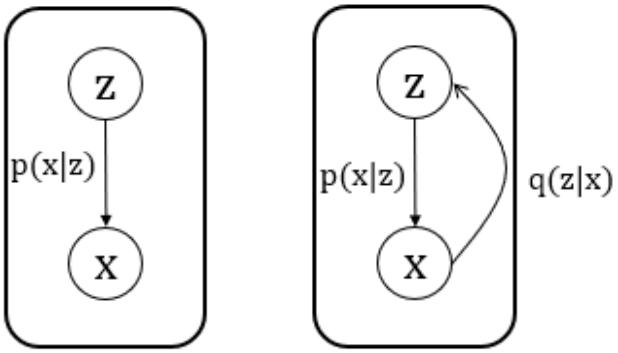
  The Bayes's theorem describes the condition probability of an event given a prior knowledge. Given a probability distribution p, the conditional probability of an event z given x is equal to

  $$p(z \mid x) = \frac{p(x \mid z)\, p(z)}{p(x)} \tag{3.4}$$

Cost function

The following part aims the obtaining of the cost function of the variational Auto-encoder Considering the graphical model reported in Figure 3.4a.

The x corresponds to the observation and the z is the latent variable. The objective of this inference problem is to compute the conditional probability of



(A) (B)

FIGURE 3.4: The graphical models that represent variational technique used for inference

z given x $p(z \mid x)$. If we consider the Eq.3.4, the conditional probability $p(z \mid x)$ depend on the marginal probability $p(x)$. As a marginal distribution, $p(x)$ is equal to

$$p(x) = \int_z p(x \mid z)\, p(z)\, dz \tag{3.5}$$

The Eq.3.5 is an intractable integral. The variational inference tries to solve this problem by avoiding it. Indeed, it tries to approximate $p(x \mid z)$ with another distribution $q(z)$

The minimization problem of the KL between this two distribution occurs

$$argmin\{KL(q(z)||p(x|z))\} \tag{3.6}$$

After several calculations reported [16], the problem is solved by maximization of the variational lower bound $\mathcal{L}$ that is composed by two terms:

$$\mathcal{L} = \mathcal{E}_{q(z \mid x)}[logp(x \mid z)] - KL(q(z) \mid \mid p(z)) \tag{3.7}$$

$$\mathcal{L} = reconstruction\ loss - regularization\ loss \tag{3.8}$$

Now, we can consider an Auto-encoder as in Figure 3.5. Let assume that exists another distribution $q(z \mid x)$ generated by the encoder which maps x into z. The Eq. 3.7 is the cost function of variational Auto-encoder. Indeed, the variational Auto-encoder cost function aims to make the distribution of the latent space $q(z \mid x)$ similar to a multivariate Gaussian distribution $\mathcal{N}$(z;0,I) with a mean equal to 0 and covariance equal to I matrix (in case of real-valued data). The maximization of the expectation term (the first term) is a problem of distance error minimization between the input and the input reconstructed by the

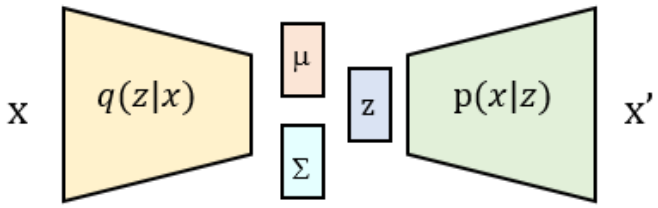


FIGURE 3.5: The general scheme of a variational Auto-encoder

decoder $p(x \mid z)$:

$$\begin{aligned} argmax\{log(p(x|\hat{x}))\} = \\ argmax\{log(e^{-|x-\hat{x}|^2})\} = \end{aligned} \tag{3.9}$$

$$argmax\{-|x-\hat{x}|^2\} \tag{3.10}$$

The maximization of the Eq. 3.10 is equal to the *minimization of the mean squared error* when the prior distribution is Gaussian distribution.

The *KL* term (the second term) means that *the distribution $q(z \mid x)$ should match another prior distribution $p(z)$* (the multivariate Gaussian distribution). This term is the "regularization term" of the total cost function.

Reparametrization trick

The backpropagation can fix the weight of the encoder and the decoder because it was introduced the *reparametrization trick* concept. The problem is that the vector z (see the Figure 3.5) is obtained by a stochastic sampling operation and the neural network backpropagates gradients through only deterministic nodes.

The reparametrization trick reparametries the sampling layer z. Indeed, the vector z is considered as the sum of a fixed μ means vector and a fixed $\sum$ covariance vector that is scaled by normal constants ϵ $\mathcal{N}(0, 1)$

$$z = \mu + \Sigma \odot \epsilon \tag{3.11}$$

In this way, the stochastic node is represented by ϵ, and the latent space z is a deterministic node concerning ϵ.

I decided to introduce this part to understand better how the cost function in Eq. 3.7 is obtained.

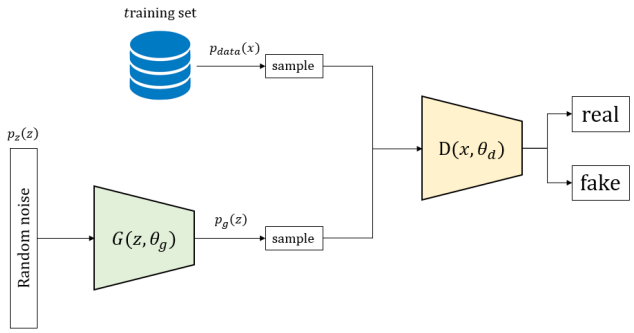


FIGURE 3.6: The structure of the Generative Adversarial Network

3.2.3 Generative Adversarial Network

The Generative Adversarial Network (*GAN*) is a particular generative model that exploits the competition between two different models. It is formed by two deep neural networks of multiple types depending on what the network needs to generate. For instance, convolutional neural networks can be used for generating images. The two adversarial models are the **discriminator** (D) and the **generator** (G). This particular model was designed for the first time by [17].

Adversarial Training

In order to generate new samples, the generator G aims to learn the data distribution of the training samples, and the discriminator D aims to estimate the probability that the input sample came from the training set rather than G. The reference paper explains *GAN* as a money counterfeiter trying to confuse the police. Each time the police realize the money is fake. The counterfeiter gets better at creating money that looks more like the real one. Ideally, after n attempts, the counterfeit money is indistinguishable from the real money for the police. In terms of probability, after n iterations, the discriminator states that the probability that a sample is from the training set or generated by G is 0.5

The Figure 3.6 shows the main structure of a *GAN*.

- $p_z(z)$ is the prior noise distribution (Gaussian, Poisson, Bernoulli distributions and other ones).
- $G(z, \theta_g)$ is the function that represents the deep neural network of the generator parametries with θ_g.
- $p_g(z)$ is the distribution of $G(z)$
- $p_{data}(x)$ is the distribution of the training data

- $D(x,\theta_d)$ is the function that represents the deep neural network of the discriminator parametries with θ_{d}.
- $D(x)$ represents the probability that x is a training sample or a generated sample (real or fake).

Adversarial training

Adversarial training consists of two main parts: the training of the discriminator and the training of the generator.

1- Discriminative Loss : the discriminator is trained to distinguish both the real distribution (the prior distribution $p_{data}(z)$) and the aggregated posterior distribution created by the generator ($p_g(z)$). In other words, the objective of the discriminator is to *maximize the probability of correctly distinguishing the input*.

2- Generative Loss : on the other hand, the generator aims at tricking the discriminator through its distributions *minimizing the probability for the discriminator to understand if the generated distributions are real or fake*.

The previous definitions can be formalized by the following cost function:

$$\min_G \max_D \mathcal{V}(G,D) = \mathbb{E}_{x\sim p_{\mathrm{data}}}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1 - D(G(z)))]. \quad (3.12)$$

in which the $D(G(z))$ is the estimate of the discriminator of the probability the $G(z)$ is real. [17] themselves discussed the criticality of a cost function. The discriminator has a much easier task than the generator: the first must distinguish the two distributions while the second must model its distribution in a way that the first does not understand. For this reason, another generator loss function is introduced to avoid the saturation of the *GAN* and its stuck in the early stages. The cost function of the generator is shown below:

$$log\ D(G(x)). \quad (3.13)$$

3.2.4 Adversarial Auto-encoder

The Adversarial Auto-encoder (*AAE*) is a generative model that combines the need of performing variational inference for the latent space of an Auto-encoder as *VAE* and the generative adversarial networks, or *GAN*.

As I already explained in the *VAE* section, one of the biggest problems of generative models is to obtain the marginal distribution in Eq. 3.5 because it is an intractable integral. Therefore, the matching between the aggregated posterior distribution of the latent space with another arbitrary prior distribution is the much more accessible and computationally low cost than the Monte Carlo

EM solution [16]. The main concept behind the *AAE* is that use the *GAN* for this matching. The structure of the *AAE* consists of an AE and a *GAN* (Figure

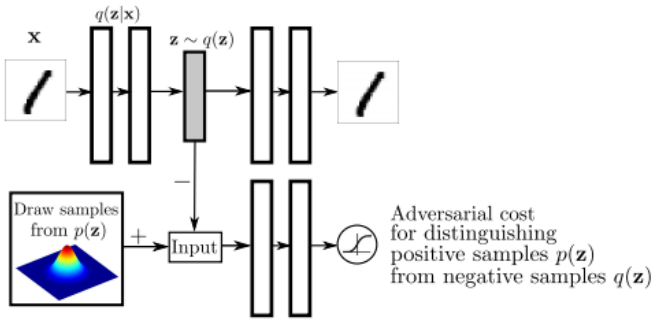


FIGURE 3.7: Original structure of Adversarial Auto-encoder

3.7). It is possible to review the structure of an Auto-encoder (explained here), with the encoder and the decoder. The neural network that maps the input x into z is the encoder and the decoder that reconstructs the input at the output. The *GAN* is composed of the discriminator, which takes as input the prior distribution *p(z)* and the aggregated posterior distribution *q(z)* that is generated by the encoder. Indeed, in this particular neural network, the encoder of the Auto-encoder becomes the generator of the *GAN*. In this way, the generator learns to fit the prior distribution.

Training phase

The training phase occurs sequentially. The training phase of the *AAE* is composed of two different steps: the training of the Auto-encoder and the training of the *GAN*. At each training epoch, the Auto-encoder is first trained, and then the discriminator and generator of *GAN* are trained. The cost function of the *AAE* is similar to the variational lower bound presented in the *VAE* training, and I want to maximize it. In particular, it is composed by two terms: *the reconstruction term* and the *regularization term*. The first term remains the minimization of a distance metric between the input and the output (the reconstructed term), but in contrast, the regularization term is acted by the *GAN*. Then, the regularization phase of the *AAE* is reported in the **Adversarial training** section of the *GAN*.

Just to summarize, the regularization term can be used to:

- generate new samples
- prevent overfitting

3.2.5 Supervised Adversarial Auto-encoders

Supervised Adversarial Auto-encoders (*sAAE*s) are a type of *AAE* that incorporates the label information of the data used for training the network. This incorporation is at the latent space level. In this way, the decoder uses both the hidden vector z and the label information to reconstruct the input data. As a result, the network learns information independently for the label, and it can be focused on the information style of the data [18].

Chapter 4

Method

In this chapter, I explain the **Method** used for developing the project. The following information is reported in order to understand what I implemented. The chapter is divided into two main parts:

- The **Data** part, obtaining the data and how it was pre-processed to be used in the method.
- The **Model** part, how the proposed neural network model is structured and how it was trained.

4.1 Introduction of the Cancer Genome Atlas

The Cancer Genome Atlas (TCGA) is an association between the National Cancer Institute (NCI), the federal government's principal agency for cancer research, and the National Human Genome Research Institute (NHGRI) that is the driving force for advancing genomics research at the National Institutes of Health (NIH). This collaboration began in 2006, and the initial focus was only on three cancer types: lung, ovarian, and brain (glioblastoma) [19]. In 2021 the TCGA has generated over two petabytes of data such as genomic, transcriptomic, and proteomic data by collecting over 11 thousand cases of 33 different types of tumor.

Their ultimate goal is to find better ways to prevent cancer and achieve better cancer patients outcomes. First of all, the molecular changes that happen in cancer cells are crucial in all phases of the clinical trials, and the characterization of those changes improves the ability to diagnose, treat and prevent cancer.

The NCI creates a Next Generation Cancer Knowledge Network called Genomic Data Commons (GDC) to put researchers in close communication and create a community where everyone has free access to qualitative data stored on the dedicated user-friendly portal. Thanks to the analysis tool DAVE they

will also have the possibility to analyze, visualize, and explore them and their applications that allow the pattern recognition to mine the huge amounts of data in the GDC portal. (https://gdc.cancer.gov/)

4.2 Retrieving the data

In the GDC Data Portal, there are several navigation options, and the Repository link directs users to the Repository Page where the data files are available for download (https://portal.gdc.cancer.gov).

On the left part of the page where we were directed, there are the data filtering columns. Each property of data can be used as a filter. The filtered data used for our model training and validation were extracted as follows:

1. **miRNA Expression Quantification:** The following filters get the kidney cancer miRNA expression data.

   Filter "Files":

   - ▷ *Data Category*: "Transcriptome Profiling"
   - ▷ *Data Type*: "miRNA Expression Quantification"
   - ▷ *Experimental Strategy*: "miRNA-Seq"
   - ▷ *Workflow Type*: "BCGSC miRNA Profiling"

   Filter "Case" (or "Biospecimen"):

   - ▷ *Samples Sample Type*: "primary tumor"
   - ▷ *Program*: "TCGA"

2. **Gene Expression Quantification:** The following filters get the kidney cancer miRNA expression data.

   Filter "Files":

   - ▷ *Data Category*: "Transcriptome Profiling"
   - ▷ *Data Type*: "Gene Expression Quantification"
   - ▷ *Experimental Strategy*: "RNA-Seq"
   - ▷ *Workflow Type*: "HTSeq - FPKM-UQ"

   Filter "Case" (or "Biospecimen"):

   - ▷ *Samples Sample Type*: "primary tumor"

▷ *Program*: "TCGA"

The reference projects for the development of the thesis are the two major histologic subtypes of renal cell carcinoma (RCC): clear cell RCC (TCGA-KIRC) and the papillary RCC (TCGA-KIRP). In addition, results from the analysis and processing of the same method of lung cancer data, in particular, Lung Adenocarcinoma (TCGA-LUAD) and Lung Squamous Cell Carcinoma (TCGA-LUSC), were added to validate the work.
The protocol used to download the TCGA data is described in the following steps:

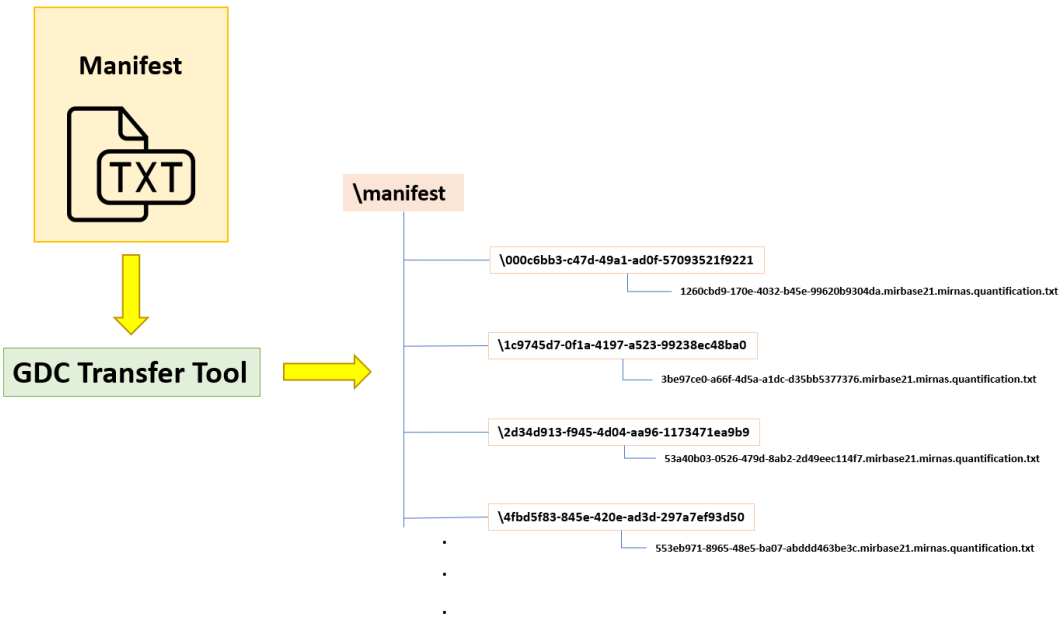


FIGURE 4.1: The workflow of a manifest file processing through the GDC Transfer Tool; on the right the directory of the miRNAs and isoform quantification files stored in a specific folder called "manifest"

Step 1 ▷ The manifest files and the information about the files are obtained using the filters previously mentioned where you need to specify the type of project (e.g., for RCC data, the projects are *TCGA-KIRC* and *TCGA-KIRP*).
The manifest files are text files that contain a list of files to be downloaded using the GDC Data Transfer Tool, recommended for transferring large volumes of data.

