POLITECNICO DI TORINO

Master degree in Biomedical Engineering

Curriculum in E-Health

Final Thesis

Design of a CNN-based method for classifying subtype of kidney cancers using miRNA isoform profiles

Supervisors
PhD. Gianvito Urgese
Prof. Elisa Ficarra
Eng. Marta Lovino

Candidate
Marco De Franchis

March 2020

Contents

Summary

The aim of this thesis is to exploit microRNA isoforms expression profiles and Artificial Intelligence (AI) tools to classify samples from different cancer studies. MicroRNA (miRNA) are small non-coding RNA molecules of 19-22 nucleotides that regulate gene expression via base-pairing with complementary sequences within mRNA molecules. Each miRNA sequence can occur with some modifications that may influence the final behavior of the molecule, this sequence is called isoform. Thanks to the evolution of sequencing technologies, an increasing number of miRNA expression data were released. The Cancer Genome Atlas (TCGA) is one of the projects that collect these kinds of data. Studies carried out on tumor and healthy samples showed differential expression of miRNA between the two categories, in particular for those miRNA families related to oncogenic or tumor suppressors gene pathways. The growing availability of such data together with the current AI tools allows us to design more powerful classification tools for tumor identification.

From this point, I decided to use miRNA isoform expression profiles as the input of Convolutional Neural Networks to predict malignancy in biological samples. With this aim I selected those cancer studies on TCGA with the highest amount of normal samples with respect to the tumoral available, that are: Kidney renal papillary cell carcinoma (KIRP), Kidney Renal Clear Cell Carcinoma (KIRC) and Kidney Chromophobe (KICH). The samples' numerosity varies among the subtypes and an imbalance between tumor and healthy samples up to a magnitude order is also present. To obtain their miRNA isoform expression profiles I considered separately two alignment tools from which I created two datasets: starting from the original TCGA alignment tool I created a table for each sample reporting its identified miRNAs in the rows and the expressions of 4 detectable isoforms in columns. From the alignment tool isormiR-SEA, which identifies a greater number of isoforms I also created a table for each sample with miRNAs in rows and up to 10 detectable isoforms in columns. Finally, for each table in the two datasets,

a column reporting the total expression for each miRNA was added. In the second part, I developed a system that, taken as input these two datasets, classifies samples from the same tissue into one of the four classes, namely the three cancer types and the healthy samples. The system compares the two datasets (which represent a different level of miRNA expression) and measures their effectiveness in classification tasks. I divided the samples of the three cancer studies in a training set, to train the classifier, and a test set to compute the performances together with cross-validation. Different configurations of the input data (isoforms and miRNAs) and classifiers (multiclass and binary, tumor subtype vs. tumor subtype and normal vs. tumor subtype) were tested. The overall results reported for each classifier test accuracies greater than 90% in both binary and multiclass approach. Nonetheless better performances were reported for the binary approach, in particular in distinguishing tumoral and normal samples (test accuracy greater than 98%). In almost all the tests, using as input datasets from IsomiR-SEA, slightly outperformed performances using TCGA dataset as input.

Chapter 1

Introduction

Thanks to the development of new technologies applied in medicine and biology, malignancy detection is nowadays conducted considering an increasing number of factors that help in obtaining more precise results. These factors comprehend data from the human genome, which is in its composition an enormous source of information since it contains the instruction for every function in the cell. Bioinformatics is a discipline that studies this type of data. In general, bioinformatics deals with techniques, algorithms, and tools for analyzing biological data, officially defined in 1970 as a "study of informatic processes in biotic systems" [12][14][13]. Today it combines methods from biology, computer science, and statistics to understand the biological processes behind the cell functionalities including aberrations that lead to developing cancer. Bioinformatic studies for cancer detection analyze different types of data deriving from the human genome. Among them, microRNA molecule is gaining more and more interest in literature as biomarker for cancer due to its regulatory role in gene expression [10]. In this thesis, I exploit data deriving from microRNA alignment pipelines in order to develop an innovative method for characterizing samples into its microRNA isoforms profile and use them as the input of a Convolutional Neural Network for any classification purpose. The purpose chosen for this thesis is classification of samples derived from specific cancer studies of kidney tissue. This particular application has to be intended as an use case of the system developed, since the sample characterization by means of miRNA isoforms can be exploited for different different classification tasks. Chapter 2 reports an introduction to the biological background of microRNA in its biosynthesis, functions, and isoforms. While, in Chapter 3, Deep Learning tools are introduced focusing on Convolutional Neural Networks. Chapter 4 introduces the data that will

be utilized for the analysis explaining the difference between the two sources in producing expression data from microRNA. In Chapter 5 the complete method for characterizing samples into its microRNA isoforms profile from the two data sources is reported together with the design of the Convolutional Neural Networks and dataset for the specific use case chosen, that is predicting malignancy in 3 classes of kidney tumors. Then, Chapter 6 reports graphical representations of the samples characterizations together with the results of the classification tasks together with a final discussion of the results for the specific use case. Finally, Chapter 7 reports some final considerations regarding the proposed method.

Computational resources were provided by HPC@POLITO, a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://www.hpc.polito.it)

Chapter 2

Biological Background

In this chapter, an overview of the biological background behind the miRNA is given. After a briefly introduction to the cell protein biosynthesis, where also miRNA is involved as regulator, a description to its biogenesis and isoforms is reported.

2.1 Protein biosynthesis in human cell

During its life cycle, the human cell carries out many complex mechanisms to exploit its functionalities, one of the most important is the synthesis of protein. This process can be summarized in two main steps: Transcription and Translation.

Transcription starts in the cell nucleus with the "unzipping" of the double-strand DNA molecule in a specific point by the enzyme helicase. The template DNA is than copied by means of RNA polymerase that reads from the 3-prime (3') end to the 5-prime (5') end so that a strand of messenger RNA (mRNA) in the 5'-to-3' direction is synthesized. In eukaryotic cells the mRNA undergoes through some post-transcriptional modification before reaching the cytoplasm and starts the Translation, one of these modifications is the splicing of introns that are the noncoding part of the gene.

During the Translation, mRNA synthesized from the nucleus is used as template and decoded by ribosomes that translates a specific sequence to a polypeptide according to the trinucleotide genetic code rules. The sequence of amino acids from this step forms the final protein.

The mRNA can undergo through different modification in order to regulate the protein production itself. Interactions with miRNA can cause such modifications.

2.2 Biogenesis and function of miRNA

In both plants and animals miRNAs are a class of ~22 nucleotides(nt) long non-coding RNA that can post-transcriptionally regulate gene expression. As described in [5], biogenesis of canonical miRNAs can be summarised in the following steps:

1. Transcription of a miRNA gene by RNA polymerase II (Pol II) to synthesizes a much longer RNA called "pri-miRNA".
2. Self-folding back of pri-miRNA forming a hairpin structure that will be the substrate for the Drosha-DGCR8 complex (called Multiprocessor in figure 2.1A).
3. Processing of the pri-miRNA by the Drosha-DGCR8 complex to release a ~60 nt steam-loop called "pre-miRNA"
4. Migration of the pre-miRNA to the cytoplasm through the action of Exportin 5 and RAN-GTP complex.
5. Cutting of the pre-miRNA near the loop by Dicer to generate the miRNA duplex, which contains the mature miRNA paired to its passenger strand (respectively red and blue strand in figure 2.1A).
6. Loading of the miRNA duplex into an Argonaute protein to form a silencing complex with the mature miRNA strand while degrading the passenger strand.

At its mature stage in the silencing complex, miRNA can be paired with an analogue sequence of the target RNAs to start the interaction that will lead to the silencing of the gene transcribed on the RNA target itself (blue; filled circle, cap; AAAAA, poly(A) tail in figure 2.1A).

If pairing is very extensive, the target can be sliced, whereas if it is not, the target can undergo other types of repression (as showed respectively at bottom-left and bottom-right in figure 2.1A) [5].

Canonical miRNAs can derive from both introns and exons of non-coding primary transcripts, some of which can codify hairpins for more than one miRNA. In addition, many canonical miRNAs derive from introns of pre-mRNAs (figure 2.1B) [5].

MiRNA can bind a specific target depending on the conservation of particular sub-sequences of nucleotide called miRNA-mRNA interactions sites. Among these sites, the seed sequence (red sequence in Figure 2.2) is the

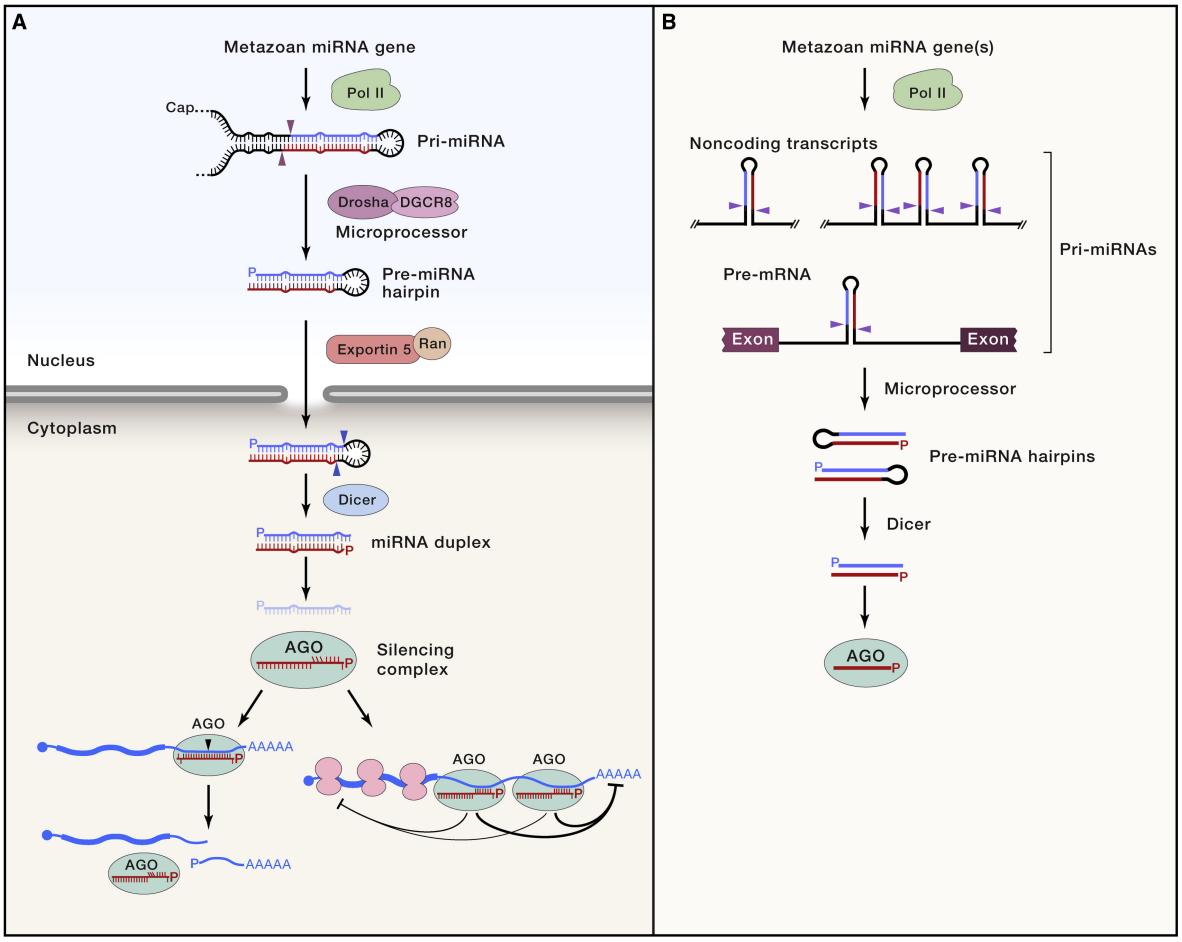


Figure 2.1. (A) Biogenesis and function of a typical miRNA. (B) Typical sources of canonical miRNAs. (source [5])

most important. Nonetheless,additional sites have been recently identified as able to guarantee high specificity in miRNA-mRNA interaction as showed in Figure 2.2 H.