The information about the files are available in form of .json or .csv extension files and they contain the unique link between the file name and the

sample identifier or cases unique identifier, the *cases ID*. (Figure 4.1). In this thesis, the data extracted are of two different types: **miRNA Expression** Quantification data and **Gene Expression** Quatification data. It is possible to obtain the miRNA and the isoforms quantification files from the first type, but only the miRNAs were saved for later analysis. From the second type, only gene expression quantification files are achieved.

Step 2 ▷ An automatic algorithm has allowed the extraction of the read counts for each sample to create the expression matrices both for the miRNA and gene expression data. (Figure 4.2).

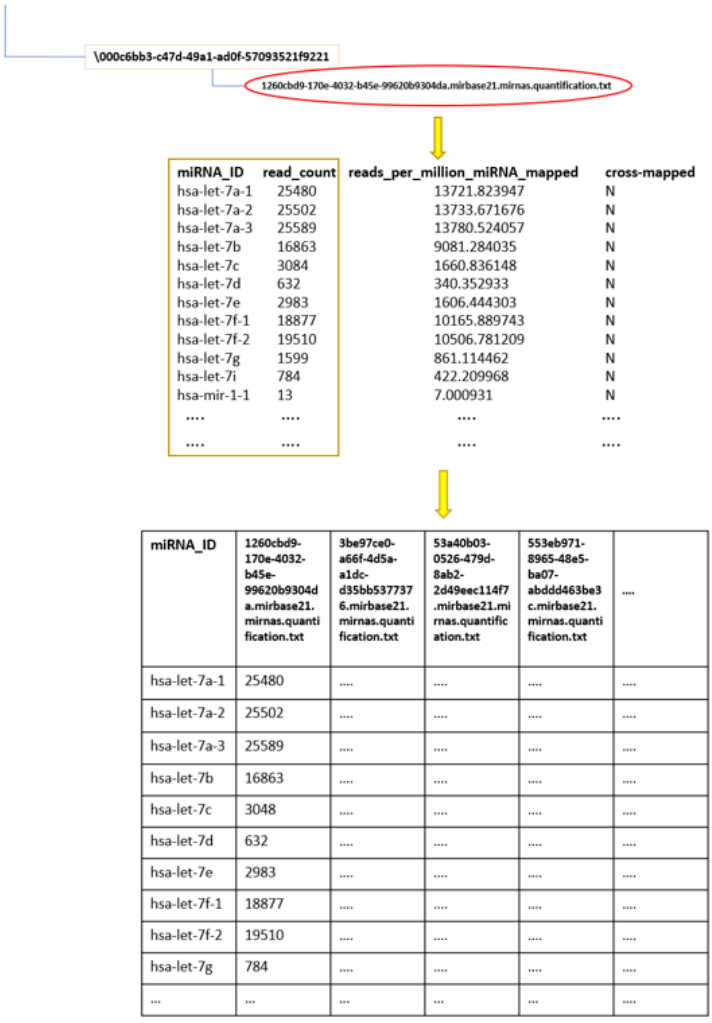
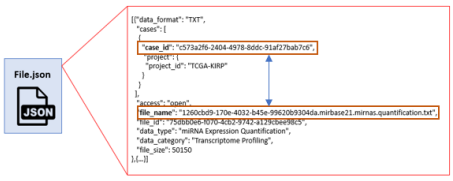


FIGURE 4.2: Creation of the raw expression matrix: the process for obtaining the matrix from top to bottom.



miRNA_ID	1260cbd9-170e-4032-b45e-99620b9304da.mirbase21.mirnas.quantification.txt	3be97ce0-a66f-4d5a-a1dc-d35bb5377376.mirbase21.mirnas.quantification.txt	53a40b03-0526-479d-8ab2-2d49eec114f7.mirbase21.mirnas.quantification.txt	553eb971-8965-48e5-ba07-abddd463be3c.mirbase21.mirnas.quantification.txt	....
hsa-let-7a-1	25480	....	....	....	....
hsa-let-7a-2	25502	....	....	....	....
hsa-let-7a-3	25589	....	....	....	....
hsa-let-7b	16863	....	....	....	....
hsa-let-7c	3048	....	....	....	....
hsa-let-7d	632	....	....	....	....
hsa-let-7e	2983	....	....	....	....
hsa-let-7f-1	18877	....	....	....	....
hsa-let-7f-2	19510	....	....	....	....
hsa-let-7g	784	....	....	....	....
...	...	...	...	...	....

FIGURE 4.3: Creation of the complete expression matrix: on the right, there is the *.json* file for changing the name of the samples with the case identifiers (case IDs)

Step 3 ▷ Furthermore, the names of the samples were changed to the corresponding identifiers (case ID), and the miRNA and gene matrixes were intersected to end up with the cases ID.

Step 4 ▷ Annotation files were created which contain the links between the cases ID and the file names both of the miRNAs and the genes for each type of project.

Step 5 ▷ The miRNA data have been normalized using DeSeq [20]. In this way, the read counts were normalized as $log_2(count+1)$.

The dimensions of the data obtained from the procedure just explained are described as follows:

Kidney Cancer Classes

▷ TCGA-KIRP: 512 cases

▷ TCGA-KIRC: 288 cases

Total miRNAs: 1544

Total genes: 19373

Lung Cancer Classes

- ▷ TCGA-LUAD: 475 cases
- ▷ TCGA-LUSC: 471 cases

Total miRNAs: 1631

Total genes: 19373

4.3 Model Architecture

The adopted approach for the implementation of the translation method was research-based on cross-function generative models. This exploratory analysis led to the development of neural network models that leverage together with the potential of adversarial training, first introduced by Goodfellow et al. [17]. This powerful neural networks system is composed of two different **supervised Adversarial Autoencoders** (*sAAE*s) for both the two data translation types. The input and output of these *sAAE*s vary depending on the size of the data, e.g. in the translation from genes to miRNAs, the input's width will be equal to the number of genes while the output's one will be equal to the number of miRNAs.

The proposed method takes inspiration from [21] because it is based on the use of Adversarial Autoencoders (*AAE*) that serves for the domain translation between multiple data types. However, I disregard the generative model part, and I assume that the encoders of the *sAAE*s used are **deterministic functions**. In this way, the Auto-encoder is not stochastic, but deterministic. In this case, the regularization term is used as a parameter that allows more stability during training and prevents problems related to overfitting. The architecture of the two *sAAE*s is different because the data have significantly different dimensions. The following sections will discuss the structure of the models used, how they were trained, and how they were combined.

The Figure 4.4 shows the final *sAAE* that can translate the two domains. The yellow part refers to the encoder that takes gene expression data as input, while the green part refers to the decoder that processes translated miRNA expression data as output. The blue vector represents the low-dimensional space where the encoder maps the meaningful information about the data. In addition, the purple vector is the label vector. In this way, I add important information to the translation that will be refined.

4.4 Training Phase

As I explained in the training phase of the **Adversarial Autoencoder** part, the original variational lower bound that I want to maximize for the training of

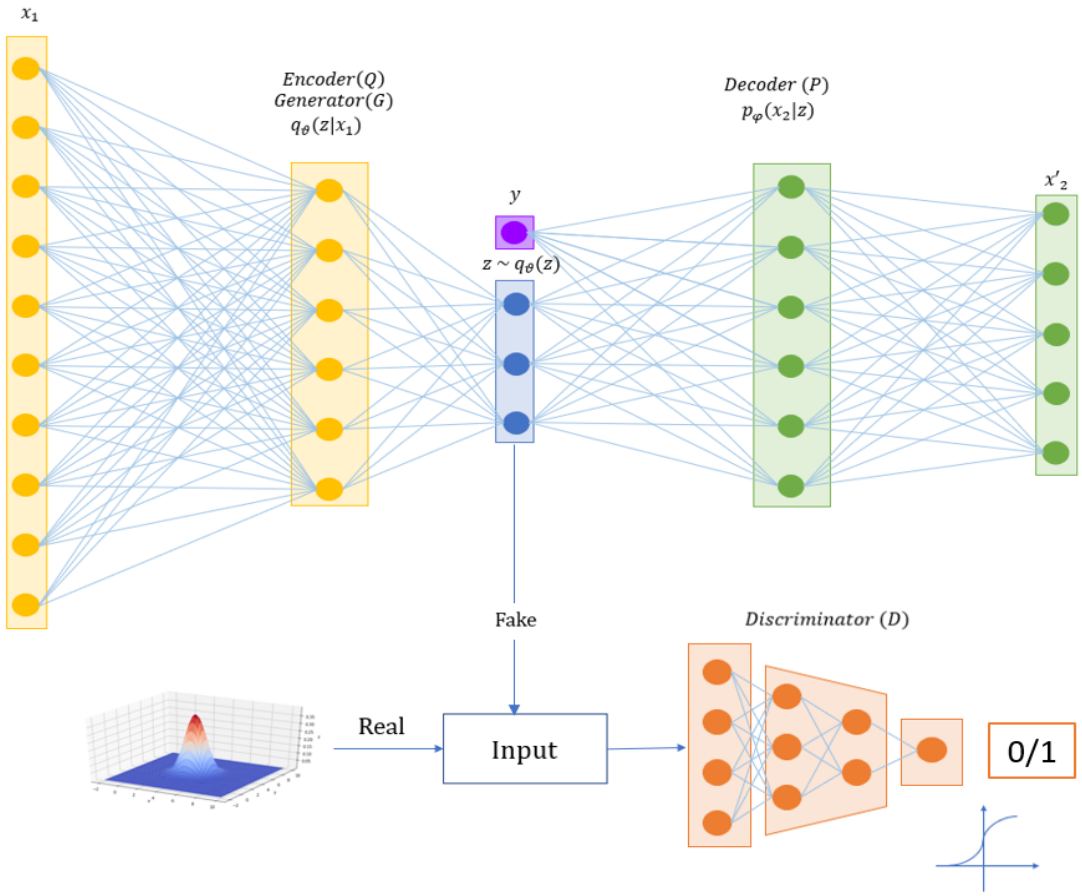


FIGURE 4.4: Scheme of the supervised Adversarial Autoencoder architecture for the gene to miRNA translation (x_1>x'_2)

an *sAAE* is composed by a reconstruction term and a regularization term. The reconstruction term becomes a translation problem if we know the ground truth. Then, the following sections are related to the two phases of the new *sAAE*.

- Translation Phase

- Regularization Phase

4.4.1 Translation Phase

Let's consider $\boldsymbol{x_i}$ the input of the encoder with function $q(z \mid x_i)$, $\boldsymbol{x'_j}$ the output of the decoder with function $p(x_j \mid z)$, and let x_j be the corresponding of the input in the other domain i.e. the ground truth (e.g. $\boldsymbol{x_1}$ is a random mini-batch of gene expression data with specific case IDs and $\boldsymbol{x_2}$ is the miRNA expression data of the same cases). During the translation phase, the data flows along the autoencoder (Figure 4.5), then the input x_1 is mapped in z by the encoder. The

latent code vector z (blue part) is the result of the encoding of the input, and it represents the aggregated posterior distribution generated by the encoder. At this point, the latent space z along with the label information y (violet part) of the encoded samples are decoded, building the translation of the input, i.e., x'_1. The training of the decoder is performed trying to *minimize the mean squared error* between the decoder output x'_1 and the ground truth x^*_1 (orange part) with the cost function:

$$\frac{1}{N}\sum_{i=1}^{N}[x_i - x'_i]^2 = \frac{1}{N}\sum_{i=1}^{N}[x_i - P(Q(x_i))]^2 \tag{4.1}$$

4.4.2 Regularization Phase

After the translation phase, the regularization phase is managed by adversarial networks. This training phase exploits the *adversarial costs* as I explained in the **Generative Adversarial Network** section. In this phase, as in [18], the encoder $q(z \mid x_i)$ of the autoencoder becomes the generator of the aggregated posterior distribution and is involved the discriminator is a neural network with single output. Also let's consider $p_Z(z)$ a prior distribution (Figure 4.6). The adversarial game just explained has been defined by [17] as the minimax

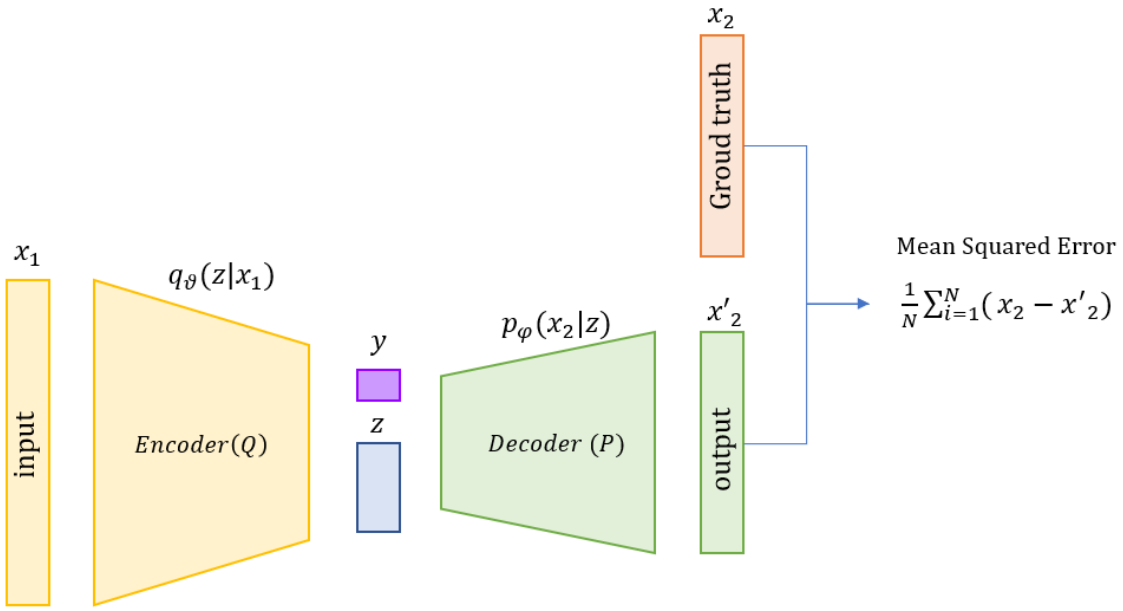


FIGURE 4.5: Translation phase data flow

game with the value function $\mathcal{V}$ in Eq.3.12

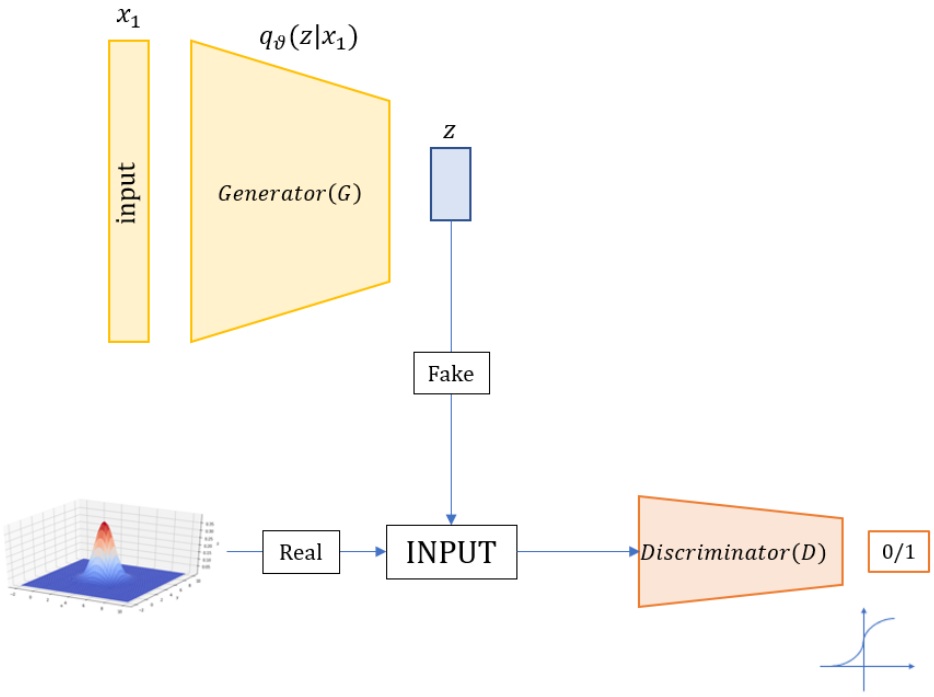


FIGURE 4.6: The work of the GAN during the regularization phase

4.4.3 Model architecture design and hyper-parameters tuning

The definition of the architecture of the networks that make up the model is presented. The first phase for constructing the models is based on hypotheses on the possible ranges of the values of interest. The parameters that characterized the architecture of the models are shown in Table 4.1 and in Table 4.2. Once these ranges were defined, a search for values was carried out by a **Random Search** algorithm. This type of searching algorithm can find models in a much more efficient way than grid search, especially in neural networks applications [12]. *Optuna* optimization framework is used for the implementation of the optimization strategy since it allows the use of a wide range of optimization algorithms as the Tree-structured Parzer models [22][23], the random search as and the grid search in an easy-to-use development environment [24]. In addition, optuna allows the implementation of pruner models such as the Asynchronous Successive Halving Algorithm (ASHA) and the Hyperband algorithm. The pruning algorithm allows a given trial to be stopped through intermediate analysis of the objective function.

In this case, **Hyperband** algorithm for the pruning tasks is used. This particular algorithm extends the idea of the Successive Halving algorithm of allocating a budget to the set of configured hyper-parameters [25]. After 50 trials, the results of this tuning is reported in Table 4.5, Table 4.3, Table 4.4. The splitting of datasets was set to 20% for gene expression data and 15% for miRNA expression data. The dimensionality of the training sets and test sets for each tumor tissue is shown in Table 4.6 and in Table 4.7

Hyper-Parameters	**Ranges**
Batch Size	[5 , 25]
Number of Hidden Layers	[1 , 2]
Learning Rate (AE)	[1E-06 , 1E-04]*
Learning Rate (D)	[1E-06 , 1E-04]*
Learning Rate (G)	[1E-06 , 1E-04]*
Hidden Size (AE)	[1000 , 3000]
Dropout (p)	[0.3 , 0.7]
Hidden Size 1 (D)	[100 , 200]
Hidden Size 2 (D)	[50 , 99]

TABLE 4.1: Parameter ranges for the model architecture design

Hyper-Parameters	**Ranges**
Batch Size	[5 , 25]
Learning Rate (AE)	[1E-06 , 1E-04]*
Learning Rate (D)	[1E-06 , 1E-04]*
Learning Rate (G)	[1E-06 , 1E-04]*
Dropout (p)	[0.3 , 0.7]

TABLE 4.2: Parameter ranges for the hyper-parameters tuning

*The data were sampled in a *log* domain between the ranges reported

The two different types of hyper-parameters tuning are performed for each translation task, i.e., considering the projects (kidney cancer data and lung cancer data) and the type of data (miRNA expression data and gene expression data). The searching of model parameters reveals that in the various types of translation, the optimal structure of the model varies enormously in function to the type of data.

The following tables show the architecture of different *sAAE*s designed by the hyper-parameters tuning:

1 - The Table 4.3 and the Table 4.4 show the dimensionality of both the Auto-Encoders and the Discriminators networks of the *sAAE*s. Each column corresponds to a layer. The value assumed by each layer is the number of neurons that make it up.
The Table 4.3 shows that the Auto-Encoders used for the translations consist of an encoder and a decoder, both composed of a hidden layer and also by a low dimensional latent space (hidden layer).

The Table 4.4 shows that the Discriminators used to regularize both types of translations are two deep neural networks composed of two hidden layers. The input layer is as wide as the low-dimensional latent space, while the output layer is a single neuron with a sigmoid activation function.

2 - The Table 4.5 shows the result of the second tuning of hyper-parameters. The first column refers to the type of translation. At each translation type, the search algorithm selected a batch size, a learning rate related to the AE, one related to the discriminator, one related to the generator, and the dimension of the test set.

NEURAL NETWORK	**Hidden1**	**Latent Space**	**Hidden1**
Autoencoder (miRNA→ *gene*)	1646	278	1646
Autoencoder (gene→ *miRNA*)	2600	273	2600

TABLE 4.3: Autoencoders' Architectures

NEURAL NETWORK	**Input**	**Hidden1**	**Hidden2**	**Output**
Discriminator (miRNA→ *gene*)	278	148	98	1
Discriminator (gene→ *miRNA*)	273	128	32	1

TABLE 4.4: Discriminators' Architectures

TRANSLATION	**Batch Size**	**lr(AE)**	**lr(D)**	**lr(G)**	**Dropout (p)**	**Test Size**
Translation miRNA→ *gene*	40	7.94e-05	8.92e-05	5.26e-06	0.57	15%
Translation gene→ *miRNA*	19	9.77e-05	1.04e-05	8.74e-06	0.61	20%

TABLE 4.5: Hyper-parameters Setting

TUMOR	TOTAL (cases)	GENOME TYPE	PROJECT	TRAINING SET	TEST SET
Kidney	800	miRNA	KIRC	410	102
			KIRP	230	58
		Gene	KIRC	435	77
			KIRP	245	43

TABLE 4.6: Dimensionality of training and test sets for each kidney cancer project and genome type

TUMOR	TOTAL (cases)	GENOME TYPE	PROJECT	TRAINING SET	TEST SET
Lung	946	miRNA	LUAD	380	95
			LUSC	400	99
		Gene	LUAD	404	71
			LUSC	400	71

TABLE 4.7: Dimensionality of training and test sets for each lung cancer project and genome type

Chapter 5

Results

The **Results** section will contain all the tables, graphs, and data that will be used for the overall evaluation of this method. In the next "discussion" section, comments and considerations will be made on what has been achieved. The next paragraphs will be structured as follows:

1- **Translation in terms of data distribution**.

We will investigate the variability in the data through *Principal Component Analysis* (PCA) and how this was maintained after being translated.

Meanwhile, through a *Cluster Analysis* we will try to highlight differences in terms of tumor subgroup distributions. We will see how some clustering techniques will perform with both the original and translated data. These clusters will also be evaluated as a function of the number of clusters using specific metrics.

2 - **Differential expression analysis**

In this second phase, we will investigate the quality of the translation by looking deeper. As a first step, the translation will be analyzed to understand where it is more faithful to the original. Therefore, *Scatter plots* will be used to compare the original expression levels to the translated expression levels of individual cases.

After that, those genes/miRNAs that differentiate tumor subclasses will be analyzed.

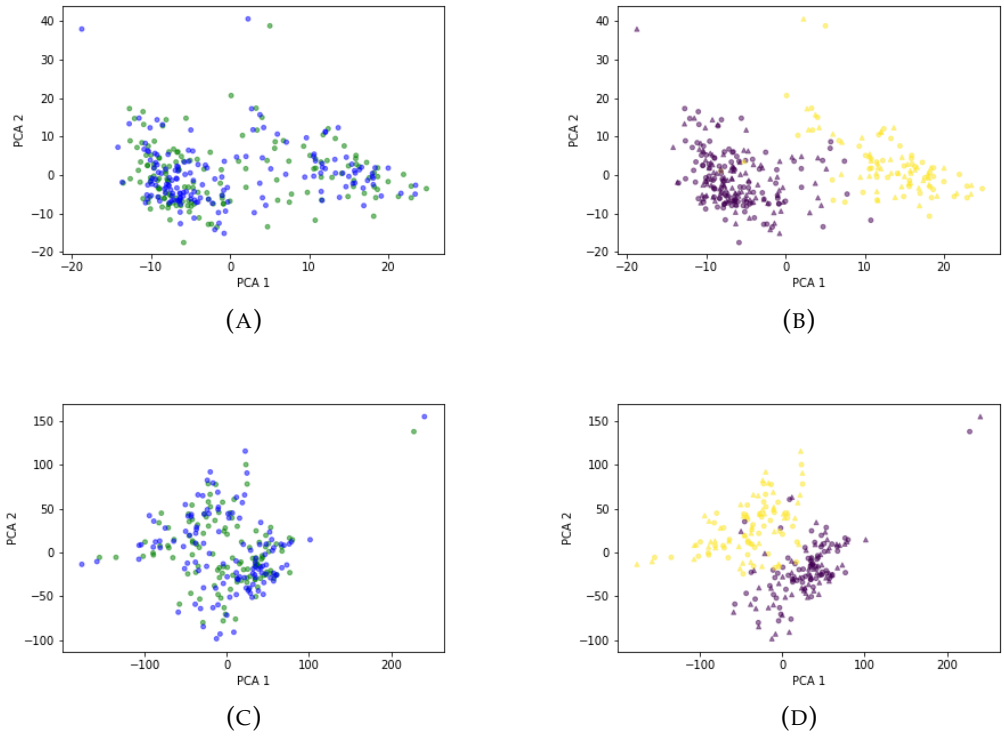
For the differential analysis, a *Statistical test* is used for determining if the mean of the genes/miRNAs as distributions of data are significantly different from the tumor subclasses. Furthermore, in order to explore these particular genes/miRNAs in more detail, some *Heat maps* were generated. This process made it possible to visualize the expression levels that differentiate the subclasses, and the *correlation* between those translated genes/miRNAs and the original ones was calculated.

The results introduced previously were obtained from miRNA and gene expression data from kidney cancer. At the end of this phase, lung cancer miRNA

and gene expression data were used to validate the method and generalize it concerning two types of cancer.

5.0.1 Principal Component Analysis

As a high-level result, the principal component analysis (or PCA) of the translated data and the original data can be shown in Figure 5.1. It is possible to



(A) (B)

(C) (D)

FIGURE 5.1: Principal component analysis of kidney cancer miRNA expression data (upper figures) and gene expression data (down figures): on the left there are (A) original data (green circles) and translated data(blue circles), on the right (B) the same original and translated data (circles and triangles respectively) with the true labels superimposed (KIRC: purple, KIRP: yellow)

create a scatter plot to evidence how the variability of the original data was maintained in the translation, considering the first two principal components. Indeed, PCA allows us to have a reduction in dimensionality while preserving most of the variability of the input data [26]. It is possible to observe how the translated and original data are distributed through the PCA of the two superimposed distributions (Figure 5.12a, Figure 5.12c). In addition, a second PCA is provided to see how variability in the distributions of tumor subclasses also

transferred from the original to the translated data. In this scatter plot, true labels were used for both data types.

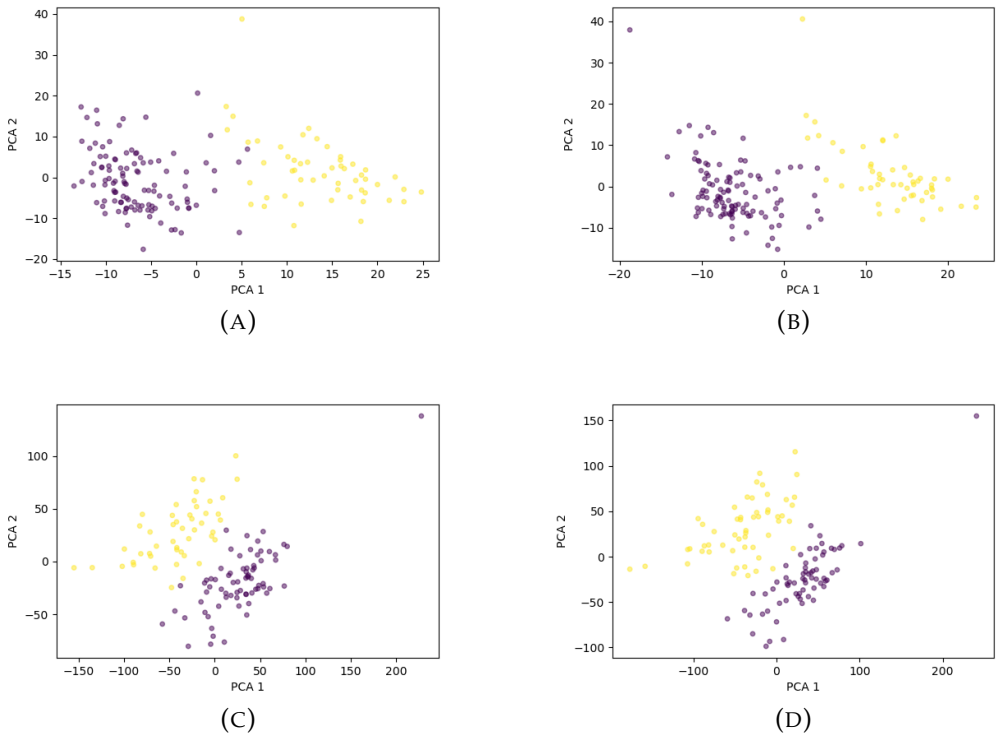


FIGURE 5.2: Clustering analysis applied to kidney cancer miRNA expression data: (A) the Spectral clustering is applied to translated data, (B) the mini-batch K-means is applied to original. The clustering methods recognize the two sub-classes of kidney cancer KIRC (purple circles) and KIRP (yellow circles). The two methods return the best RAND index for both original and translated data.