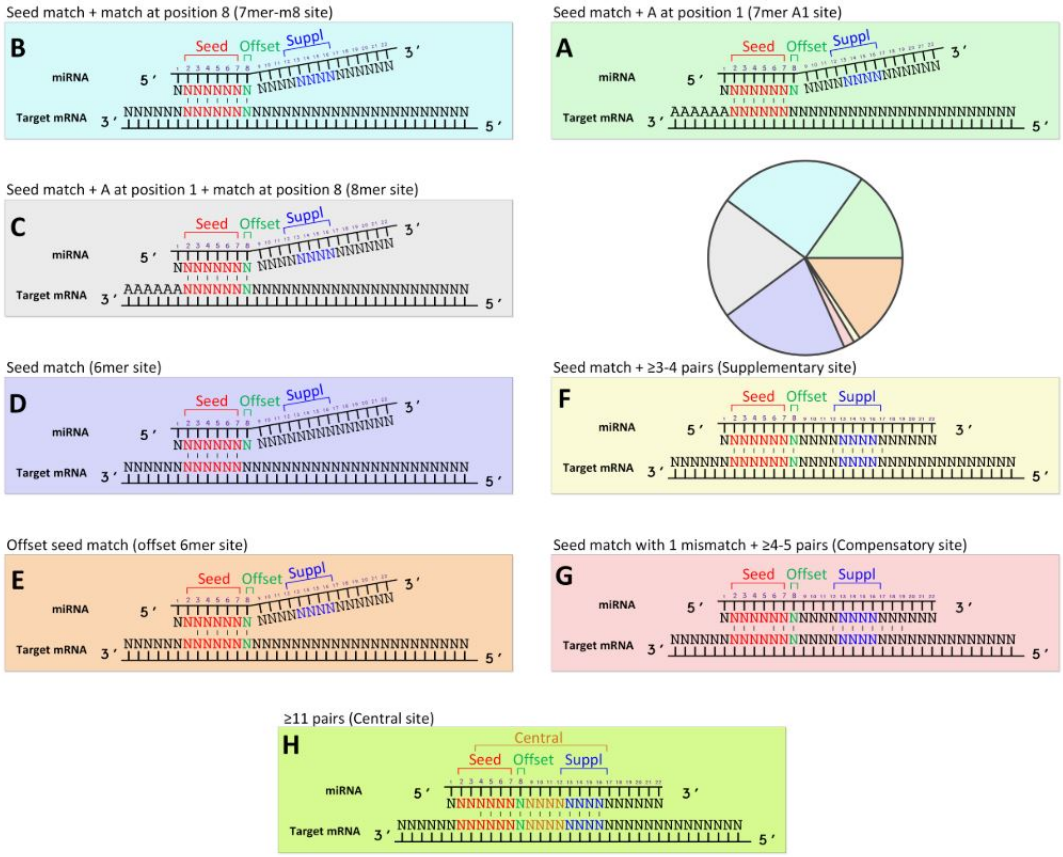


Figure 2.2. miRNA-mRNA Interaction sites schemes. The miRNA-mRNA interaction sites reported in H with red, green, blue and brown colors are known respectively as seed (nt 2-7), offset (nt 8), supplementary (13-16) and central (nt 3-16). In the pie chart is reported the percentage of observed miRNAs-mRNA interactions that use one of the seven binding mechanism shown in the boxes from A to G. (source [30])

2.3 miRNA isoforms (isomiRs)

Alterations during the biogenesis pathway of miRNA can generate multiple miRNA isoforms (isomiRs) from the same miRNA gene. As described in [30], processes like exoribonucleases, nucleotidyl transferase activity, RNA editing, and Single Nucleotide Polymorphisms (SNPs) from the miRNA loci are considered the main causes of such miRNA alterations.

Accordingly to [24], isomiRs can be separated into three main classes: 3' isomiRs, 5' isomiRs and polymorphic (SNP) isomiRs (reported in Figure 2.3). These isomiRs may have a huge impact on the capability of the miRNAs

to regulate gene expression. They can both lead the miRNAs to lose their capability to downregulate the mRNA, or even to make them acting on a complete different set of target mRNAs [30].

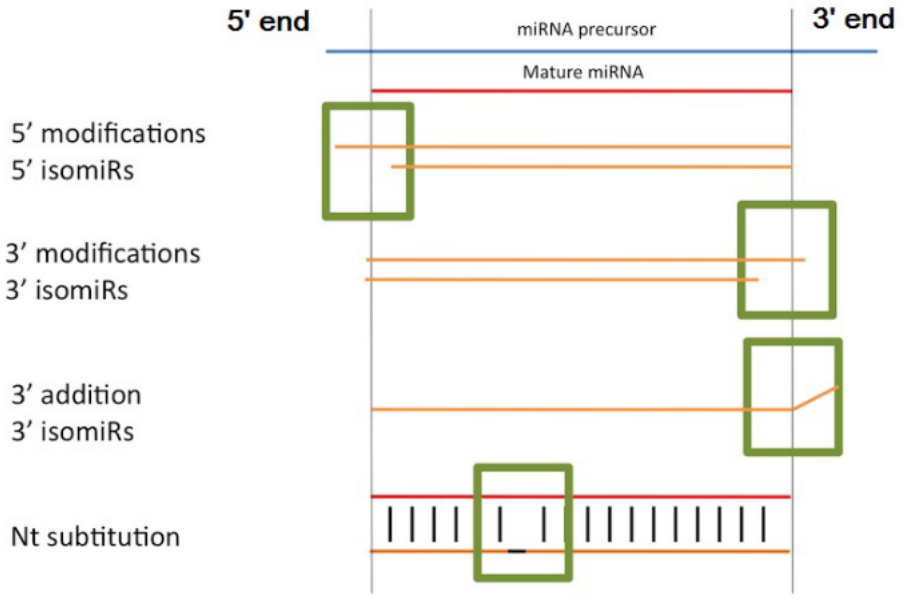


Figure 2.3. The three main isomiR types. In first line the 5' dicing site is upstream or downstream from the reference miRNA sequence. In second line the 3' dicing site is upstream or downstream from the reference miRNA sequence. In third line nucleotides are added to the 3' end of the reference miRNA. In the fourth line an SNP where nucleotides changes from the miRNA precursor.(source [30])

2.4 miRNA in cancer

Due to its regulatory role in gene expression, it has become clear over the last decades that aberrations in miRNA expression can be present in human malignancies. Many studies compares miRNA expression profiles between tumoral and normal tissues showing that up or down regulations of some miRNAs with roles in oncogenic and tumor suppressor pathways are present. In tables 2.1 and 2.2 some example of such differences from the *let-7* miRNA family are reported. Table 2.1 shows differential regulations comparing tumoral and normal tissue while in table 2.2 the regulation is also linked to the related miRNA target involved in tumoral pathways, references to these

data can be found in [17].

miRNA	**up/down reg.**	**cancer**
let-7	up	colon cancer
	down	breast cancer
	down	prostate cancer
	down	hepatocellular cancer
	down	gastric tumor
	up	uterine leiomyoma
	up	pancreatic cancer
	up	hepatocelluar carcinoma
	down	lung cancer

Table 2.1. let-7 miRNAs up- or down-regulated in various tumors relative to normal tissues(source [17])

miRNA		**effect on cell growth**	**Note (cancer type, etc)**
let-7	+	inhibition of cell growth	lung cancer cell lines
let-7	+	inhibition of cell growth (G1 arrest)	A549 lung cancer line or HepG2 cell line
	-	enhanced cell growth	A549 lung cancer line
let-7c	+	inhibition of cell growth (G1 accumulation)	Hepa-1
let-7g	+	inhibition of cell growth	lung cancer cell lines
	-	enhanced cell growth	
let-7a-3	+	increased anchorage independent growth (soft agar assay)	549 lung cancer line
let-7	-	enhanced cytotoxicity (more apoptosis) by gemcitabine, 5-FU, camptothecin	cholangiocarcinoma cell lines

Table 2.2. Phenotypes of cells are described after ectopic expression (denoted as "+" in the second column) or inhibition ("-" in the second column) of a let-7 family miRNAs [17]

Chapter 3

Deep Learning

In this chapter tools used for the thesis purpose are introduced, starting with an introduction to Artificial Neural Networks and some metrics regarding the evaluation of the models. Eventually, Deep Learning algorithms are introduced focusing on Convolutional Neural Networks(CNN).

3.1 Introduction to Artificial Neural Networks

In the past decades Machine Learning (ML) has gained an incredible interest in computer science. The success of such topic relies on the fact that its algorithms are able to extract information from raw data in order to represent it into a model that will be used to infer things about other data not yet modeled.

3.1.1 Artificial Neural Networks

Artificial Neural networks are machine learning models that represent the basis of the CNN [27]. These networks are non-linear structures of statistical data organized as modeling tools used to simulate complex relationships between inputs and outputs that other analytical functions cannot represent. As in the mammalian brain, the fundamental components of a neural network are the neurons (a.k.a. nodes) and the connections between them. These connections can change over the time while training, like in its biological corresponding. In the figure 3.1 a simple structure of neural network is showed. An artificial neural network receives external signals on a layer of input nodes connected with numerous internal nodes, organized in several levels. Each node processes the received signals and transmits the result to

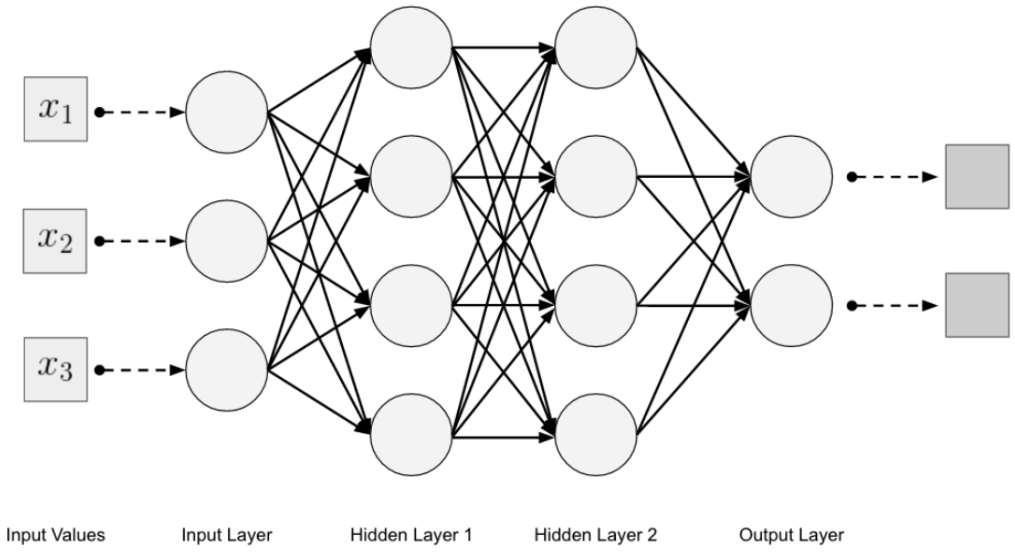


Figure 3.1. Multilayer neural network topology.(source [27])

subsequent nodes. This processing occurs in two phases: the input signal is multiplied by the weight of the connection and then all the results obtained are added up and sent to a specific function of the neuron that is called activation function. A representation of this process that shows an artificial neuron is reported in figure 3.2 The first model of artificial neuron is the so called Perceptron. This model, used for binary classification, has the same structure of the neuron shown in figure 3.2 with a step function as activation function. Later on, different activation functions were used for artificial neurons like the Sigmoid function:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{3.1}$$

Considered the n-dimensional input of the neuron as $x = (x_1, x_2, x_3..., x_n)$, $w = (w_1, w_2, w_3, ..., w_n)$ its weight vector and b its bias, the output of the neuron is computed as showed in function (3.2)

$$y = g(w \cdot x + b) \tag{3.2}$$

When the output goes only to the neurons input of the subsequent layer like in figure 3.1 we are talking about **Feed-Forward Neural Network**.

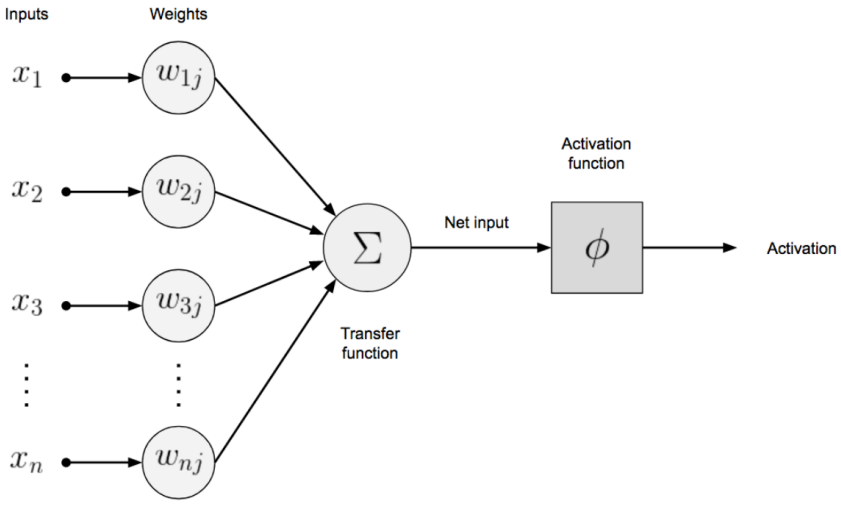


Figure 3.2. Details of an artificial neuron.(source [27])

3.1.2 Training of a Neural Network

The aim of the training (or learning) process is calculate weights and biases that amplify the input signal while reducing its noise to obtain the best prediction.
This process consists in a continuous readjustment of these parameters accordingly to the error in predicting a certain outcome.
The learning algorithm mostly associated with neural networks is the **backpropagation learning**.

Backpropagation Learning

Feed-Forward Neural Networks compute the output with a forward pass of the input through the network. If the output matches the label, nothing changes. If the output does not match the label, an adjustment of the weights is required.
The key of this adjustment in the Backpropagation learning is to backpropagate the error of the prediction and divide its contribution to each weight of the network.
The main steps are the following, considering **w** any weight of the network:

1. Propagation of the input signal up to the output layer where the prediction of the network is computed (forward-pass phase).

2. Computation of a loss function L that quantify the goodness of the prediction, its main argument is the difference between the current and the expected prediction.

3. Computation of the gradients of L for each parameter of the network with the chain rule back to the input layer (e.g. for a certain weight $\frac{\partial L}{\partial \mathbf{w}}$). This is how the contributions of every parameter to the prediction error is quantified (backward-pass phase).

4. Adjustment of the parameters by means of the computed gradients so that the loss function is minimized. For the weights of the network a learning-rate α can be considered in order to module the adjustment so that the new value of a certain weight will be:

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial L}{\partial \mathbf{w}} \tag{3.3}$$

5. Iterate until certain conditions are met.

The conditions for the iterations to stop can be different. It is usually set a certain number of *epochs* after that the training is stopped. The number of epochs is an hyperparameter that defines the number of times that the learning algorithm will work through the entire training set. Learning phase can be stopped also valuating the variation of certain performance metrics, avoiding the model to be trained up to the maximum number of epochs set. This technique is called **Early Stopping**.

3.2 Testing and optimization of a model

In order to assess the performance of our model we need to take into account different aspects such as the goodness of classification of previously unseen samples as well as its ability to generalize on a training dataset. The latter aspect takes into account not only how good, numerous and balanced our training samples are, but also whether the model itself has the right setting for the classification purpose. Searching for the right setting of the model is an optimization problem. From now on the set of samples used in the training phase and the previously unseen samples will be called respectively **Training set** and **Test set**. In this section some metrics and techniques to asses the hereinabove aspects are introduced.

3.2.1 Metrics

Confusion Matrix

One of the tools for evaluating models in classification tasks is the **confusion matrix** (figure 3.3).

	P' (Predicted)	N' (Predicted)
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

Figure 3.3. The confusion matrix in binary classification (source [27])

The confusion matrix is filled accordingly to the number of predicted labels with respect to the actual ones. In particular, in case of binary classification the following metrics are reported:

- True Positive (TP): positive predicted labels actually positive.
- False Positive (FP): positive predicted labels actually negative.
- True Negative (TN): negative predicted labels actually negative.
- False Negative (FN): negative predicted labels actually positive.

Different evaluations of the model can be computed from the confusion matrix counts. Here two measures that are generally used for assessing binary classifiers, i.e. **Accuracy** and **F1 score** are described.

Accuracy

Informally, it is the fraction of the correct predictions with respect to all the predictions of the model. That is, in binary classification:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

This metric alone is not sufficient to evaluate the model and in case of imbalanced data it can be also misleading. To clarify this concept an example

is reported. Let's consider a diagnostic test to evaluate the presence of a certain disease.
Taken 500 control samples (450 ill, 50 health) the confusion matrix reports the following counts:

$$TP = 445, \quad FP = 45, \quad TN = 5, \quad FN = 5 \tag{3.5}$$

The overall accuracy is 90%, which does not mean that the prediction of the test is correct nine time out of ten, because the number of True Negatives (samples correctly predicted as health) are only 10% of the total health samples tested. In these cases, other metrics should be considered.

F1-score

In binary classification, it is the harmonic mean of *precision* and *recall*. The first measures the proportion of the correct positive predictions, while the second the proportion of actual positive predictions. Mathematically:

$$Precision = \frac{TP}{TP + FP} \tag{3.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.7}$$

$$\textbf{F1 score} = \frac{2TP}{2TP + FP + FN} \tag{3.8}$$

K-folds Cross-Validation

It is a method used both to estimate the ability of the model to generalize on the training set and it is used in those cases where the training set has a limited number of samples. In cross-validation the training set is split into K number of splits (folds), then two groups for K training phases are created: K-1 splits belong to the training group while the remaining split belongs to the validation group to tests the model. On every training phase the splits rotate between the two groups and eventually all the possible variations are made. When validation group is composed by only one sample, this method is also called **Leave-one-out**. On every phase the model is tested and the metrics calculated so that mean and standard deviations of the metrics can be obtained to evaluate the aforementioned aspect.

3.3 Convolutional neural network

Convolutional neural network is one of the fundamental network architectures in deep learning. Compared to machine learning, artificial neural networks, deep learning neural networks (DLNN) have more complex architectures with a larger number of neurons, hidden layers and connections. Such complexity allow DLNN to analyze more complex data, such as images, for which DLNN where firstly conceived. Having multiple hidden layers means having different levels of representation of the input data that allows DLNN to automatically extract higher-order features.
CNNs try to learn these higher-order features in the data using convolution.

3.3.1 Architecture

CNNs try to transform input data from the input layer through all the network into a set of class scores. These scores represent how confidently the network assigned the input data to each class.
CNN architecture is composed of three main parts as showed in figure 3.4:

- Input layer
- Feature-extractions layers
- Classification Layers

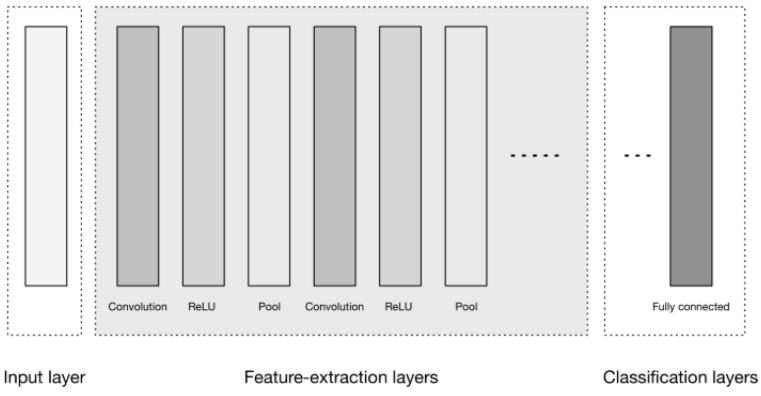


Figure 3.4. CNN High-level architecture (source [27])

The input layer can be N dimensional (in case of RGB image, the third dimension is given by each channel of the image). A fourth dimension can

be also considered if the samples are batched together.
Features-extraction layers are generally composed of repeated patterns of Convolutional Layers and Pooling layers (ReLU layers in figure 3.4 indicates layers of rectified linear unit activation function). Finally, there are the classification layers that can be composed of one or more fully connected layers to produce the class probability or score from the higher-order features previously extracted.

Convolutional Layers

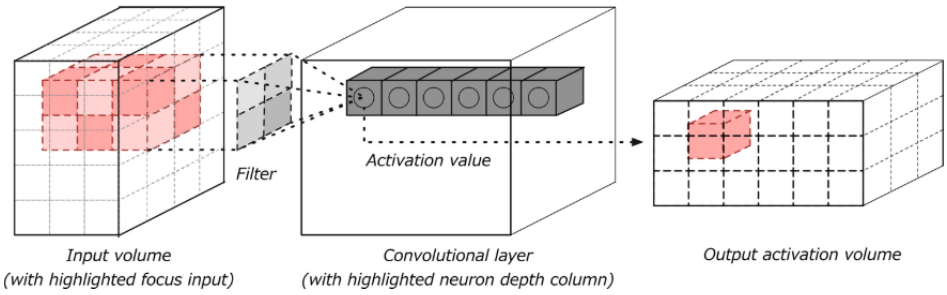


Figure 3.5. Convolution layer, from input to output (source [27])

Convolutional layers are considered the main blocks of the CNN architectures. The output volume is computed from the following steps:

1. Convolution of the whole input matrix through a sliding window (Kernel or Filter), obtaining a new matrix, usually with smaller dimensions.