5.0.2 Clustering

Cluster analysis was used to understand how different clustering techniques interpret the original and translated data distributions. The steps that were followed are:

1- Given the number of true labels in the samples (tumor subgroups = 2), the same clustering techniques with a number of clusters equal to 2 were applied to the original and translated data. The clustering techniques considered in this phase are *K-Means* [27], *Spectral Clustering* [28], *Hierarchical Clustering* [29], *Mini-Batch K-Means* [30] implemented by clustering modules provided by [31]. The clustering techniques considered

reported the best adjusted RAND indexes. The adjusted RAND index

	Clustering Techniques	Adjusted RAND index
Original	Hierarchical Clustering	0.697
	K-Means	0.784
	Spectral Clustering	0.782
	Mini-Batch K-Means	**0.829**

	Clustering Techniques	Adjusted RAND index
Translated	Hierarchical Clustering	0.924
	K-Means	**0.949**
	Spectral Clustering	0.949
	Mini-Batch K-Means	0.924

TABLE 5.1: Adjusted Rand index (ARI) for each type of clustering technique used applied to original (left table) and translated (right table) kidney cancer miRNA expression data

	Clustering Techniques	Adjusted RAND index
Original	Hierarchical Clustering	0.691
	K-Means	0.691
	Spectral Clustering	**0.748**
	Mini-Batch K-Means	0.021

	Clustering Techniques	Adjusted RAND index
Translated	Hierarchical Clustering	0.720
	K-Means	**0.808**
	Spectral Clustering	0.748
	Mini-Batch K-Means	0.021

TABLE 5.2: Adjusted Rand index (ARI) for each type of clustering technique used applied to original (left table) and translated (right table) kidney cancer gene expression data

is a measure of similarity between two clusterings. Generally used to compare true labels from predicted ones in clustering evaluation [32] but also in supervised classification contexts [33]. It takes on values between 0 and 1 inclusive. The RAND would return a value equal to 0 if the label assignment were random, while it is equal to 1 if the two clusterings are identical.

The list showing the clustering techniques used and their calculated RAND indexes are in Table 5.1 and in Table 5.2. In Table 5.1 and Table 5.2 there are data on kidney cancer, miRNA and gene expression data, respectively. The K-Means technique reports the adjusted RAND value in the case of kidney cancer miRNA data, but for the original data, the K-Means technique was used with mini-batch optimization (batch size = 10). At the same time, for the same tumor tissue data, in the case of gene expression data, the highest adjusted RAND is reported by Spectral Clustering for the original data and K-Means for the translated data. Some coincident values between the original and translated data, such as the RANDs returned by Spectral Clustering and Mini-Batch K-Means, both for gene expression data. However, the adjusted RAND values are significantly higher for the translated data than for the original data.

2- The algorithms with the best adjusted RAND were selected. I evaluated the goodness of clustering as a function of the number of clusters

	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Original	Mini-Batch K-Means	**2**	**0.829086**	**0.120906**	**18.97699**	**0.716972**	**0.722307**
		3	0.718394	0.116977	12.82099	0.543454	0.667789
		4	0.380579	0.075979	10.77927	0.324491	0.593276
		5	0.407101	0.065776	9.527051	0.369395	0.711377
		6	0.406526	0.066482	8.531183	0.347083	0.763793
		7	0.414393	0.005685	6.05615	0.307579	0.655762
		8	0.314366	0.050919	7.378385	0.280452	0.772602
		9	0.35209	-0.00294	4.992346	0.309204	0.71002
		10	0.362138	-0.01295	5.010496	0.291965	0.763398

	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Translated	K-Means	**2**	**0.949514**	**0.209972**	**31.85936**	**0.89407**	**0.89407**
		3	0.917579	0.209644	17.91261	0.780324	0.875498
		4	0.526946	0.114943	18.91571	0.490097	0.868805
		5	0.834204	0.181156	12.54027	0.593367	0.829134
		6	0.354293	0.105874	15.42256	0.367298	0.917103
		7	0.801329	-0.07612	8.47799	0.597835	0.830566
		8	0.318843	0.089597	13.15987	0.322746	0.870613
		9	0.345992	0.061616	8.758376	0.34025	0.900265
		10	0.35746	-0.00224	8.146712	0.32507	0.841829

TABLE 5.3: Calculated metrics to evaluate the *Mini-Batch K-Means* applied to the original kidney cancer miRNA data as a function of the number of clusters (upper table) and to evaluate the *K-Means* applied to the translated kidney cancer miRNA data as a function of the number of clusters (lower table).

by calculating some specific metrics. These metrics are the *RAND index*, the *Average Silhouette Width* [34], the *Calinski-Harabasz score* [35], the *Homogeneity* and *Completeness score* [36]. The lists of metrics calculated for evaluating the clustering technique as a function of the number of clusters are shown in Table 5.3 and in Table 5.4 (the graphs that reported the values of the Table 5.3 and in Table 5.4 are in Figure 5.3 5.4, 5.5, 5.6).

What I expect from these metrics is a small discrepancy between the original and translated data.

5.0.3 Differential Expression Analysis

In this second part of the results, various points will be touched upon to analyze translation more deeply. This part can also be divided into several steps:

1- Expression levels of both translated and original individual cases were considered to understand which types of genes/miRNAs the translator predicted best. For this purpose, scatter plots relating original and translated data from a single case were used. The plots related several cases are reported in Figure 5.7 and Figure 5.8 . The predicted regression line was overplotted.

	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Original	Spectral Clustering	**2**	**0.748862**	**0.093026**	**11.6613**	**0.639519**	**0.645393**
		3	0.506113	0.091251	9.548117	0.580936	0.555874
		4	**0.808787**	**0.078886**	**5.649847**	**0.763397**	**0.672517**
		5	0.582654	0.081747	8.35747	0.730116	0.403122
		6	0.584379	0.085973	7.91261	0.758171	0.401099
		7	0.423608	0.081665	5.840382	0.635687	0.311673
		8	0.416279	0.067674	7.191988	0.782185	0.333112
		9	0.391346	0.068814	6.852578	0.782185	0.312001
		10	0.679586	-0.10996	4.229151	0.724801	0.459473

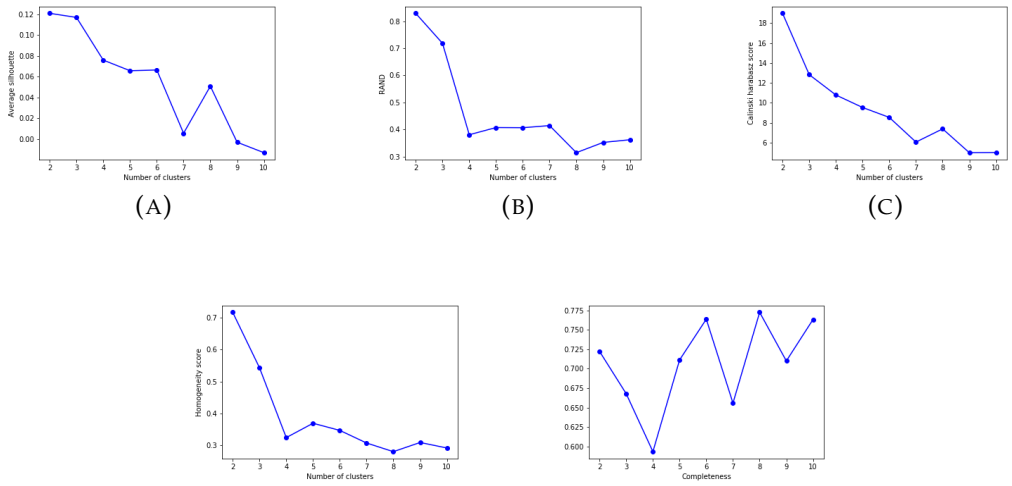
	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Translated	K-Means	**2**	**0.83889**	**0.123413**	**14.66016**	**0.746016**	**0.748939**
		3	0.703327	0.119763	11.81116	0.574464	0.609015
		4	0.76154	0.127551	11.23876	0.569863	0.747301
		5	0.642954	0.096003	7.935659	0.468999	0.673186
		6	0.594246	0.093968	8.685866	0.39685	0.71519
		7	0.360215	0.08239	9.626706	0.321676	0.781518
		8	0.331058	-0.05076	6.457652	0.26528	0.596079
		9	0.377802	0.0473	6.261602	0.313644	0.741459
		10	0.356345	0.086227	8.041752	0.321412	0.793258

TABLE 5.4: Calculated metrics to evaluate the *Spectral Clustering* applied to the original kidney cancer gene data as a function of the number of clusters (upper table) and to evaluate the *K-Means* applied to the translated kidney cancer gene data as a function of the number of clusters (lower table).

2- After this, I went directly to the differential expression analysis. Both gene expression and miRNA expression data were divided by label. In the case of kidney cancer data, the two subclasses are KIRC and KIRP. Thus, it is desired to investigate the difference between the genes/miRNAs that characterize these two subclasses. For this reason, a Welch's t-test was done to both miRNAs and genes between the two labels. Thus, it was possible to obtain a p-value for each gene/miRNA. A diagram illustrating the step just explained is shown in Figure 5.9. It was possible to derive the number of genes and miRNAs differentially expressed within the two subclasses considering different p values. Table 5.5 and Table 5.6 report these numbers for p values less than 5.E-03, 5.E-06, 5.E-12, 5.E-16, 5.E-21, 5.E-26. Also reported is the number of genes and miRNAs that are in common. When the p-value decreases, the number of genes and miRNAs differentially expressed also decreases, but these genes and miRNAs create more differentiation between the subclasses.

3- At this point, the genes, and miRNAs that differentiate the subclasses are available. The questions to be answered are how were the data translated, what correlation is there with the original data, and most importantly, was biological consistency maintained?

Heat maps were generated to visually compare the differentiation and



(A) (B) (C)

(D) (E)

FIGURE 5.3: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) Completeness score for a range of cluster numbers. The clustering technique used is the Mini-Batch K-Means applied to the original kidney miRNAs expression data.

miRNA	*p<5.E-02*	*p<5.E-06*	*p<5.E-12*	*p<5.E-16*	*p<5.E-21*	*p<5.E-26*
Original	452	144	44	20	6	4
Translated	712	231	84	49	23	8
Common	417	143	44	20	6	4

TABLE 5.5: Differential Expression Analysis: each reported value represents the number of miRNAs that are differentially expressed by tumor subclasses under certain p values. The rows show the types of data (original and translated) and also those that are in common.

Gene	*p<5.E-02*	*p<5.E-06*	*p<5.E-12*	*p<5.E-16*	*p<5.E-21*	*p<5.E-26*
Original	10150	3312	827	188	24	3
Translated	11326	4511	1534	519	160	31
Common	9476	3166	804	185	24	3

TABLE 5.6: Differential Expression Analysis: each reported value represents the number of genes that are differentially expressed by tumor subclasses under certain p values. The rows show the types of data (original and translated) and those in common.

how it results in the original and translated data. Heat maps are particular diagram that shows the intensity of each element of a matrix with a color. The maximum and minimum values set the ends of the color

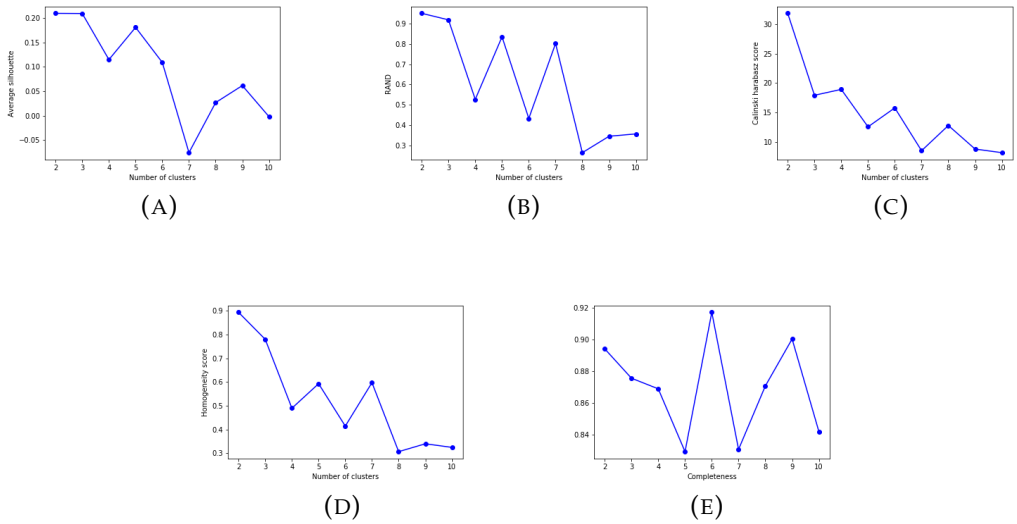


FIGURE 5.4: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) Completeness score for a range of cluster numbers. The clustering technique used is the K-Means applied to the translated kidney miRNAs expression data.

scale. Typically, dark colors are associated with low-intensity values and light colors with high-intensity values. In these maps, we find the same cases in a row and the same genes or miRNAs in columns, the visual comparison is facilitated.

The heat maps are shown in Figure 5.10 and in Figure 5.11.

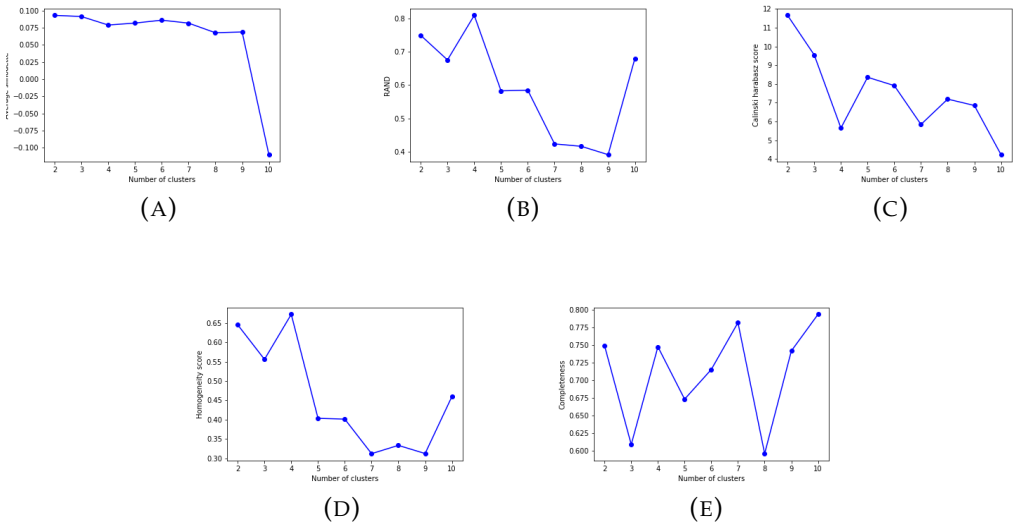
4- The correlation between the original and translated data can be seen qualitatively from the heat maps. To obtain this data quantitatively, the Pearson correlation between the original and translated genes/miRNAs was calculated as

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \tag{5.1}$$

where cov function is the covariance of the two random variables X and Y, and σ_x, σ_y are the standard deviation of X and Y respectively. If ρ=1, means that X and Y have a perfect correlation meanwhile if ρ=0 represent the absense of a relation between the two variables.

The following data report the total number of translated genes/miRNAs that have a significant correlation with the original ones:

- out of a total of 1530 miRNAs, **807** translated miRNAs have a Pearson's correlation greater than 0.75 (p value<0.05).



(A) (B) (C)

(D) (E)

FIGURE 5.5: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) Completeness score for a range of cluster numbers. The clustering technique used is the Mini-Batch K-Means applied to the original kidney genes expression data.

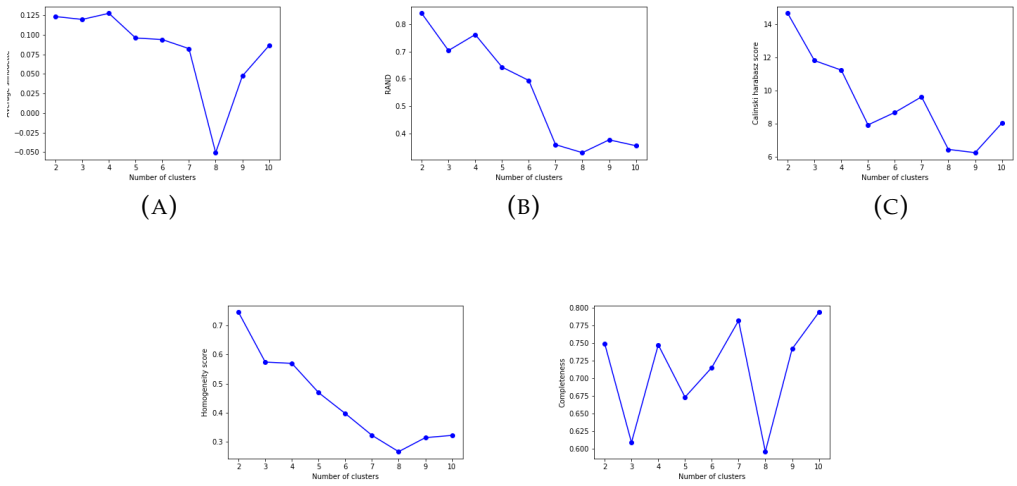
- out of a total of 19372 genes, **15475** translated genes have a Pearson's correlation greater than 0.75 (p value<0.05).

The question to be answered is how many and which of these genes and miRNAs differentiate tumor subclasses? The numbers that summarize this analysis are:

- out of a total of 417 differentially expressed miRNAs (p<5E-02, see Table 5.5), **318** translated miRNAs have a Pearson's correlation of 0.75 (p value<0.05).
- out of a total of 9476 differentially expressed genes (p<5E-02, see Tab 5.6), **7030** translated genes have a Pearson's correlation of 0.75 (p value<0.05).

Then, the p-values resulting from the T-test and the correlation values are considered for the genes/miRNAs in the original dataset and the translated one. The idea is to evaluate how many genes/miRNAs are differentially expressed in the two classes with respect to the correlation and p-value values changes.

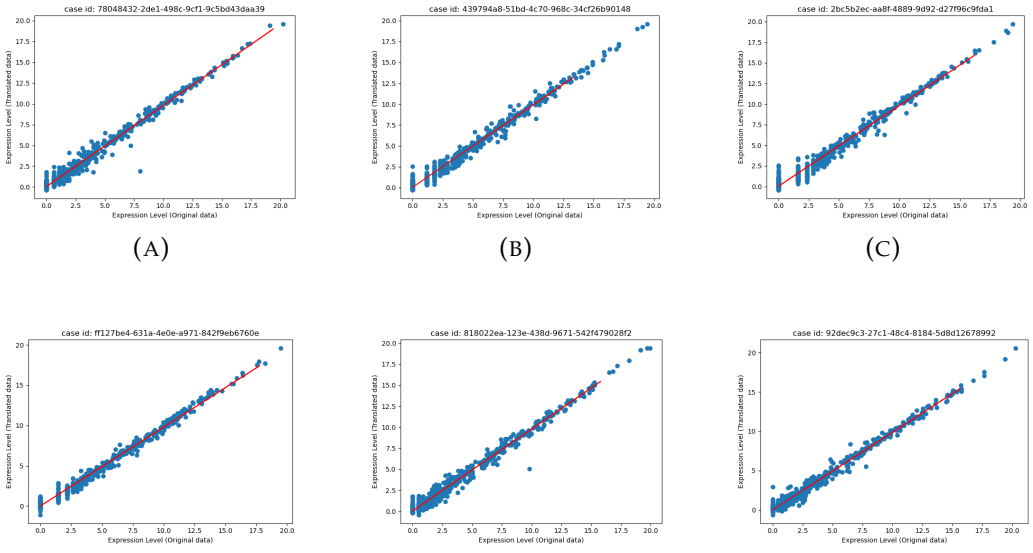
Table 5.7 shows how the number of genes varies while Table 5.8 shows how the number of miRNAs varies.



(A) (B) (C)

(D) (E)

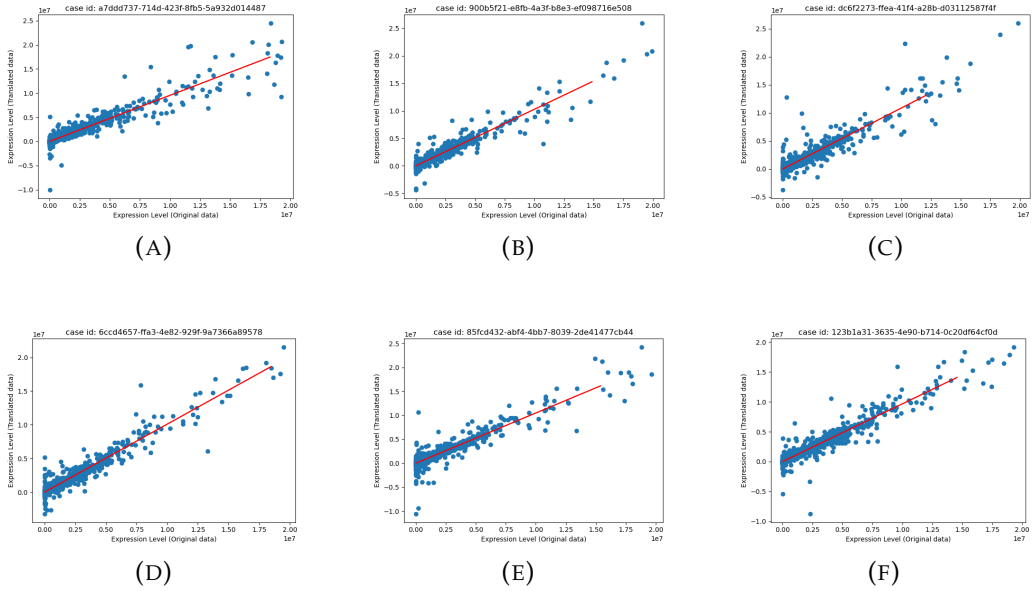
FIGURE 5.6: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) the Completeness score for a range of cluster numbers. The clustering technique used is the K-Means applied to the translated kidney genes expression data.



(A) (B) (C)

(D) (E) (F)

FIGURE 5.7: Scatter plots of kidney miRNA expression data. Six different random cases are used. The graphs show the original data in the abscissa and the translated data in the ordinate.



(A) (B) (C)

(D) (E) (F)

FIGURE 5.8: Scatter plots of kidney gene expression data. Six different random cases are used. The graphs show the original data in the abscissa and the translated data in the ordinate.

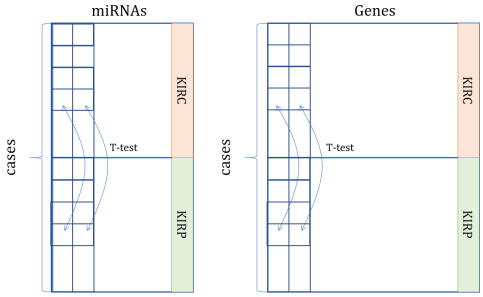
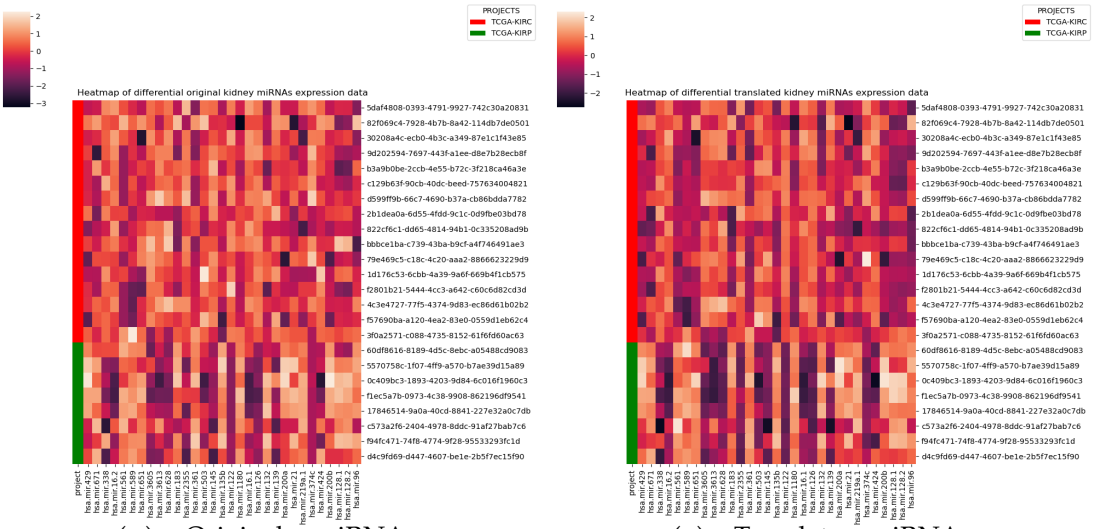


FIGURE 5.9: Welch's T-test application

	p<5E-2	**p<5E-6**	**p<5E-12**	p<5E-16	**p<5E-21**
Total*	9476	3166	609	185	24
$\rho > 0.75$	7030	2846	590	180	24
$\rho > 0.80$	6052	2608	573	180	24
$\rho > 0.85$	4613	2162	510	165	24
$\rho > 0.90$	2670	1428	363	128	17
$\rho > 0.95$	695	377	99	29	3

TABLE 5.7: The number of genes that differentiate the subclasses in relation with the correlation's values.

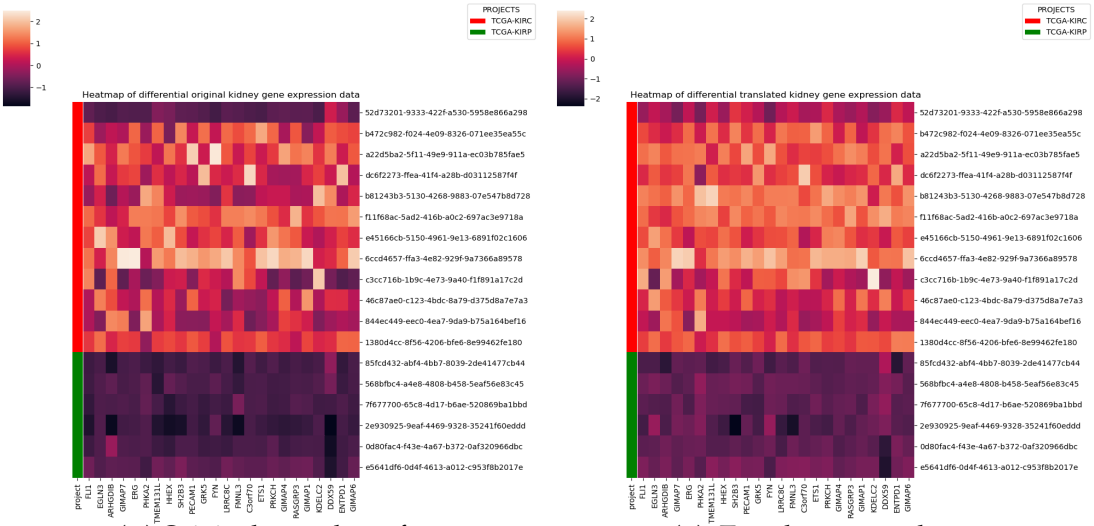
*The total numbers is the number of *common* genes/miRNAs that differentiate the subclasses, see Table 5.5



(A) Original miRNA data of kidney cancer

(B) Translate miRNA data of kidney cancer

FIGURE 5.10: Heatmap of differential miRNAs



(A) Original gene data of kidney cancer

(B) Translate gene data of kidney cancer

FIGURE 5.11: Heatmap of differential genes

So fixed a p-value, the translated genes and miRNAs that most differentiate the subclasses and have a correlation greater than 0.75 with the corresponding original data are shown in Tables 5.9.

According to Table 5.5 and Table 5.6, the Table 5.9 shows the 41 miRNAs obtained by p<5E-12 p-value selecting in the differentiation phase and a ρ>0.75 at left and the 24 genes obtained by p<5E-21 p-value selecting in the differentiation phase and a ρ>0.75 at right.

	p<5E-2	**p<5E-6**	**p<5E-12**	p<5E-16	**p<5E-21**
Total*	417	143	44	20	6
$\rho > 0.75$	318	132	41	20	6
$\rho > 0.80$	263	118	37	18	6
$\rho > 0.85$	204	96	33	18	6
$\rho > 0.90$	105	54	23	15	5
$\rho > 0.95$	19	14	9	7	1

TABLE 5.8: The number of miRNAs that differentiate the subclasses in relation with the correlation's values.

*The total numbers is the number of *common* genes/miRNAs that differentiate the subclasses, see Table 5.5

5.0.4 Validation

In this part, I validate what precedes this paragraph. Validation is based on the application of the proposed method to another tumor tissue data. The tumor tissue selected for validation is the lung tissue because it is composed of two populous tumor classes, LUSC and LUAD tumors. The data types will remain unchanged. The dimensionality of the validation data is reported in Table 4.7. The next few pages will be reserved to explain the same process described in the previous paragraphs. For more details about the following results, refer to the section **Results**.

PCAs are depicted below to see how variability in the original data transferred to the translated data. PCAs with the true label superimposed are also available in Figure 5.12 to see how intraclass variability was also maintained after the translation. RAND indexes for the various clustering techniques (number of clusters=2) were calculated. In this case, the clustering techniques that reported the best RAND are K-Means, both original and translated miRNA expression data and also for original gene expression data, while Spectral clustering for translated gene expression data. The list showing the clustering techniques used and their calculated RAND indexes are in Tables 5.10 and in Tables 5.11. The clustering techniques that reported the best RAND were evaluated by various metrics. The list of evaluation metrics used as a function of the number of clusters are shown in Tables 5.13 and Tables 5.12 (the graphs that reported the values of the Table 5.12 and Table 5.13 are in Figure 5.14, 5.15, 5.16, 5.17). The K-Means returned the best adjusted RAND score for the miRNA, both original and translated data. At the same time, the K-means reported the best RAND for the original gene expression data, while the Spectral Clustering returned the best RAND for the translated gene expression data.

The scatter plots reported the relation between original and translated data provided by individual cases are shown in Figure 5.18 and Figure 5.19.

The results related to Welch's T-test done for both miRNAs and genes between the two subgroups label are shown in Table 5.14 and Table 5.15.

The heatmaps that shown the intensity of lung miRNA and gene expression data, both original and translated types, are reported in Figure 5.20 and Figure 5.21.