2. The resulting matrix is *activated* by an activation function, forming an *activation map* or $featuremap$.

3. For all the filters in the Convolutional layer, a new feature map is computed following the first two steps.

4. All the computed feature maps are stacked together along the third dimension of the output volume.

Convolution

Convolution is a mathematical operation that merges two sets of information, in the case of CNNs they are the input data and a Kernel function. Considering a kernel $K[i, j]$ with dimensions [2I+1,2J+1] where $(i, j) = (0,0)$ is the

center of the kernel, the output g[m,n] applying the kernel function on the input a[m,n] is:

$$g[m,n] = K[i,j] * a[m,n] = \sum_{i=-I}^{+I} \sum_{j=-J}^{+J} K[i,j] \cdot a[m-i, n-j] \tag{3.9}$$

The kernel function is slid across the input data and multiplied by the input data values within its bounds, the result of each step is a single output entry as showed in figure 3.6

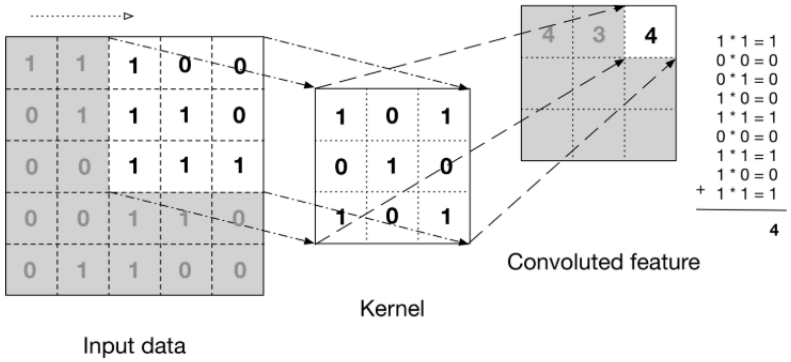


Figure 3.6. Convolution operation (source [27])

Convolution represents how the filter analyzes the input data in order to recognize some patterns. The moving window is composed of weights that are multiplied by the input values of the overlapping area to produce the feature map. Each element in the feature maps is *activated* by an activaction function, usually ReLU (indicated as layer in figure 3.4) so that, the feature map will only retain positive values, the rest are zeros. Eventually, the filters in the convolutional layer will be trained to locally recognize a specific pattern in the input data. Farther along in the network filters can recognize nonlinear combinations of features, detecting increasingly global patterns.

Pooling layers

Pooling layers can be seen as downsampling of the previous Convolutional output volume. They reduce progressively layer dimensions over the network, which helps control overfitting and reduces the number of parameters of the network. A typical operation of downsampling is *maxpooling*, where the output of the previous layer is divided into smaller windows from which only the max value is retained.

Classification layers

Classification layer come right after a flattening operation, where the matrices in the previous pooling layer are flattened so that all the elements lay on a single dimension. From this point, the architecture of the CNN has the same structure of an artificial neural network, with fully-connected layers up to the output layer. The output layer has a number of neurons equal to the number of classes to be recognized. These neurons usually report a score for each class obtained after all the forward steps of the input data up to the output layer so that they can be assigned to a certain class accordingly to the score.

3.3.2 Design

CNN design involves different types of parameters to be set. Network complex architectures could improve pattern recognition but increase the number of parameters to be trained and computational costs. Convolutional layer has different **hyperparameters**, to set the dimensionality of the output volume the following have to be considered:

- **Kernel (or filter) size**: Smaller size collects more local information but increases the number of observation areas in the input data, which is directly correlated to the number of neurons in the output volume.
- **Number of filters**: Different filters can recognize different patterns, increasing the number of filters increases the third dimension in the output volume.
- **Stride**: It's the sliding window step. The parameter controls the overlapping area of subsequent sliding windows. Smaller steps allocate more neurons in the output volume.
- **Zero-Padding**: Setting this parameter adds zero to the input outlines so that the output volume has the same spatial size of the input volume.

These parameters can be set accordingly to the classification task. Other procedures consider the tuning of one or more parameters in order to obtain the best performances from the model.

Chapter 4

Data

In this chapter, data sources utilized in this Thesis will be introduced. The type of miRNA isoforms expression data strongly depends on the alignment procedure. Two sources of data coming from two different alignment procedures have been considered:

- From The Cancer Genome Atlas
- From IsomiR-SEA tool

The two sources will be described in this chapter while the procedures to obtain the miRNA isoforms profiles will be discussed in chapter 5 applied to a specific use case, that is the class of samples derived from kidney tissue.

4.1 Introduction to miRNA alignment files

To produce the miRNA expression profiles of a particular sample, the miRNA must be sequenced through a sequencing machine. Sequencing machines are instruments that allow obtaining the ordered sequences of nucleotide basis (Adenine, Thymine, Cytosine, Guanine) of a particular DNA or RNA sample taken as input. In the case of RNA molecule, uracil is present in place of thymine. The machines produce a text file in the so called FASTA or FASTQ format, reporting the sequences of nucleotide bases (or reads) of the molecules that have been detected in the sample. In the FASTA format for nucleotide sequences, each read is reported as a string of letters preceded by a line that contains the identifier of that read, it usually starts with a grater than character '>'. The difference with FASTQ formats is that FASTQ reports in the next line of the read a sequence of characters for each nucleotide

base indicating the quality with whom the particular base was identified. Sequencing of short molecules like miRNA is usually done using a specific approach that demonstrated to be more accurate in sequencing molecules of up to 500 base pairs, the Next-Generation Sequencing.

4.1.1 Next Generation Sequencing

Next-generation sequencing (NGS) is a high-throughput approach to DNA and RNA sequencing that applies massively parallel processing to produce billions of nucleotide sequences. Different technologies using NGS have similar procedure steps:

1. **Sample Preparation**: The input for the sequencing machines are prepared, also called library preparation, though either amplification of the nucleotides sequence or ligation with custom adapters. Those adapters are fragments of nucleotides that enable hybridization to the sequencing chips in the machine and provide priming sites for the sequencing primers so that the process of sequencing can start.

2. **Sequencing:** Each pre-processed fragment is amplified creating clusters that act as individual sequencing reactions. The sequence of each fragment is optically read, depending on the technology, from repeated cycles of nucleotide incorporation. In Illumina system, for instance, the cycles are represented by repeated incorporation of fluorescent nucleotides detected by the camera and then removal of the fluorescent groups.

3. **Data Output:** The machine generate FASTQ format files, containing the sequences for each cluster.

Different technologies use NGS approach, the main differences lie in the sequencing technique, here some examples are reported:

- Pyrosequencing
- Sequencing by Synthesis
- Sequencing by Ligation
- Ion Semiconductor Sequencing

References and supplementary details can be found in [6].

The technology that uses NGS to reveal and quantify presence of miRNA in samples is called **microRNA sequencing (miRNA-seq)**.

4.1.2 Short-read alignment algorithms

Sequencing of miRNA molecules produces ~21 nucleotides reads that require a localization in the genome to be identified. For such short sequences, the localization in the genome can be tricky, not only because of the enormous difference of magnitude order (21 bases vs. 3 billion bases) but also because mutation, insertion or deletion in the sequence to be localized must be taken into account. Algorithms designed for localizing short reads in the genome are called short-read alignment algorithms. There are different types of short-read alignment algorithms in the market implementing different searching techniques. Here is reported a list of popular short read alignment software from [20]:

- Bfast
- Bowtie
- BWA
- Novoalign

The output of an alignment algorithm is a file reporting, for each read in the output sequencing file, where the read was aligned in the reference. The most common format is the Sequence Alignment Map (SAM), which reports together with the read position in the reference also other parameters that refer to the result of the alignment, its compressed binary version is the BAM format, reference and more details can be found in [21].

4.2 The Cancer Genome Atlas

The Cancer Genome Atlas (TCGA) is a project begun in 2005 from the National Cancer Institute and the National Human Genome Research Institute funded by the US government that aims to catalog and discover carcinogenic alterations in the genome [32]. Today TCGA has molecularly characterized over 20,000 primary cancer and matched normal samples spanning 33 cancer types using different techniques that include gene expression profiling, copy number variation profiling, SNP genotyping, genome wide DNA methylation profiling, miRNA profiling, and exon sequencing.
Data from TCGA can be retrieved in the Genomic Data Portal.

4.2.1 Data for miRNA analysis

Genomic Data Portal comprehends three types of data deriving from miRNA analysis:

1. **Aligned Reads**: Usually BAM file originated from the alignment with BWA-MEM algorithm [19].

2. **miRNA Expression Quantification**: A text file with a tabular structure reporting expression of any region in the precursor miRNA grouped together. In particular it reports read count, reads per million (RPM, that is a normalization done dividing by the sum of all the reads and multiply by one million) and a label indicating whether the pre-miRNA was cross-mapped (i.e. when a read exactly match two different sequence without recognizing the difference, the read count is assigned to both sequences and pre-miRNA is labeled as cross-mapped) [9].

3. **Isoform Expression Quantification**: Compared to the miRNA Expression Quantification, reads are now referred to the region inside the pre-miRNA, along with their coordinates (example in 4.1).

The pre-miRNA regions were annotated with different labels: "mature" (i.e. the mature strand), "star strand" (i.e. the passenger strand), "precursor" (i.e. precursor miRNA), "stem loop" (i.e. from 1 to 6 bases outside the mature strand, between the mature and star strands). The annotation label can be "unannotated", that is when the read was aligned to any region other than the mature strand in miRNAs where there is no star strand annotated [1]. These annotation refers to the **miRBase** database [2]. miRBase is a database that collects miRNA sequences from different organisms discovered in RNA deep sequencing analysis [16]. Each entry in the sequences database is a portion of a miRNA transcript (termed mir in the database), with information on the location and sequence of the mature miRNA sequence (termed miR). Due to the increasing quantity of data of miRNA-seq from different studies, ensuring the quality of miRNA annotations collected by miRBase has become challenging. Therefore, miRBase aims to provide post analyses of published microRNA sequences, including feedback from the users navigating in the miRBase database entries, asking for a confidence level. Each mature sequence in the database has its "Accession" that is the identification string starting with "MIMAT" also reported in the Isoform Expression Quantification file.

miRNA ID	Isoform coordinates	read count	RPM	cross-mapped	miRNA region
hsa-let-7a-1	hg38:chr9:94175940-94175962:+	1	0.412032	N	precursor
hsa-let-7a-1	hg38:chr9:94175960-94175982:+	2	0.824063	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175960-94175984:+	1	0.412032	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175960-94175985:+	1	0.412032	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175961-94175982:+	1	0.412032	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175961-94175983:+	1	0.412032	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175961-94175984:+	8	3.296253	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175961-94175985:+	1	0.412032	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175981:+	112	46.147545	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175982:+	2641	1088.175598	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175983:+	2202	907.293702	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175984:+	5234	2156.573677	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175985:+	155	63.864906	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175986:+	3	1.236095	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175981:+	1	0.412032	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175982:+	9	3.708285	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175983:+	8	3.296253	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175984:+	39	16.069235	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175985:+	1	0.412032	N	mature,MIMAT0000062
...	...	...	...	...	...

Table 4.1. Isoform Expression Quantification file

4.3 IsomiR-SEA

Tools for miRNA alignment have been sofar developed using different alignment techniques [36]. IsomiR-SEA (ISEA) is an alignment tool developed by G. Urgese et al. in 2016 [31] for miRNA-seq data. The name stands for isomiRNA Seed Extension Aligner related to the technique applied in the alignment where each read (also called tag) is aligned starting from the seed (nt. 2-7 figure 2.2) of miRNA mature sequences taken as input from a genome reference file. Alignment results of ISEA compared to other alignment tools report a more detailed analysis of the sequence taking into account possible in all the alignment phases, the positions of the encountered mismatches, thus allowing to distinguish among the different isomiRs and conserved miRNA-mRNA interaction sites [31]. The output of ISEA is composed of 3 files, a .log file a .tag and a .gff file. The log file refers to the parameters set for the alignment, the .tag and .gff file report the results of the alignment. A custom post processing procedure, developed for a M.Sc. Thesis, has been used to obtain the alignment results in a database file containing a table named Sample_MII_iso_inter with all the aligned tag grouped in the detected isoforms for each sample (figure 4.1). Each column has the following meaning:

- **MI**: an unique index value assigned to each miRNA sequence in the input reference file.

- **IEX**: a boolean indicating whether the isoform corresponds to the canonical mature sequence reported in the reference database.
- **I5P**: an integer indicating whether the isoform has insertion(+) or deletion(−) in the 5' end compared to the mature sequence.
- **I3P**: an integer indicating whether the isoform has insertion(+) or deletion(−) in the 3' end compared to the mature sequence.
- **IMS**: stands for *multiple nucleotide polymorphism isoform*, it is a boolean indicating if the isoform presents multiple mismatch w.r.t. the mature sequence.
- **ISN**: stands for *single nucleotide polymorphism isoform*, it is a boolean indicating if the isoform presents at least a mismatch w.r.t. the mature sequence.
- **INS**: a boolean indicating presence of mismatch in the seed.
- **IOS**: a boolean indicating whether the offset site (nt.8) is conserved in the isoform.
- **ISS** a boolean indicating whether the supplementary site (nt.13 to 16) is conserved in the isoform.
- **IPS**: a boolean indicating whether the compensatory site (nt 12 to 20 approximately) is conserved in the isoform.
- **ICS**: a boolean indicating whether the central site (nt 4 to 16 approximately) is conserved in the isoform.
- **MII**: an unique index assigned to each miRNA while reading the input reference files, MI refers to the absolute miRNA sequence while MII is specific of the organism.
- **IC5**: a boolean indicating whether the 5' end of the isoform is aligned to the 5' end of the miRNA mature sequence, it is set only if an insertion ore deletion if present, otherwise is '?'.
- **IC3**: a boolean indicating whether the 3' end of the isoform is aligned to the 3' end of the miRNA mature sequence, it is set only if an insertion ore deletion if present, otherwise is '?'.

- **sample_isea_db_id**: the id of the sample in which the isoform was identified.
- **iso5**: a sign indicating whether the 5' end of the isoform corresponds (=) to the 5' end of the mature sequence, or an insertion(+) or deletion(−) is present.
- **iso3**: a sign indicating whether the 3' end of the isoform corresponds (=) to the 3' end of the mature sequence, or an insertion(+) or deletion(−) is present.
- **iso**: a string collapsing IEX,I5P,iso5,IMS,ISN,iso3,I3P labels.
- **inter**: a string collapsing INS,IOS,ISS,IPS,ICS labels.
- **mmCount (Sum)**: a float indicating the sum of all the reads mapped with the same alignment score on different miRNAs.
- **TI (Count)**: an integer indicating how many unique tags were grouped in that isoform.
- **mism (mean)**: a float indicating the mean of mismatches for all the tag grouped in the isoform.
- **MI4S (Mean)**: a float indicating the mean of the number of miRNAs the grouped tags were aligned with.
- **MSD (Mean)**: a float indicating the mean of the differences in the scores between two subsequent alignments of miRNA with the same tag sequence.
- **TC (Sum)**: a float indicating the number of tags grouped within the isoform.
- **AL (Mean)**: a float indicating the mean of the length of alignment for every tag grouped within the isoform.
- **AM (Mean)**: a float indicating the mean of the alignment score for every tag grouped within the isoform.
- **countMultimap (Mean)**: a float indicating the mean of the number of reads that were mapped to multiple miRNAs associate to the current row.
- **SD (Mean)**: a float indicating the mean of the differences between the tags and the miRNA grouped with the isoform.

For ISEA analysis **MirGeneDB** reference genome file was taken as input. MirGeneDB is another database of validated and annotated miRNA. A study supporting mirGeneDB [11] shows that only approximately 16% of the 7,095 metazoan entries in miRBase are robustly supported as miRNA genes. While in mirGeneDB all the miRNA entries are supported by validated experimental procedures.

Table: Sample_MII_iso_inter

New Record | Delete Record

	iso	inter	MI	IEX	I5P	IMS	ISN	I3P	INS	IOS	ISS	IPS	ICS	MII	IC5	IC3	sample_isea_db_id	iso5	iso3	mmCount (Sum)	TI (Count)	mism (Mean)	MI4S (Mean)	MSD (Mean)	TC (Sum)	AL (Mean)	AM (Mean)	countMultimap (Mean)	SD (Mean)
	Filter	Fil...															Filter			Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	000fd591-93e5-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
2	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	02456117-dbe8-...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
3	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	044ae540-8d92-...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
4	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	05fd4af5-d627-4...	+	+	0.0	1	0.0	9.0	1.0	2.0	22.0	19.0	0.0	-2
5	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	091a9ffa-6646-4f...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
6	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	0cc147f5-0206-4...	+	+	0.0	2	0.0	9.0	1.0	3.0	22.0	19.0	0.0	-2
7	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	252e59d4-9f71-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
8	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	33f21d8b-05d5-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
9	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	4fc04f76-2262-4c...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
10	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	51c8805c-cb0d-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
11	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	539a2c65-7655-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
12	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	548177e1-5f4a-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
13	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	56d8a725-5f34-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
14	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	57af17c8-43ae-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
15	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	5c38c113-0461-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
16	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	635553f5-9b38-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
17	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	64b14e3a-e529-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
18	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	732440a9-09dc-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
19	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	76c337d1-2575-4...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2
20	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	7812eced-7063-4...	+	+	0.0	2	0.0	9.0	1.0	2.0	22.0	19.0	0.0	-2
21	FF+FF+F	FTTTT	0	F	1	F	F	1	F	T	T	T	T	3973	F	F	786845bb-943b-...	+	+	0.0	1	0.0	9.0	1.0	1.0	22.0	19.0	0.0	-2

Figure 4.1. Sample-MII-iso-inter Table

Chapter 5

Method

In this chapter, I describe the method that allowed to classify use case specific samples starting from their miRNA isoform profile exploiting CNN. Firstly, I describe how the input data for the classifier were designed and how I retrieve these data from the two sources (Isomir Expression quantification files from TCGA and Sample_MII_iso_inter tables from ISEA). Eventually, in the same section, I explain why I chose kidney cancer samples as use case in the whole panorama of cancer studies. Then I describe the classifier design itself, which architecture and hyperparameters have been chosen for the first tests and finally which tests have been planned to evaluate the problem in a binary or multiclass manner.

5.1 From miRNA alignment maps to miRNA expression matrices

The current miRNA isoform expression profiles allow us to retrieve different detailed information regarding the expression of the miRNAs. Due to the different types of alignment, the information ends up being organized differently so that a common way to represent the expression for each miRNA mature sequence divided into its isoforms is needed. The idea was to organize the information in tabular structures so that for each miRNA mature sequence in the row the expression of its isoforms detected in the sample is reported in columns (see table 5.1 for an example).

For each row, the first column contains an identifier to the miRNA mature sequence (that is set up differently depending on the miRNA reference

miRNA_ID	**iso_1**	**iso_2**	**...**	**exact**	**log2(RPM+1)**
miRNA_1	% of iso_1	% of iso_2	...	% of exact	total expression
miRNA_2	% of iso_1	% of iso_2	...	% of exact	total expression
...	...	...	...	...	...
miRNA_n	% of iso_1	% of iso_2	...	% of exact	total expression

Table 5.1. Schema of miRNA expression matrix

database) while from the second to the second to last column a float indicating the expression for each isoform in the sample is reported. This value is computed starting from the read counts identified in the sample for the reads that were attributed to that specific isoform from the aligner. The read count is divided by the total number of reads aligned to that miRNA. This normalization allows representing for each miRNA the percentage of expression of its isoforms detected in the sample. Finally, the last column reports a float indicating the total expression of that specific miRNA (exact sequence together with isoforms) detected in the sample. This value is computed starting from the sum of the total read count of all the isoforms identified in the sample for that specific miRNA and then divided by the total read count of all the isoforms of all the miRNAs identified in the sample so that the expression of each miRNA is normalized in the total sample. The value is then multiplied by one million (doing the so-called RPM normalization, usually done for count read normalization also in other studies [9]) so that the expression of a miRNA in a specific sample is comparable to one from another sample. This normalization can lead to values with discrepancies of several magnitude orders, for this reason, a supplementary adjustment is done, which is to sum by one the RPM value and compute the base two logarithms. This technique, also done in [9], allows not shadowing the contribution of miRNAs in the sample with a lower expression that is nonetheless significant.

5.1.1 From TCGA alignment

Starting from the Isoform Expression Quantification File described in section 4.2.1 I developed a pipeline in Python[26][23] that creates together with the reference genome coordinates of miRBase the miRNA expression matrix. The reference genome coordinates file of miRBase reports the genome coordinates along with the annotations of a particular organism for each miRNA gene sofar detected. The reference file for homo sapiens is called *hsa.gff3*

downloadable from the miRBase site [2]. The file has a tabular structure as shown in table 5.2, the columns report the following information (from left to right in 5.2):

- The Chromosome name.
- The sequence type, it can be miRNA_primary_transcript (i.e. hairpin precursor sequence) or miRNA (mature sequence).
- An integer indicating the start position in the chromosome.
- An integer indicating the start position in the chromosome.
- An integer indicating the end position in the chromosome.
- A sign between + or -, indicating whether it is located in the positive or negative strand of the DNA.
- The annotations, usually including the miRNA mature sequence ID in miRBase (starting with MIMAT) and the precursor ID (starting with MI) in case of miRNA mature sequence type, otherwise only the precursor ID is reported. Annotations include also a miRNA name, no more used as a reference because of issues [3].

chr1	miRNA_primary_transcript	17369	17436	-	ID=MI0022705;Alias=MI0022705;Name=hsa-mir-6859-1
chr1	miRNA	17409	17431	-	ID=MIMAT0027618;Alias=MIMAT0027618;Name=hsa-miR-6859-5p;Derives_from=MI0022705
chr1	miRNA	17369	17391	-	ID=MIMAT0027619;Alias=MIMAT0027619;Name=hsa-miR-6859-3p;Derives_from=MI0022705
chr1	miRNA_primary_transcript	30366	30503	+	ID=MI0006363;Alias=MI0006363;Name=hsa-mir-1302-2
chr1	miRNA	30438	30458	+	ID=MIMAT0005890;Alias=MIMAT0005890;Name=hsa-miR-1302;Derives_from=MI0006363
chr1	miRNA_primary_transcript	187891	187958	-	ID=MI0026420;Alias=MI0026420;Name=hsa-mir-6859-2
chr1	miRNA	187931	187953	-	ID=MIMAT0027618_1;Alias=MIMAT0027618;Name=hsa-miR-6859-5p;Derives_from=MI0026420
chr1	miRNA	187891	187913	-	ID=MIMAT0027619_1;Alias=MIMAT0027619;Name=hsa-miR-6859-3p;Derives_from=MI0026420
chr1	miRNA_primary_transcript	632615	632685	-	ID=MI0039740;Alias=MI0039740;Name=hsa-mir-12136
chr1	miRNA	632668	632685	-	ID=MIMAT0049032;Alias=MIMAT0049032;Name=hsa-miR-12136;Derives_from=MI0039740
chr1	miRNA_primary_transcript	1167104	1167198	+	ID=MI0000342;Alias=MI0000342;Name=hsa-mir-200b
chr1	miRNA	1167124	1167145	+	ID=MIMAT0004571;Alias=MIMAT0004571;Name=hsa-miR-200b-5p;Derives_from=MI0000342
chr1	miRNA	1167160	1167181	+	ID=MIMAT0000318;Alias=MIMAT0000318;Name=hsa-miR-200b-3p;Derives_from=MI0000342

Table 5.2. hsa.gff3 file from miRBase [2]

Since Isomir Expression Quantification file reports only the alignment coordinates of the reads in the genome, the detectable isoforms for each miRNA consider only insertion or deletion or exact match w.r.t. the miRNA mature sequence. The intermediate columns in the matrix are:

- **I5P**: reporting the reads with a maximal deviation of two positions w.r.t. the 5' end coordinates of the miRNA mature sequence in the genome coordinates file.