The relationship between the differentiation made by the T-test and the

miRNAs	Correlation (ρ)	p-values
hsa.mir.561	0.882998	1.12E-08
hsa.mir.3913.1	0.856304	9.42E-08
hsa.mir.16.1	0.958688	1.70E-13
hsa.mir.122	0.918144	2.60E-10
hsa.mir.628	0.94959	1.46E-12
hsa.mir.2355	0.834361	4.04E-07
hsa.mir.183	0.946077	3.01E-12
hsa.mir.429	0.942948	5.51E-12
hsa.mir.145	0.962912	5.30E-14
hsa.mir.1271	0.929724	5.13E-11
hsa.mir.1180	0.934153	2.56E-11
hsa.mir.126	0.981014	3.65E-17
hsa.mir.132	0.782689	6.18E-06
hsa.mir.3605	0.903421	1.49E-09
hsa.mir.424	0.859526	7.45E-08
hsa.mir.200a	0.97655	3.65E-16
hsa.mir.200b	0.976617	3.53E-16
hsa.mir.128.2	0.799139	2.82E-06
hsa.mir.671	0.861704	6.34E-08
hsa.mir.96	0.915777	3.52E-10
hsa.mir.576	0.853618	1.14E-07
hsa.mir.651	0.819033	9.90E-07
hsa.mir.338	0.936714	1.67E-11
hsa.mir.16.2	0.956571	2.92E-13
hsa.mir.219a.1	0.907447	9.54E-10
hsa.mir.589	0.850516	1.41E-07
hsa.mir.21	0.960774	9.72E-14
hsa.mir.361	0.948893	1.69E-12
hsa.mir.195	0.880605	1.38E-08
hsa.mir.128.1	0.797598	3.05E-06
hsa.mir.210	0.944425	4.16E-12
hsa.mir.10b	0.879172	1.56E-08
hsa.mir.215	0.972561	2.01E-15
hsa.mir.503	0.78965	4.48E-06
hsa.mir.374c	0.883678	1.05E-08
hsa.mir.1307	0.831875	4.70E-07
hsa.mir.3613	0.884119	1.01E-08
hsa.mir.17	0.837345	3.35E-07
hsa.mir.143	0.920427	1.92E-10
hsa.mir.135b	0.928291	6.36E-11
hsa.mir.139	0.956741	2.80E-13

Genes	Correlation (ρ)	p-values
SH2B3	0.884007882	1.14E-06
EGLN3	0.923617799	4.58E-08
GIMAP1	0.917970765	7.95E-08
ERG	0.92049666	6.24E-08
DDX59	0.902175309	3.09E-07
ENTPD1	0.92406171	4.37E-08
FLI1	0.921623677	5.59E-08
LRRC8C	0.949070861	1.94E-09
C3orf70	0.943954417	4.10E-09
PECAM1	0.89557233	5.10E-07
ARHGDIB	0.883605479	1.17E-06
KDELC2	0.869188069	2.84E-06
PHKA2	0.966450311	7.26E-11
GIMAP4	0.938933287	8.02E-09
FMNL3	0.877678277	1.71E-06
RASGRP3	0.928151355	2.85E-08
GIMAP7	0.921483415	5.66E-08
GRK5	0.933377524	1.58E-08
ETS1	0.951139467	1.40E-09
PRKCH	0.889965953	7.61E-07
HHEX	0.905744046	2.32E-07
TMEM131L	0.950307667	1.60E-09
GIMAP6	0.945081715	3.50E-09
FYN	0.890223128	7.48E-07

TABLE 5.9: The tables contain Pearson correlation values between the original and translated miRNAs (left) and genes (right) that differentiate the two subclasses of kidney cancer with p-values less than 5E-12 and 5E-21 respectively. Only correlation values greater than 0.75 were considered. Out of a total of 44 miRNAs differentially espressed (see 5.5), 40 translated miRNAs were included. Out of a total of 24 genes differentially espressed (see 5.6), all 24 traslated genes were included.

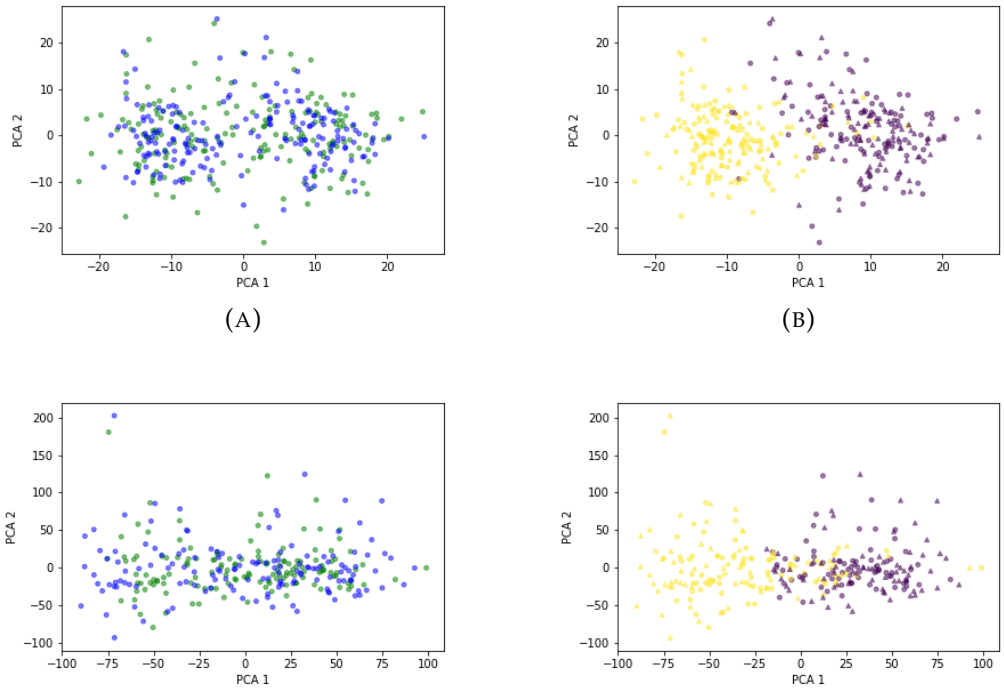


FIGURE 5.12: Principal component analysis of lung cancer miRNA expression data (upper figures) and gene expression data (down figures): on the left there are (A) original data (green circles) and translated data(blue circles), on the right (B) the same original and translated data (circles and triangles respectively) with the true labels superimposed (LUAD: purple, LUSH: yellow)

correlation between the original and the translated data can be found in Table 5.8 and Table 5.17.

In Table 5.18 is reported the Pearson correlation between miRNAs and genes both translated and original. Those miRNAs/genes differentiate the subclasses of tumor, information reported by Welch's T-test.

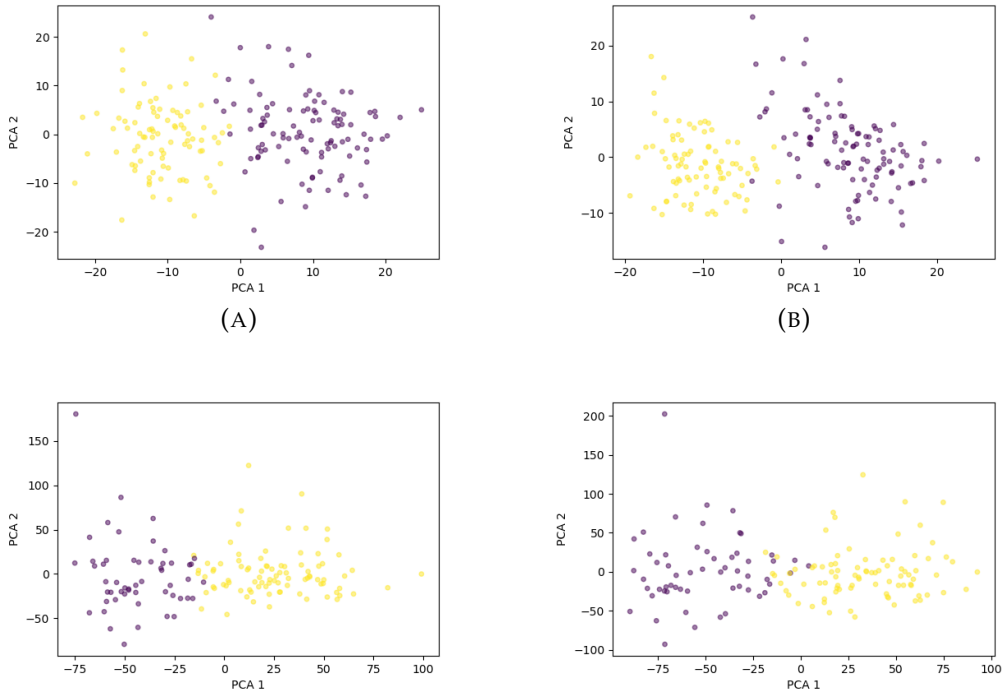


FIGURE 5.13: Clustering analysis applied to lung cancer miRNA expression data (upper figures) and gene expression data (lower figures): the K-Means is applied to both original (A) and translated (B) miRNA expression data and also to original gene expression data (C) whilst the Spectral Clustering is applied to translated gene expression data. The clustering methods recognize the two sub-classes of lung cancer LUAD (purple circles) and LUSH (yellow circles). The two methods return the best RAND index for both original and translated data.

	Clustering Techniques	Adjusted RAND index
Original	Hierarchical Clustering	0.725
	K-Means	**0.743**
	Spectral Clustering	0.725
	Mini-Batch K-Means	0.743

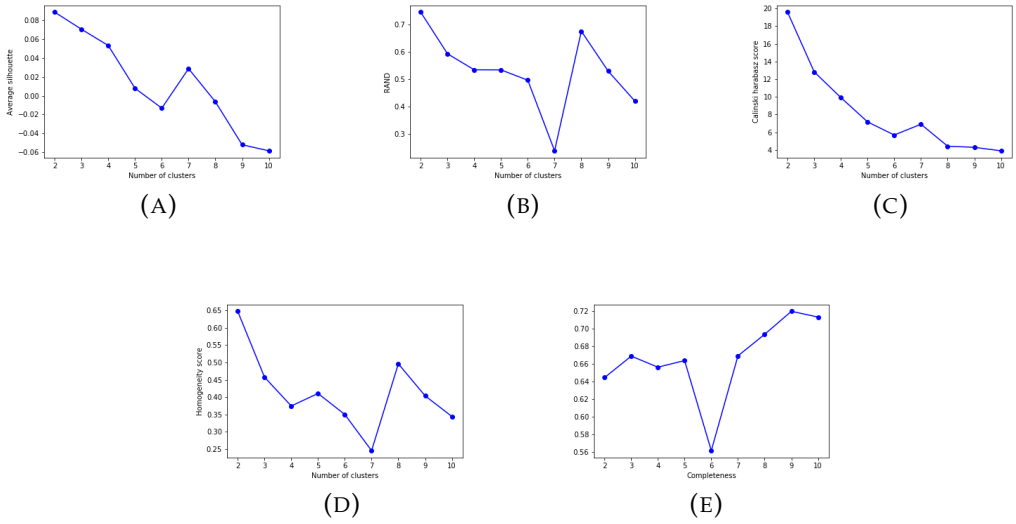
	Clustering Techniques	Adjusted RAND index
Translated	Hierarchical Clustering	0.837
	K-Means	**0.837**
	Spectral Clustering	0.818
	Mini-Batch K-Means	0.762

TABLE 5.10: Adjusted Rand index (ARI) for each type of clustering technique used applied to original (left table) and translated (right table) lung cancer miRNA expression data

	Clustering Techniques	Adjusted RAND index
Original	Hierarchical Clustering	0.472
	K-Means	**0.575**
	Spectral Clustering	0.533
	Mini-Batch K-Means	-0.003

	Clustering Techniques	Adjusted RAND index
Translated	Hierarchical Clustering	0.533
	K-Means	0.619
	Spectral Clustering	**0.665**
	Mini-Batch K-Means	0.173

TABLE 5.11: Adjusted Rand index (ARI) for each type of clustering technique used applied to original (left table) and translated (right table) lung cancer gene expression data



(A) (B) (C)

(D) (E)

FIGURE 5.14: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) the Completeness score for a range of cluster numbers. The clustering technique used is the K-Means applied to the original lung miRNAs expression data.

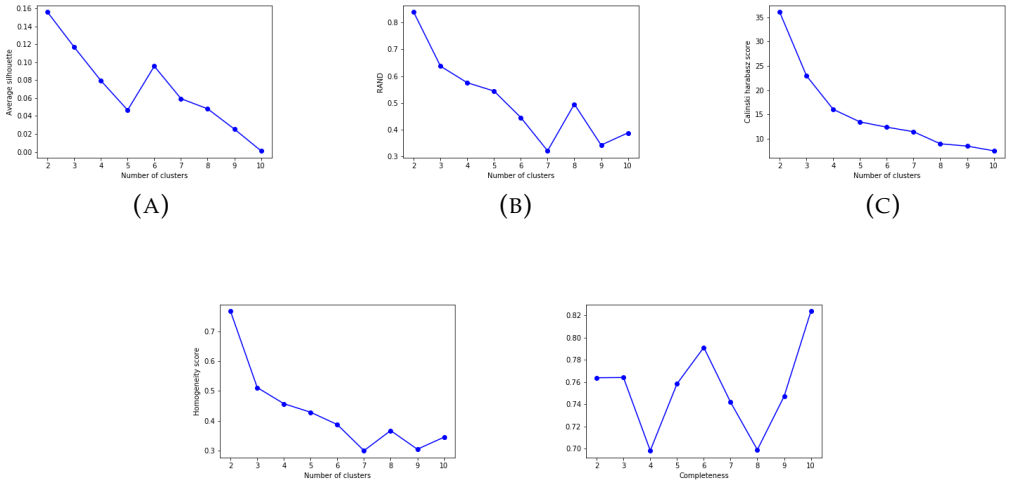


FIGURE 5.15: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) the Completeness score for a range of cluster numbers. The clustering technique used is the K-Means applied to the translated lung miRNAs expression data.

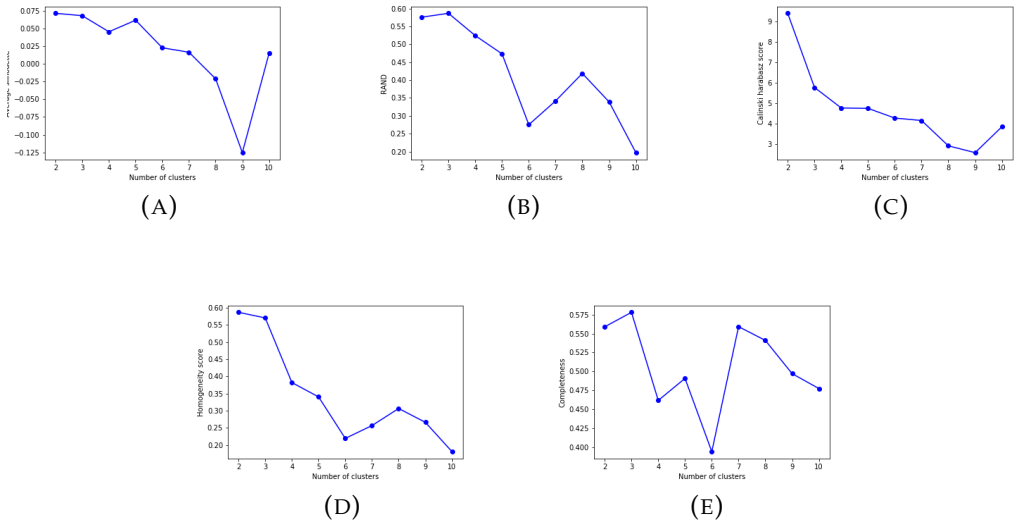
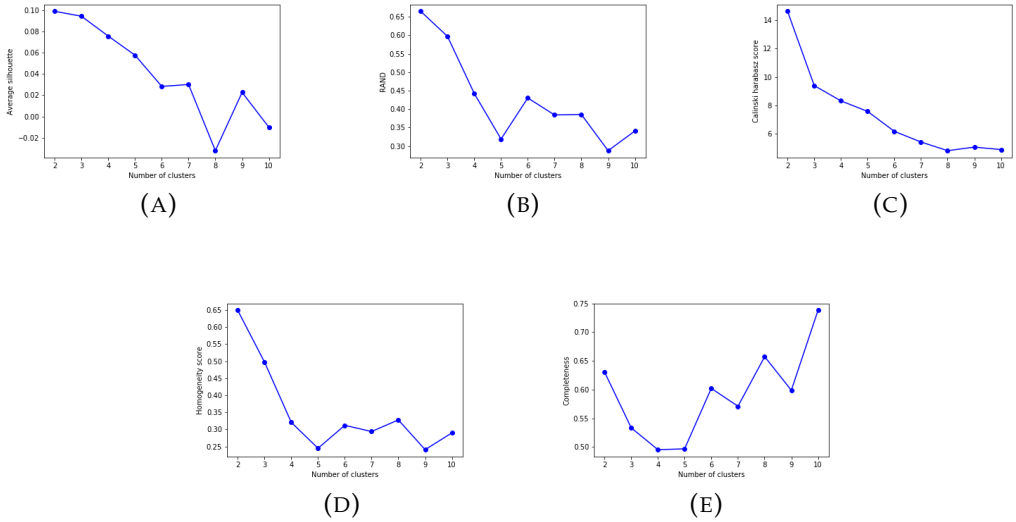


FIGURE 5.16: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) the Completeness score for a range of cluster numbers. The clustering technique used is the K-Means applied to the original lung gene expression data.



(A) (B) (C)

(D) (E)

FIGURE 5.17: The graphs reported (A) the silhouette, (B) the RAND index, (C) the Calinski Harabasz score, (D) the Homogeneity score and (E) the Completeness score for a range of cluster numbers. The clustering technique used is the Spectral Clustering applied to the translated lung gene expression data.

	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Original	K-Means	**2**	**0.743695**	**0.08829**	**19.5834**	**0.647562**	**0.644709**
		3	0.591785	0.070218	12.82455	0.457633	0.669038
		4	0.533148	0.053025	9.91686	0.37425	0.656496
		5	0.532879	0.007812	7.168435	0.410399	0.664021
		6	0.495312	-0.01318	5.685169	0.349756	0.561863
		7	0.226797	0.02672	6.919867	0.244967	0.665453
		8	0.674237	-0.00641	4.419881	0.49605	0.693756
		9	0.528674	-0.05186	4.29381	0.403544	0.719827
		10	0.419002	-0.0583	3.897953	0.343939	0.713363

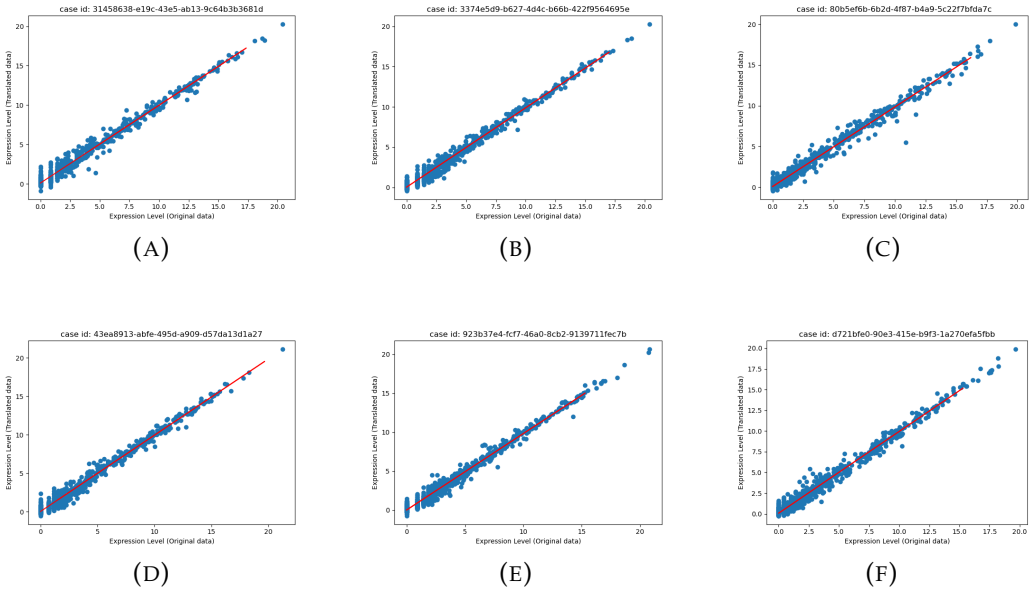
	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Translated	K-Means	**2**	**0.837819**	**0.155762**	**36.05297**	**0.768139**	**0.763708**
		3	0.636909	0.116777	22.98539	0.511452	0.763933
		4	0.575263	0.079401	16.07102	0.457058	0.69812
		5	0.544302	0.046584	13.49185	0.428392	0.758342
		6	0.445959	0.095545	12.41482	0.38684	0.791026
		7	0.287759	0.032908	11.94143	0.284627	0.748333
		8	0.49572	0.047996	8.995248	0.366631	0.698933
		9	0.343275	0.025422	8.528527	0.303832	0.747019
		10	0.38857	0.00117	7.552718	0.34428	0.823516

TABLE 5.12: Calculated metrics to evaluate the *K-Means* applied to the original lung cancer miRNA data as a function of the number of clusters (upper table) and to evaluate the *Spectral Clustering* applied to the translated lung cancer miRNA data as a function of the number of clusters (lower table).

	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Original	K-Means	**2**	**0.575636**	**0.071239**	**9.411687**	**0.586364**	**0.558959**
		3	0.58635	0.067917	5.783055	0.570112	0.57808
		4	0.524081	0.045229	4.775725	0.381841	0.461777
		5	0.47307	0.061513	4.756729	0.340101	0.490882
		6	0.275003	0.022603	4.283062	0.219261	0.394167
		7	0.34149	0.016321	4.162227	0.256659	0.559207
		8	0.418383	-0.02126	2.923523	0.306389	0.541017
		9	0.338805	-0.12544	2.58828	0.26616	0.497135
		10	0.196879	0.014519	3.854418	0.180688	0.47748

	Clustering Techniques	Number of Clusters	Adjusted RAND index	Avg. Silhouette Width	Calinski-Harabasz Score	Homogeneity Score	Completeness Score
Translated	Spectral Clustering	**2**	**0.665043**	**0.098974**	**14.64232**	**0.64875**	**0.630525**
		3	0.597226	0.094341	9.392246	0.497453	0.533198
		4	0.442474	0.075636	8.310124	0.320716	0.495237
		5	0.318837	0.05769	7.567373	0.244872	0.496686
		6	0.430133	0.028267	6.157142	0.311956	0.601981
		7	0.384145	0.030071	5.402818	0.2938	0.570848
		8	0.385554	-0.03225	4.790545	0.327814	0.657222
		9	0.287361	0.022615	5.042153	0.240722	0.598686
		10	0.340705	-0.01021	4.866959	0.289835	0.737925

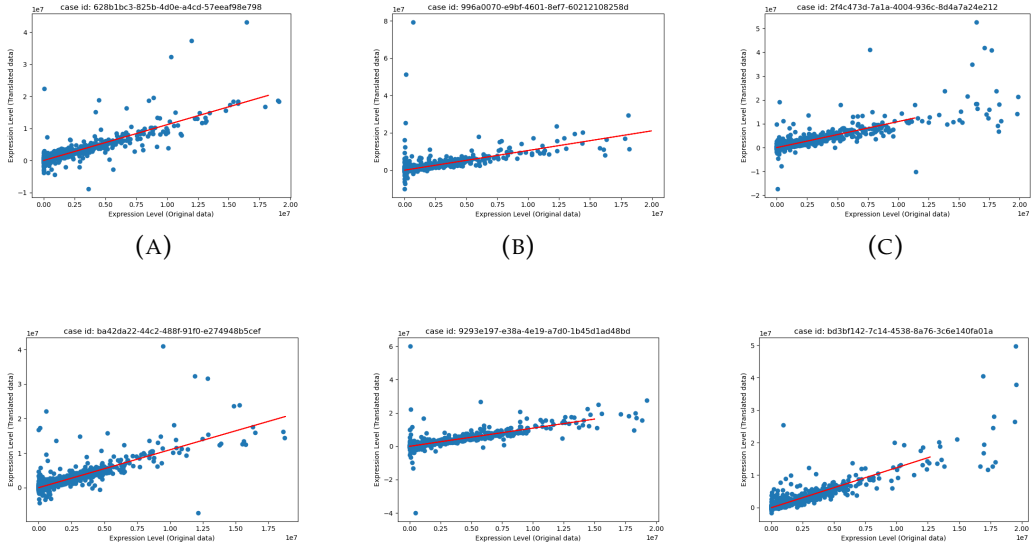
TABLE 5.13: Calculated metrics to evaluate the *K-Means* applied to the original lung cancer gene data as a function of the number of clusters (upper table) and to evaluate the *Spectral Clustering* applied to the translated lung cancer gene data as a function of the number of clusters (lower table).



(A) (B) (C)

(D) (E) (F)

FIGURE 5.18: Scatter plots of lung miRNA expression data. Six different random cases are used. The graphs show the original data in the abscissa and the translated data in the ordinate.



(A) (B) (C)

(D) (E) (F)

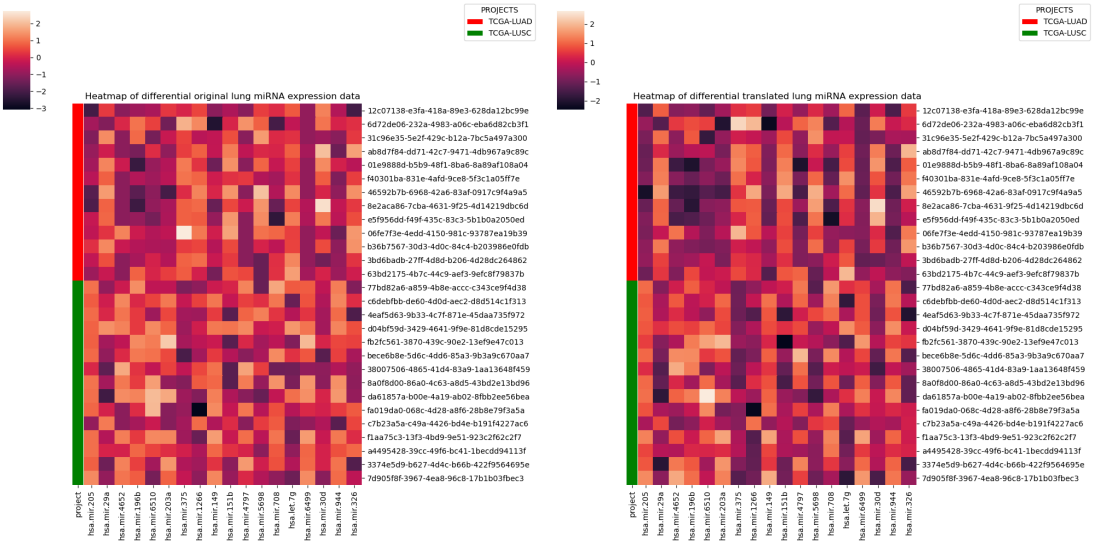
FIGURE 5.19: Scatter plots of lung gene expression data. Six different random cases are used. The graphs show the original data in the abscissa and the translated data in the ordinate.

miRNA	*p<5.E-02*	*p<5.E-06*	*p<5.E-12*	*p<5.E-16*	*p<5.E-21*	*p<5.E-26*
Original	484	105	27	13	7	5
Translated	773	221	77	45	26	13
Common	452	105	27	13	7	5

TABLE 5.14: Differential Expression Analysis: each reported value represents the number of miRNAs that are differentially expressed by tumor subclasses under certain p values. The rows show the types of data (original and translated) and also those that are in common.

Gene	*p<5.E-02*	*p<5.E-06*	*p<5.E-12*	*p<5.E-16*	*p<5.E-21*	*p<5.E-26*
Original	9108	1803	144	8	0	0
Translated	11364	3942	844	251	43	5
Common	8653	1769	141	7	0	0

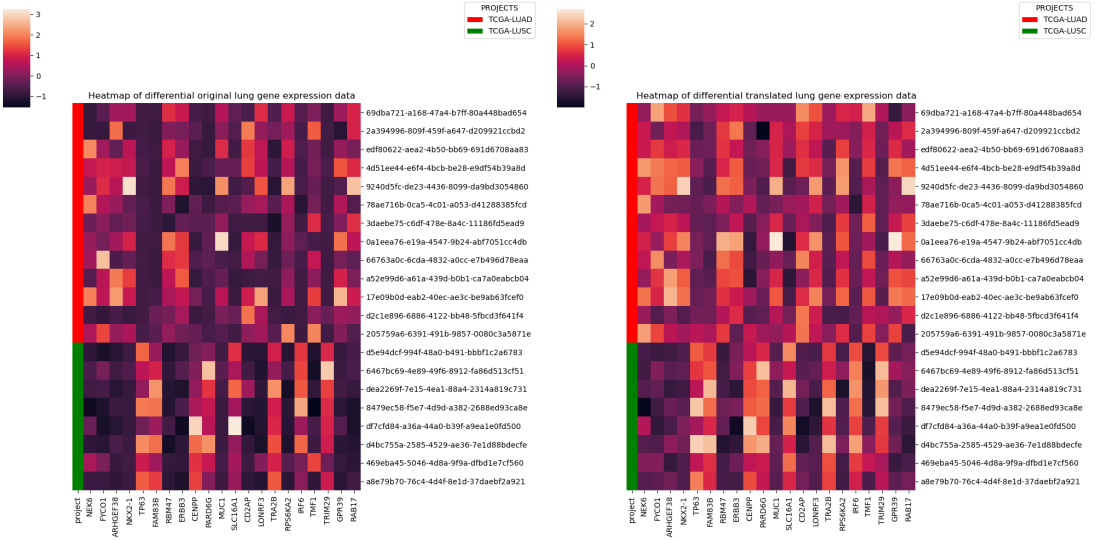
TABLE 5.15: Differential Expression Analysis: each reported value represents the number of genes that are differentially expressed by tumor subclasses under certain p values. The rows show the types of data (original and translated) and also those that are in common.



(A) Original miRNA data of lung cancer

(B) Translate miRNA data of lung cancer

FIGURE 5.20: Heatmap of differential miRNA



(A) Original gene data of lung cancer

(B) Translate gene data of lung cancer

FIGURE 5.21: Heatmap of differential gene

	p<5E-2	**p<5E-6**	**p<5E-12**	**p<5E-16**	**p<5E-21**
Total*	452	105	27	13	7
ρ>0.75	350	97	26	13	7
ρ>0.8	291	83	25	12	6
ρ>0.85	190	59	18	10	6
ρ>0.9	93	28	10	8	4
ρ>0.95	10	5	0	0	0

TABLE 5.16: The number of miRNAs that differentiate the subclasses in relation with the correlation's values.

*The total numbers is the number of *common* miRNAs that differentiate the subclasses, see Table 5.14

	p<5E-2	**p<5E-6**	**p<5E-12**	**p<5E-16**	**p<5E-21**
Total*	8653	1769	141	7	0
ρ>0.75	5006	1310	129	7	0
ρ>0.8	3749	1061	113	5	0
ρ>0.85	2349	717	88	5	0
ρ>0.9	1002	302	49	5	0
ρ>0.95	127	29	4	1	0

TABLE 5.17: The number of genes that differentiate the subclasses in relation with the correlation's values.