- **I3P**: reporting the reads with any deviation in the position w.r.t. the 3' end coordinates of the miRNA mature sequence in the genome coordinates file.
- **I5P-I3P**: reporting the reads that satisfied both the I5P and I3P conditions.
- **EX**: reporting the reads with an exact match in the coordinate positions of the miRNA mature sequence in the genome coordinates file.

An input/output schema of the Python script is reported in figure 5.1.

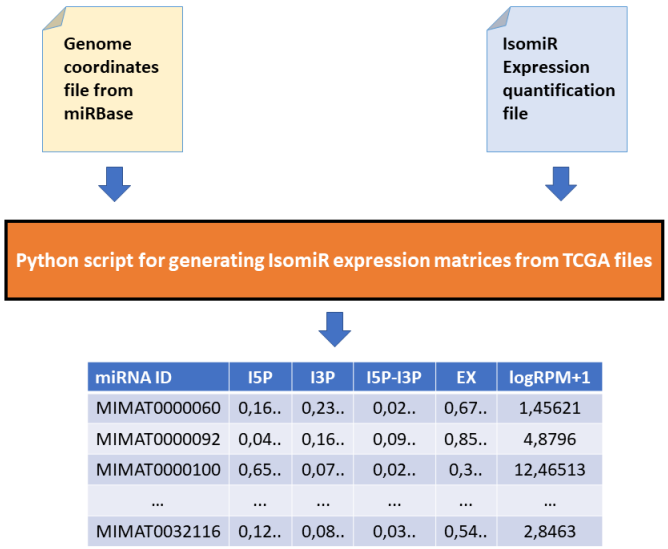


Figure 5.1. Input/output schema for the TCGA files pipeline

The **pipeline** to create an expression matrix follows these steps:

1. The genome coordinates file is loaded as a DataFrame (a variable type from the Pandas library [22] that replicates the tabular structure of the original file) that here I call *Coordinates-DF*.
2. A list of unique mature sequence identifiers (MIMATs) is extracted from the annotations column and sorted in ascending order.
3. A DataFrame is created, with the MIMATs as indexes (rows) and five columns, each for every isoform together with the total read count. It represents the expression matrix to be filled, that here I call *Expression-DF*.

4. The Isoform expression quantification file of a specific sample is loaded as DataFrame, that here I call *Isoform-DF*.

5. For each row in *Isoform-DF*, if the annotations indicate coordinates related to a MIMAT the script searches for the coordinates of that MIMAT in the *Coordinates-DF*.

6. The script compares the coordinates of the *Isoform-DF* row with the coordinates of the MIMAT, depending on the conditions reported in the column list (above) the read count of that specific *Isoform-DF* row is reported to the corresponding MIMAT row in *Expression-DF* for the corresponding isoform detected.

7. Steps 5 and 6 are iterated until the last row in *Isofrom-DF*.

8. The *Expression-DF* is normalized using the procedure reported in 5.1 and saved as numpy array (variable type from Numpy library [25]).

The final mirna expression matrix for a given sample is composed of 2652 rows (corresponding to the 2652 MIMATs present in miRBase) and 5 columns, one for each isoform including the total expression of the MIMAT (reported as **logRPM**).

5.1.2 From IsomiR-SEA alignment

IsomiR-SEA alignment and output post-processing provide the Sample_MII_iso_inter table. The table can group alignment information deriving from multiple samples, in addition, it reports a huge number of characteristics referring to all the interaction sites in the miRNA. This level of detail allows identifying a greater number of isoforms. In this work, to differently characterize the isoforms I considered the columns **IEX**, **I5P**, **IMS**, **ISN**, **I3P**, **INS** (ref. on section 4.3). I chose these columns because comparing to detectable mismatches in other sites (IOS, ISS, IPS, ICS), seed mismatches (INS) can have a stronger impact on the miRNA-mRNA interaction, being the seed their primary interaction site (ref. on section 2.2). While ISN and IMS represent mismatches in the whole sequence so that also mismatches in other sites can be included. Combinations of the values reported in the chosen columns allow separating the aligned reads into 16 different isoforms reported in table 5.3.

Due to its file format, miRNA expression matrix for each sample can be extracted from the Sample_MII_iso_inter table using specific SQL queries.

	Sample_MII_iso_inter columns					
Isoforms	**IEX**	**I5P**	**IMS**	**ISN**	**I3P**	**INS**
I5P-MS-I3P	F	!=0	F	T/F	!=0	T
I5P-MS	F	!=0	F	T/F	==0	T
MS-IM	F	==0	T	T/F	==0	T
I5P-MS-IM	F	!=0	T	T/F	==0	T
MS-IM-I3P	F	==0	T	T/F	!=0	T
I5P-MS-IM-I3P	F	!=0	T	T/F	!=0	T
MS-I3P	F	==0	F	T/F	!=0	T
MS	F	==0	F	T/F	==0	T
I5P-IM-I3P	F	!=0	T	T	!=0	F
	F	!=0	F	T	!=0	F
	F	!=0	T	F	!=0	F
IM-I3P	F	==0	T	T	!=0	F
	F	==0	F	T	!=0	F
	F	==0	T	F	!=0	F
IM	F	==0	T	T	==0	F
	F	==0	F	T	==0	F
	F	==0	T	F	==0	F
I5P-IM	F	!=0	T	T	==0	F
	F	!=0	F	T	==0	F
	F	!=0	T	F	==0	F
I5P-I3P	F	!=0	F	F	!=0	F
I5P	F	!=0	F	F	==0	F
I3P	F	==0	F	F	!=0	F
EX	T	==0	F	F	==0	F

Table 5.3. Isoforms extracted from Sample_MII_iso_inter

For this reason, I created a custom script in Python that, taken as input a file containing instruction to create the query, it extracts the miRNA expression matrix with the read counts (not normalized) from the Sample_MII_iso_inter table and reports it into a DataFrame structure. To query the Sample_MII_iso_inter table I used the SQLite3 library in Python. The instructions to create the query are retained in a csv file, reported in a tabular structure on table 5.4). Each row represents a particular isoform to be extracted using the condition reported in the second column, while the first column contains the name attributed to the isoform. The last row (TOT) doesn't refer to an isoform but allows collecting the total read count for each MII, that refers to a particular miRNA mature sequence identifier from the mirGeneDB database (ref. 4.3).

Different miRNAs with similar sequence are likely to have the same mRNA target (section 2.2). For this reason, i decided to group miRNAs with similar sequence so that the miRNA expression matrix for each sample ends up having in the rows expression of miRNA groups with similar sequence. This similarity was computed by isomiR-SEA taking as input the mature sequences of mirGeneDB and aligned with the mature sequences of mirGeneDB

Isoform name	Query conditions
I5P-MS-I3P	I5P!=0 AND I3P!=0 AND INS=="T" AND IMS=="F"
I5P-MS	I5P!=0 AND I3P==0 AND INS=="T" AND IMS=="F"
MS-I3P	I5P==0 AND I3P!=0 AND INS=="T" AND IMS=="F"
MS	I5P==0 AND I3P==0 AND INS=="T" AND IMS=="F"
MS-IM	I5P==0 AND I3P==0 AND INS=="T" AND IMS=="T"
I5P-MS-IM	I5P!=0 AND I3P==0 AND INS=="T" AND IMS=="T"
MS-IM-I3P	I5P==0 AND I3P!=0 AND INS=="T" AND IMS=="T"
I5P-MS-IM-I3P	I5P!=0 AND I3P!=0 AND INS=="T" AND IMS=="T"
I5P-IM-I3P	I5P!=0 AND I3P!=0 AND (IMS=="T" or ISN=="T") AND INS=="F"
IM-I3P	I5P==0 AND I3P!=0 AND (IMS=="T" or ISN=="T") AND INS=="F"
I5P-IM	I5P!=0 AND I3P==0 AND (IMS=="T" or ISN=="T") AND INS=="F"
IM	I5P==0 AND I3P==0 AND (IMS=="T" or ISN=="T") AND INS=="F"
I5P	I5P!=0 and I3P==0 AND IMS=="F" AND ISN=="F" AND INS=="F"
I3P	I5P==0 and I3P!=0 AND IMS=="F" AND ISN=="F" AND INS=="F"
EX	IEX=="T"
TOT	MII>=0

Table 5.4. Table from query.csv file

themselves. The miRNA groups were created considering an alignment score between each couple of miRNA greater than 15. From the 1171 distinct miRNA mature sequences in mirGeneDB I obtained 923 miRNA groups that correspond to the final number of rows in the expression matrix. The indexes to group each miRNA are retained in the GeneDB_TI _MIN so that a unique query can be created for each sample. The final schema of the input/output pipeline is reported in figure 5.2.

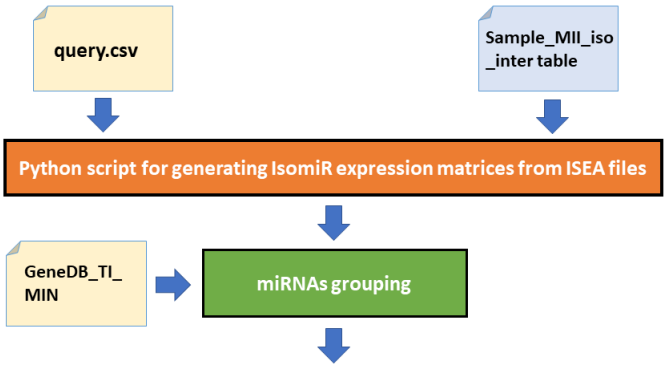
The steps of the pipeline are the following:

1. The algorithm extracts the unique list of samples from the Sample_MII_iso_file using a specific query.

2. For each sample in the list of samples, the query is constructed from the query.csv attaching the GeneDB_TI _MIN so that the miRNAs can be grouped.

3. The query is processed from Sample_MII_iso_and the expression matrix extracted (example of a query is reported in appendix A.1).

4. The query result is loaded into a DataFrame variable then normalized using the procedure reported in 5.1.Finally the matrix is saved as numpy array.



Group ID	I5P	I3P	I5P-I3P	...	MS	EX	logRPM+1
1	0,0835	0,233	0,0256	...	0,05	0,67	1,45621
2	0,0442	0,162	0,089	...	0,078	0,45	4,8796
...	...	...	...	...	...	...	...
891	0,0698	0,0545	0,0563	...	0,042	0,12	5,5543
892	0,1227	0,0851	0,0457	...	0,074	0,54	2,8463

Figure 5.2. Input/output schema for the ISEA files pipeline

The final miRNA groups expression matrix has 932 rows and 16 columns reporting the isoform expression and the total expression for each miRNA group (reported as **loagRPM**).

5.1.3 Use case

In the panorama of miRNA-seq available data from the Genomic Data Portal, only three primary sites report more than one thousand samples from miRNA-seq analysis. In figure 5.3 a stacked bar chart shows the distribution of tumoral and normal tissue in the three sites. For the classification

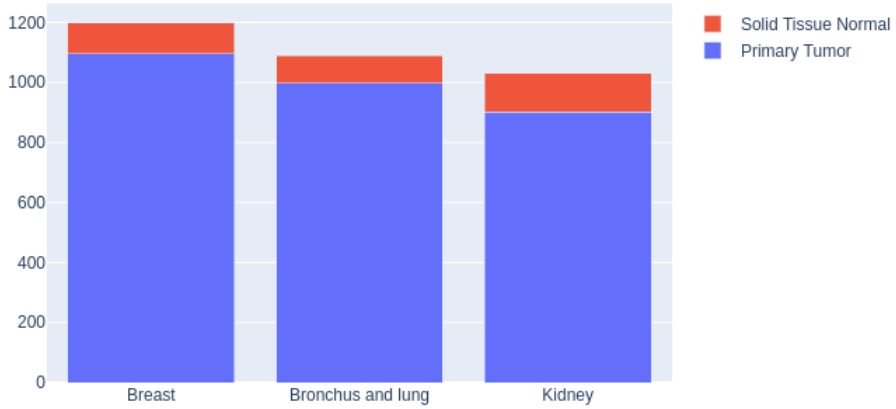


Figure 5.3. MiRNA-seq samples' distribution in Breast, Bronchus+lung and Kidney.(created with [15])

purpose, I looked for the highest normal/tumoral samples ratio for choosing the primary site, which lead me to the kidney. Kidney samples derive in turn from three different kidney cancer subtypes assigned to specific TCGA projects: Kidney renal papillary cell carcinoma (KIRP), Kidney Renal Clear Cell Carcinoma (KIRC) and Kidney Chromophobe (KICH). The samples' numerosity varies among the subtypes and an imbalance between tumor and healthy samples up to a magnitude order is also present (figure 5.4).

Chosen these three cancer subtypes, classification performances for both binary and multiclass condition will be evaluated (as described in 5.3). Since I considered tumors coming from the same tissue, I decided to put together normal samples, while the tumoral samples remain separated in the three classes (KIRP, KIRC, KICH). Consider samples coming from the same tissue, lead me to another consideration: since the miRNA expression mainly depends on the type of tissue, only a small percentage of the total miRNAs previously considered will be expressed in the specific kidney tissue and reducing the number of miRNAs (rows in the matrices) can considerably simplify the feature space for the classification tasks. Therefore a filter based on expression was designed. It can be applied (differently) in both databases

Figure 5.4. Samples' distribution of the kidney cancer subtypes (created with [15]).

(miRNA expression matrices from TCGA align. and miRNA groups expression matrices from ISEA align.) and select those rows that satisfied specific expression conditions in the whole tissue. To evaluate the expression of the miRNAs in the whole tissue I took all the samples (normal and tumoral) and calculate the matrices (expressing read count, not normalized) from the two alignment procedures (TCGA and ISEA) separately. Then I summed up the matrices, separately from the two alignment procedures, so that I obtained the whole expression in read counts for every miRNA/miRNA group in the last column. Then I normalized in RPM and obtained the miRNA/miRNA group normalized expression in the whole tissue using the two alignment procedures. I chose to filter those miRNAs/miRNA groups that in the whole tissue reported an expression value greater than 10 RPM and 20 RPM. This lead to a reduction in the number of rows of a magnitude order for both the databases (from TCGA align. and ISEA align.).

Due to the redundancy of information and a priori knowledge of the data I decided to group some isoforms of the miRNA groups expression matrices following the schema reported in table 5.5.

Table 5.6 report the final dimension of the datasets.

Isoform groups	Isoforms
I5P or MS or IM or I3P	I5P-MS-I3P
	I5P-MS
	MS-IM
	I5P-MS-IM
	MS-IM-I3P
	I5P-MS-IM-I3P
MS-I3P	MS-I3P
MS	MS
I5P-IM-I3P	I5P-IM-I3P
IM or I3P	IM-I3P
	IM
I5P-IM	I5P-IM
I5P-I3P	I5P-I3P
I5P	I5P
I3P	I3P
EX	EX

Table 5.5. New isoforms for miRNA groups expression matrices

Align. procedure	Initial	**RPM >10**	**RPM >20**
TCGA align.	2652 x 5	190 x 5	156 x 5
ISEA align.	932 x 16	175 x 11	141 x 11

Table 5.6. New dataset dimensions (rows x columns)

5.2 Classification tool design

To classify kidney samples exploiting the expression of the isoforms for each miRNA/miRNA group as they are represented in the datasets I used a CNN with the following architecture (as reported in figure 5.5):

1. A 2D convolutional layer.
2. A dense layer with 64 neurons.
3. An output layer.

The kernel dimension for the convolutional layer varies according to the input,

I chose to set a monodimensional kernel that covers all the columns (isoforms) taking a single row at a time(miRNA/miRNA group). It slides vertically the input computing in the output of the convolutional layer a single value that refers to a single row. I added no more convolutional layer as I wanted to keep separate the information for each miRNA/miRNA group taken from the isoforms. The output layer is composed by a number of neurons related to the classification task (binary or multiclass). Since I wanted to test different configurations (section 5.3) the final architecture of the input and output layer varies with every test. CNN were implemented and fit through the Keras library [8].

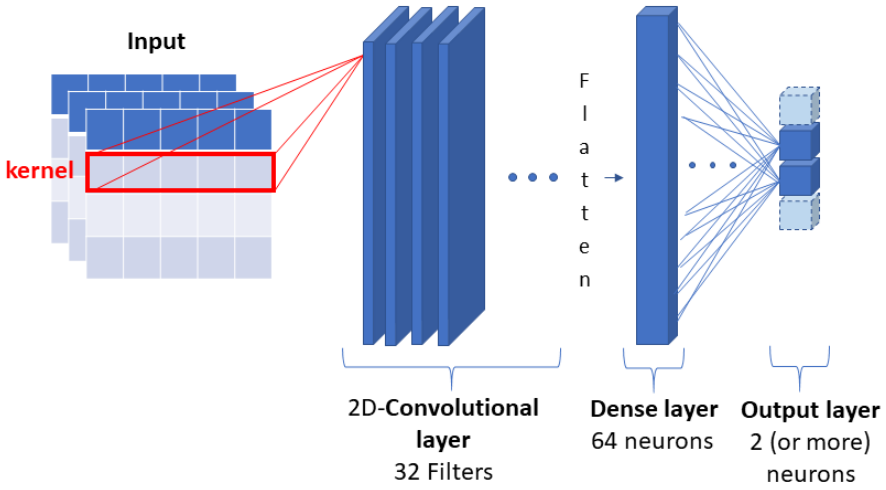


Figure 5.5. CNN architecture

5.2.1 Training and test set preparation

The overl dataset is composed of 1033 samples divided in 3 different TCGA project (cancer subtypes) including normal samples as showed in figure 5.4. Accordingly to common machine learning procedure, I split the whole dataset in training and test set, sampling respectively 929 (90%) and 104 (10%) entries in the whole datasets. Sampling the datasets I chosed each set to be proportionally composed by samples from all the chosed TCGA projects divided in tumoral and normal sample. In table 5.7 the final split is reported. Both training and test sets turn out to be unbalanced, since the limited

TCGA Project	Type	**Training set**	**Test set**
KIRC	Tumor	492	52
	Normal	30	7
KIRP	Tumor	259	32
	Normal	65	5
KICH	Tumor	59	7
	Normal	24	1

Table 5.7. Training and test split.

availability of both normal and tumoral samples in Genomic Data Common. During the training phase, unbalanced dataset can strongly compromise the performance of the classifier, for this reason I used an oversampling technique called SMOTE [7] to create artificial samples in the training set so that the majority and minority classes have always the same number of data. SMOTE creates artificial samples of the minority class by selecting for each sample a certain number of neighbors of the same class in the feature space, the samples are created randomly between the samples and the selected neighbors for all the neighbors, figure 5.6 reports a schema of the oversampling algorithm. This aspect can be critical in the case of outliers within minority classes since the algorithm doesn't recognize the outliers and simply creates artificial samples that can have the same characteristics as the latter. Since SMOTE technique can be applied only to multidimensional samples in an array-like structure, a flattening operation to the miRNA/miRNA groups expression matrices must be done. For this reason, I decided to flatten the matrices extracting each column and stack it in a monodimensional array, this procedure is repeated for all the columns. Figure 5.7 reports a schema of the procedure to flatten the matrices. The SMOTE technique was applied exploiting the library imbalanced-learn [18].

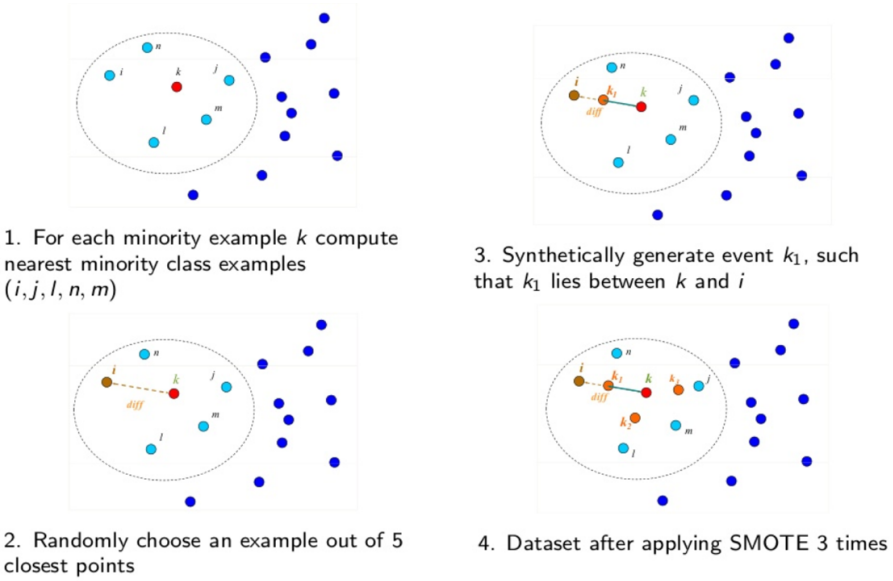


Figure 5.6. Summary scheme of SMOTE technique (source [4])

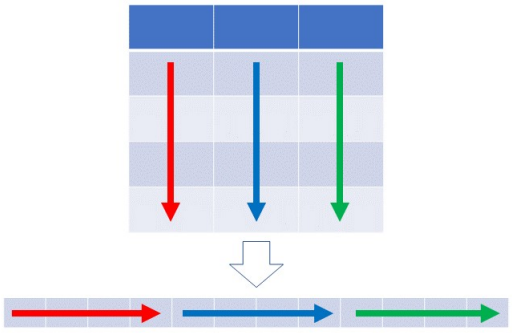


Figure 5.7. Flattening operation scheme

5.2.2 Hyperparameters settings

To configure the CNN, the 2D-convolutional layer must be set in its hyperparameters. As listed in section 3.3.2, the chosen hyperparameters are:

- **Kernel (or filter) size**: $1xN$ window, where N is the number of isoforms (columns) in the input matrix.
- **Number of filters**: 32.