*The total numbers is the number of *common* genes that differentiate the subclasses, see Table 5.15

miRNAs	Correlation (ρ)	p-values
hsa.mir.29a	0.933895	4.02E-13
hsa.mir.6510	0.902176	5.46E-11
hsa.mir.149	0.931405	6.41E-13
hsa.mir.93	0.889943	2.35E-10
hsa.let.7g	0.875841	1.04E-09
hsa.mir.26a.2	0.867775	2.24E-09
hsa.mir.181b.1	0.818977	9.95E-08
hsa.mir.944	0.85903	4.90E-09
hsa.mir.4652	0.844316	1.63E-08
hsa.mir.4709	0.803713	2.60E-07
hsa.mir.6512	0.836043	3.04E-08
hsa.mir.151b	0.89484	1.34E-10
hsa.mir.769	0.906509	3.11E-11
hsa.mir.375	0.88415	4.43E-10
hsa.mir.203a	0.92756	1.27E-12
hsa.mir.4728	0.829434	4.89E-08
hsa.mir.6499	0.947948	1.95E-14
hsa.mir.326	0.761207	2.56E-06
hsa.mir.205	0.93199	5.76E-13
hsa.mir.1266	0.901349	6.07E-11
hsa.mir.708	0.823544	7.34E-08
hsa.mir.30d	0.914208	1.06E-11
hsa.mir.26a.1	0.870941	1.67E-09
hsa.mir.5698	0.810025	1.77E-07
hsa.mir.196b	0.907629	2.68E-11
hsa.mir.3662	0.859866	4.56E-09

Genes	Correlation (ρ)	p-values
NECTIN1	0.886757	8.63E-08
MUC1	0.913878	7.14E-09
RPS6KA2	0.821077	5.09E-06
ABCC3	0.818035	5.89E-06
ARRB1	0.883049	1.15E-07
WDR53	0.910224	1.04E-08
REPS1	0.839891	1.91E-06
ZFP64	0.832516	2.85E-06
PKP1	0.963235	2.67E-12
MMS22L	0.797617	1.49E-05
GPR39	0.776149	3.54E-05
SMIM14	0.897357	3.54E-08
ERBB3	0.857698	6.72E-07
RAB17	0.895992	3.99E-08
SGMS2	0.809815	8.67E-06
NKX2-1	0.922923	2.58E-09
DQX1	0.832966	2.78E-06
TRA2B	0.901377	2.46E-08
TRIM29	0.915256	6.16E-09
IRF6	0.905929	1.60E-08
CENPP	0.859	6.19E-07
TMEM125	0.89985	2.83E-08
NEK6	0.760331	6.33E-05
RBM47	0.930292	1.02E-09
FANCE	0.880166	1.44E-07
RORC	0.927698	1.43E-09
LONRF3	0.881169	1.33E-07
TPCN1	0.932749	7.34E-10
OCLN	0.911825	8.86E-09
FAM83B	0.943739	1.41E-10
PERP	0.890078	6.59E-08
SENP5	0.921622	3.01E-09
ACSL5	0.920417	3.46E-09
SLC16A1	0.822063	4.85E-06
DDAH1	0.865849	3.96E-07
ARHGEF38	0.857773	6.68E-07
FBXO45	0.920915	3.27E-09
TP63	0.953547	2.37E-11
ACOX2	0.882897	1.17E-07
PARD6G	0.919507	3.84E-09
FYCO1	0.815722	6.58E-06
SUCLG2	0.79769	1.48E-05

TABLE 5.18: The tables contain Pearson correlation values between the original and translated miRNAs (left) and genes (right) that differentiate the two subclasses of lung cancer with p-values less than 5E-12 and 5E-14 respectively. Only correlation values greater than 0.75 were considered. Out of a total of 27 miRNAs differentially espressed (see 5.14), 26 translated miRNAs were included. Out of a total of 45 genes differentially espressed (see 5.15), 42 traslated genes were included.

Chapter 6

Discussions

The following chapter is related to the discussions of the thesis. An interpretation of the obtained results will be provided.

The following steps will discuss all the results obtained concerning the tables, figures, and graphs in the Results section.

1- The principal component analysis shows that the translated data distributions are in the area where the original data are spread out. Looking at the scatter plots in Figure 5.1 and Figure 5.12 it is possible to see that the variability of the original data was maintained after the translation as the intra-class variability. In other words, translated data preserve variability in tumor subclasses. The overlap of cases in the PCA is not punctual, but considering the amount of translated genes/miRNA, I can be satisfied from a graphical viewpoint.

2- The following step was the cluster analysis. The Tables 5.1, 5.2, 5.10, 5.11 containing the adjusted RAND data for each clustering technique tell which of them comes closest to the true label and quantifies this closeness. I collected clustering techniques evaluation metrics to describe the distributions of the original and translated data. In order to compare them, metrics were calculated for a range of cluster numbers. The adjusted rand obtained from the translated data is generally higher than the original data. There are few coincident values between the original and the translated kidney cancer data as the RANDs returned by Spectral Clustering and Mini-Batch K-Means, for both gene expression data. At the same time, the metrics of the original data are also generally lower than those of the translated data. However, the Figures 5.3, 5.4 plots show that the metrics reported from both the original and translated kidney cancer data follow a similar trend. The previous observation can be extended to the other pairs of Figures 5.5, 5.6, and 5.14, 5.15 and 5.16, 5.17.

   In addition, the average Silhouette score and the Calinski-Harabasz score reported the best value for a number of clusters equal to 2 in all possible cases.

3- The analysis of individual patients is characterized by the scatter plots shown in Figure 5.7, 5.8, 5.18, 5.19. These plots provide a significant result: observing the distribution of the data concerning the regression line, one can see that the translation from genes-to-miRNAs is better than the translation from miRNAs-to-genes. In addition, translation from genes to miRNAs improves as the expression value of the miRNA to be translated increases. In contrast, this trend is not visible in the case of translated genes.

4- The first step of the differential expression analysis was to investigate if the same genes/miRNAs that differentiate tumor subclasses in the original dataset are the same as those in the translated dataset. This information is reported in the Table 5.5, 5.6, 5.14, 5.15.

On average, over 95% of the original miRNAs that differentiate the classes are among the translated ones, while over 96% of the original class-differentiating genes fall within the translated ones. However, the translator has produced new differential genes/miRNAs. Nevertheless, I investigated the differential genes/miRNAs in common between the original and translated data. The question was, how were these genes/miRNAs translated? The heat maps were generated. This qualitative visualization shows a difference between the two types of translations: it can be noticed that the gene-to-miRNA translation remains more faithful to the original (Figure 5.10, Figure 5.20) while the other translation (Figure 5.11, Figure 5.21) produced genes that differentiate the two classes more distinctly.

However, by comparing the original and translated maps, it can be said that the biological coherence of both miRNAs and genes was preserved. In fact, in one class, the up-regulated genes/miRNAs relative to the other one have remained, as well as those that are down-regulated.

I carried out literature research to confirm the biological consistency maintained by the translated data.

- ▷ J.Wu et al.[3] showed how the CD39 alias ENTPD1 gene with a high expression value is a powerful prognostic marker of ccRCC patients.
- ▷ Hamamoto et al.[5] reported in the paper that the miRNAs hsa-miR-375, hsa-miR-205, and hsa-miR-196b are valid molecular markers for the classification of subtypes of non-small cell lung cancer (NSCLC).
- ▷ N.Okabe et al. [37] introduced the FAM83B as a new biomarker for the diagnosis and prognosis for lung squamous cell carcinoma.
- ▷ D.Petillo et al. [38] documented the overexpression of hsa-miR-424 in clear cell RCC relative to papillary RCC

Chapter 7

Conclusions

In the conclusion section, I will review the purpose and why of this method. I will show what is in the literature.
I will also discuss the limitations of my method, its strengths, and what improvements can be adopted. Talking about what I got, I would start from the beginning, i. e. why this method was developed.

I wanted to create a deep neural network model capable of translating two different transcriptomics data. In particular, I aimed at predicting gene expression data consistent with tissue type given miRNA expression data and vice versa. The proposed method shows that a supervised Adversarial Autoencoder is capable of performing this task.

The proposed method allows to obtain faithfully translated data. It requires a significant amount of data, and the data required for a faithful translation must be very similar to the training data. If sAAEs are used this way, the translations will report high biological consistency for high expression value data (especially for translated miRNAs) and statistical correlation greater than 75% for over 70% of differentially expressed genes/miRNAs.
If applied to several tumor tissues, the performance could decrease since the network is sensitive to expression levels. If a gene/miRNA typically has a lower expression value than it does in another tissue, it could be translated in a worse way than if they were translated separately. This factor could further limit the method because there would be a data limitation.

The proposed method is also limited at the implementation level. The translation between the two data types is purely deterministic, limiting the network to translations between only two data types at once i.e., miRNAs-to-genes or vice versa. Furthermore, this consequentially causes the fact that an sAAE can perform one-way translations only. The networks also require labeled data, and in order to train them, I need data for which the same patients are available. This approach severely limits the translation.

Several methods have been proposed in the literature for domain-to-domain or even intra-domain translation like "Cycle GAN" [39], "Disco GAN" [40] and the "Cross-modal Autoencoders" [21] methods.

- The CycleGAN method approaches the proposed method using the adversarial loss for the mapping function between two domains. It also

uses two cost functions defined as cycle consistency loss, by which it minimizes the distance of reconstructed samples.

- ▷ The DiscoGAN method exploits two GANs, i.e., two generators and two discriminators that train simultaneously. The training takes place on two strands. This method has been used to learn domain relations of images. Consider the two domains, A and B. In both strands, a domain translation takes place via a generator ($A \rightarrow B$ and $B \rightarrow A$). The obtained translation will be completed with a discriminator. It will have to understand which is the translated image and which is the real one. The real image will be the input of the opposite strand. Then the translation will be re-translated. This new image is the reconstruction of the starting image. Another cost function is introduced to minimize the distance between the new image with the starting image.

- ▷ The cross-modal autoencoders method exploits adversarial autoencoders to perform domain translation, crossing them to translate both images and genomic data as well genomic data of different types. The application is based on chromatin biology and the data used are ATAC-seq and RNA-seq from primary human immune cells, i.e., CD4+ lymphocytes.

The methods previously mentioned are structured probabilistic models [11]. The complexity introduced by these models is high. These models allow for greater generalization than deterministic models such as the one proposed. However, in this way, I was able to obtain accurate results without the need for probabilistic formalisms.
To make the translation more generalizable, I need to reintroduce the probabilistic assumptions proposed by [21] and a more significant amount of data.

Bibliography

[1] Lin Liu et al. "Comparison of next-generation sequencing systems". In: *J Biomed Biotechnol* 2012.251364 (2012), p. 251364.

[2] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. "Deep learning in bioinformatics". In: *Briefings in Bioinformatics* 18.5 (July 2016), pp. 851–869. ISSN: 1467-5463. DOI: 10.1093/bib/bbw068. eprint: https://academic.oup.com/bib/article-pdf/18/5/851/25581102/bbw068.pdf. URL: https://doi.org/10.1093/bib/bbw068.

[3] Jie Wu et al. "pHigh Expression of CD39 is Associated with Poor Prognosis and Immune Infiltrates in Clear Cell Renal Cell Carcinoma/p". In: *OncoTargets and Therapy* Volume 13 (Oct. 2020), pp. 10453–10464. DOI: 10.2147/ott.s272553. URL: https://doi.org/10.2147/ott.s272553.

[4] Youssef M Youssef et al. "Accurate molecular classification of kidney cancer subtypes using microRNA signature". In: *European urology* 59.5 (2011), pp. 721–730.

[5] Junko Hamamoto et al. "Identification of microRNAs differentially expressed between lung squamous cell carcinoma and lung adenocarcinoma". In: *Molecular medicine reports* 8.2 (2013), pp. 456–462.

[6] Colles Price and Jianjun Chen. "MicroRNAs in cancer biology and therapy: Current status and perspectives". In: *Genes & Diseases* 1.1 (Sept. 2014), pp. 53–63. DOI: 10.1016/j.gendis.2014.06.004. URL: https://doi.org/10.1016/j.gendis.2014.06.004.

[7] Zhong Wang, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature Reviews Genetics* 10.1 (Jan. 2009), pp. 57–63. DOI: 10.1038/nrg2484. URL: https://doi.org/10.1038/nrg2484.

[8] Fatih Ozsolak et al. "Direct RNA sequencing". In: *Nature* 461.7265 (Sept. 2009), pp. 814–818. DOI: 10.1038/nature08390. URL: https://doi.org/10.1038/nature08390.

[9] Felix Richter. "A broad introduction to RNA-Seq". In: *WikiJournal of Science* 4.1 (2021), p. 4. DOI: 10.15347/wjs/2021.004. URL: https://doi.org/10.15347/wjs/2021.004.

[10] Michael A. Nielsen. *Neural Networks and Deep Learning*. misc. 2018. URL: http://neuralnetworksanddeeplearning.com/.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[12] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2 (2012).

[13] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[14] Dor Bank, Noam Koenigstein, and Raja Giryes. *Autoencoders*. 2021. arXiv: 2003.05991 [cs.LG].

[15] Kien Mai Ngoc and Myunggwon Hwang. "Finding the Best k for the Dimension of the Latent Space in Autoencoders". eng. In: *Computational Collective Intelligence*. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 453–464. ISBN: 9783030630065.

[16] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[17] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[18] Alireza Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[19] Tracy Hampton. "Cancer Genome Atlas". In: *JAMA* 296.16 (Oct. 2006), pp. 1958–1958. ISSN: 0098-7484. DOI: 10.1001/jama.296.16.1958-d. eprint: https://jamanetwork.com/journals/jama/articlepdf/203742/jha60008-4-1.pdf. URL: https://doi.org/10.1001/jama.296.16.1958-d.

[20] Michael I Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome Biology* 15.12 (Dec. 2014). DOI: 10.1186/s13059-014-0550-8. URL: https://doi.org/10.1186/s13059-014-0550-8.

[21] Karren Dai Yang et al. "Multi-domain translation between single-cell imaging and sequencing data using autoencoders". In: *Nature Communications* 12.1 (2021), p. 31. ISSN: 2041-1723. DOI: 10.1038/s41467-020-20249-2. URL: https://doi.org/10.1038/s41467-020-20249-2.

[22] James Bergstra et al. "Algorithms for hyper-parameter optimization". In: *25th annual conference on neural information processing systems (NIPS 2011)*. Vol. 24. Neural Information Processing Systems Foundation. 2011.

[23] Tanay Agrawal. "Optuna and AutoML". In: *Hyperparameter Optimization in Machine Learning*. Springer, 2021, pp. 109–129.

[24] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631.

[25] Lisha Li et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6765–6816.

[26] Hervé Abdi and Lynne J. Williams. "Principal component analysis". In: *WIREs Computational Statistics* 2.4 (2010), pp. 433–459. DOI: https://doi.org/10.1002/wics.101. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wics.101. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.101.

[27] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[28] Andrew Ng, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems* 14 (2001), pp. 849–856.

[29] Stephen C Johnson. "Hierarchical clustering schemes". In: *Psychometrika* 32.3 (1967), pp. 241–254.

[30] David Sculley. "Web-scale k-means clustering". In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 1177–1178.

[31] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[32] José E Chacón. "A close-up comparison of the misclassification error distance and the adjusted Rand index for external clustering evaluation". In: *British Journal of Mathematical and Statistical Psychology* 74.2 (2021), pp. 203–231.

[33] Jorge M Santos and Mark Embrechts. "On the use of the adjusted rand index as a metric for evaluating supervised classification". In: *International conference on artificial neural networks*. Springer. 2009, pp. 175–184.

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

[35] T. Caliński and J Harabasz. "A dendrite method for cluster analysis". In: *Communications in Statistics* 3.1 (1974), pp. 1–27. DOI: 10.1080/03610927408827101. eprint: https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101. URL: https://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

[36] Andrew Rosenberg and Julia Hirschberg. "V-measure: A conditional entropy-based external cluster evaluation measure". In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 2007, pp. 410–420.

[37] Naoyuki Okabe et al. "FAM83B is a novel biomarker for diagnosis and prognosis of lung squamous cell carcinoma". In: *Int J Oncol* 46.3 (2015), pp. 999–1006. DOI: 10.3892/ijo.2015.2817. URL: https://doi.org/10.3892/ijo.2015.2817.

[38] David Petillo et al. "MicroRNA profiling of human kidney cancer subtypes". In: *Int J Oncol* 35.1 (2009), pp. 109–114. DOI: 10.3892/ijo_00000318. URL: https://doi.org/10.3892/ijo_00000318.

[39] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

[40] Taeksoo Kim et al. "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks". In: *CoRR* abs/1703.05192 (2017). arXiv: 1703.05192. URL: http://arxiv.org/abs/1703.05192.