- **Stride**: Kernel scans vertically the input matrix one row at a time with no overlap:
- **Zero-Padding**: No zero-padding set.

5.2.3 Training design

To train the CNN in both binary and multiclass classification I used the following settings:

- **Batch size**: 8 samples.
- **Number of epochs**: 100.
- **Loss function**: Categorical cross-entropy. This loss function can be applied only when each sample corresponds to a specific class. For this reason, the output layer must have a softmax activation function. The categorical cross-entropy has the following formula for a riven output y:

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * \log(\hat{y_{ij}})) \tag{5.1}$$

  Where $\hat{y}$ is the actual class, M refers to the number of output neurons that is also the number of classes, while N is the number of observations.
- **Optimizer**: Adadelta[34], an optimizer that dynamically regulates the learning rate of the gradient descent algorithm.

To optimize the training process I used the **Early Stopping** technique (ref. section 3.1.2) so that if the validation set accuracy doesn't reach a new maximum after 35 epochs, the training stops. Once the training is over, the best model over the whole epochs is saved (i.e. the model that reached the best validation set accuracy). This technique is called **Model Checkpoint**.

5.2.4 Model evaluation

To evaluate each model, since I am dealing with a limited number of samples in the training set, I utilize the **K-fold Cross-Validation** technique (ref. section 3.2) setting K equal to 10. At each fold i compute the following metrics of the best model on the validation set:

- Accuracy (acc)

- F1-score (f1)
- Loss
- Confusion matrix (CM)

At the end of Cross-Validation(CV), I compute the mean and standard deviation of the accuracy, F1-score and loss among all the folds. Finally, I train the model with the whole training set and compute the same metrics using the test set, this results are indicators of the model ability to classify unseen data (generalization).

5.3 Test plan

To test the capability of the two miRNA isoforms profiles (TCGA and ISEA) to classify cancer samples I considered two classification approaches:

1. **Binary**: I took all the possible combinations of the four classes to make different binary classifiers, which means distinguishing either two types of cancer subtypes (es. KIRP vs. KIRC) or normal tissue and cancer subtype specific tissue (es. KIRC vs. Normal).
2. **Multiclass**: I tested the ability of the CNN to classify a cancer sample into one of the three subtypes and then a classifier that distinguishes samples among all the four classes.

Since the two miRNA isoform profiles report different types of isoforms I decided to test whether the larger availability of isoforms from ISEA alignment procedure can improve the model performance compared to the smaller set of isoforms from TCGA alignment procedure. The two types of alignment were tested also comparing the same type of isoforms, to see whether one of the two alignment procedures reflects better performances in the classification tool from similar types of information. Together with the isoforms also the two levels of miRNA/miRNA groups expression were tested, that are RPM>10 and RPM>20 reported in table 5.6.

Chapter 6

Results and discussion

This chapter reports firstly two representations of the data generated from the pipelines introduced in section 5.1.1 and 5.1.2, then, a section dedicated to the results from the tests planned in section 5.3. Finally, a discussion of the results from this particular classification task in both approaches is reported in the last section.

6.1 Data representation

In this section, representations of the datasets resulting from the two different alignment procedures are reported. The data are represented by means of the two principal components from the principal component analysis (PCA) [33] in figures 6.1 and 6.2, each point represents a single sample from one of the four classes: **KIRC**, **KIRP**, **KICH** and **Normal**, each labeled with a different color. In both graphs KIRC and Normal samples have a higher scattering compared to the other classes, nonetheless, the graph from IsomiR-SEA alignment shows a relatively higher separability of the classes compared to the representation using TCGA alignment.
In Supplementary materials A.2 is reported an isoform-level representation of the datasets by means of horizontal boxplots considering the specific miRNA/miRNA groups expression filtering as described in table 5.6. Each boxplot in figuresA.2 represents the range of values of a specific isoform (in vertical axes) of a specific class (related to the color association in the legend). Comparing the two types of miRNA/miRNA groups in both alignments they show almost no difference in terms of isoform expression. Regarding the TCGA alignment all the classes report a higher expression of I3P isoform compared to the other isoforms. Isoforms in both alignments don't show a

sensible variability among the classes except for MS and MS-I3P isoforms present in both graphs of the ISEA alignment.

TCGA alignment

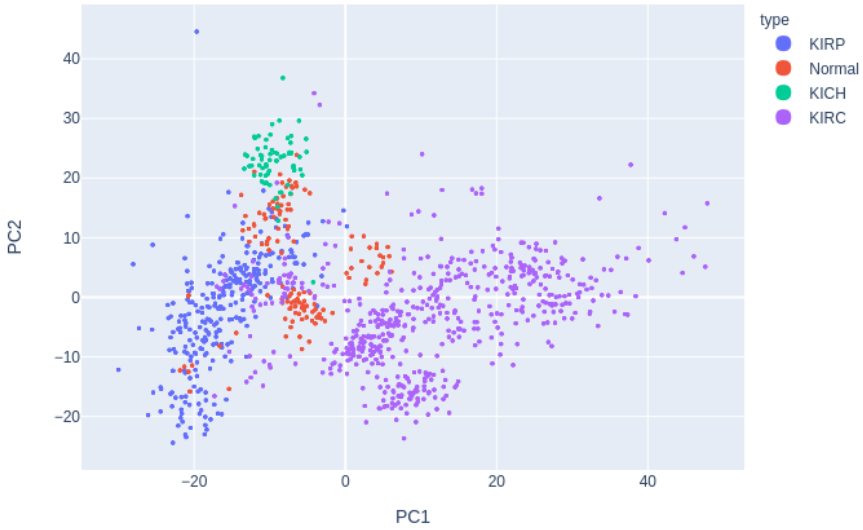


Figure 6.1. Samples from TCGA alignment represented using the first two components (PCA1 and PC2) from PCA (created with [15]).

IsomiR-SEA alignment

Figure 6.2. Samples from ISEA alignment represented using the first two components (PCA1 and PC2) from PCA (created with [15]).

6.2 Classification results

In this section, the results from the two classification approaches (binary and multiclass) are reported in different sections. For each classifier the results consider all the combination of two sizes of the datasets (reported in table 5.6) together with the following isoforms combinations (i.e. columns of the matrices):

- From TCGA(miRNAs):
  - I5P,I3P, I3P-I5P, EX and logRPM.
  - I5P,I3P, EX and logRPM.
  - I3P, EX and logRPM.
  - EX and logRPM.
  - logRPM only.
- From ISEA (miRNA groups):
  - All (all the isoform in table 5.5.)
  - I5P,I3P, I3P-I5P, EX and logRPM.
  - I5P,I3P, EX and logRPM.
  - I3P, EX and logRPM.
  - EX and logRPM.
  - logRPM only.

Results for each classifier are represented with 2 groups of boxplots (example in figure 6.3), one for each alignment procedure (TCGA on the left and ISEA on the right). Each group contains 4 boxplots, each boxplot comes from a set of a particular averaged cross-validation metric (Accuracy or F1-score) using a specific dataset (>10 RPM datasets or >20 RPM datasets) of all the combination of isoforms. The first boxplot to the left in the group derived from the mean cross-validation accuracies of all the classifier (one for each set of isoforms) using the >10 RPM dataset from the alignment procedure related to the group. The second boxplot derived from the mean cross-validation accuracies of all the classifier (one for each set of isoforms) using the >20 RPM dataset from the alignment procedure related to the group. The third boxplot derived from the mean cross-validation f1-scores of all the classifier (one for each set of isoforms) using the >10 RPM dataset from the alignment

procedure related to the group. The third boxplot derived from the mean cross-validation f1-scores of all the classifier (one for each set of isoforms) using the >20 RPM dataset from the alignment procedure related to the group.
Finally, the details of the best classifier is reported together with the results from Cross-Validation and Test Set. Since Test Set is affected of class imbalance, together with the mentioned metrics, the confusion matrix is also reported. While, F1-score in multiclass approach were computed as the average weighted of F1-scores by support for each class. It accounts for class imbalance and can result in an F1-score that is not between precision and recall.

6.2.1 Binary approach

From the binary approach resulted 6 classifiers overall, derived from the possible combinations of the 4 classes, distinguishing both two types of cancer subtypes (KIRP vs. KIRC, KIRC vs. KICH and KIRP vs. KICH) and normal tissue and cancer subtype specific tissue (KIRC vs. Normal, KIRP vs. Normal and KICH vs. Normal).

KIRP vs. KIRC

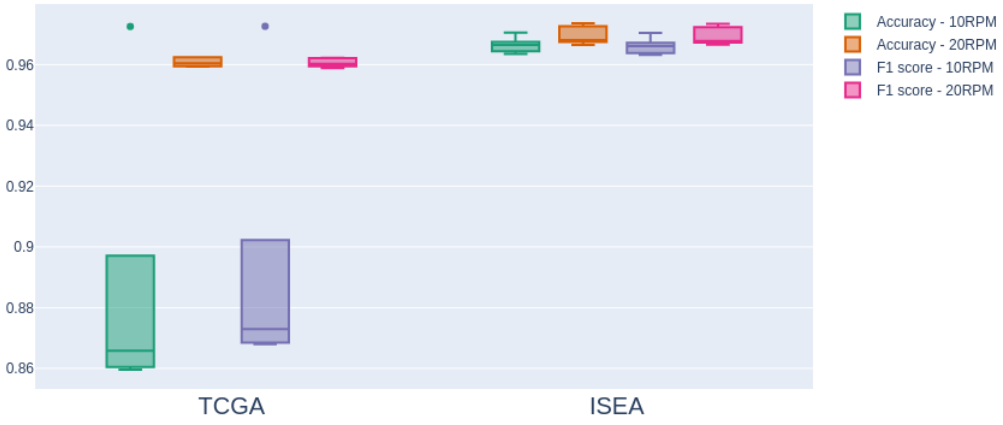


Figure 6.3. KIRP vs. KIRC binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA

- Expression Filter: 20 RPM
- Columns:
  - Isoforms: I5P, I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.97 (+/- 0.02)
  - F1-Score: 0.97 (+/- 0.02)
  - Loss: 0.232 (+/- 0.033)
- **Test Set**:
  - Accuracy: 0.95
  - F1-Score: 0.95
  - Loss: 0.239

Confusion matrix		**P.**	
		KIRP	**KIRC**
A.	**KIRP**	31	1
	KIRC	3	49

KIRC vs. KICH

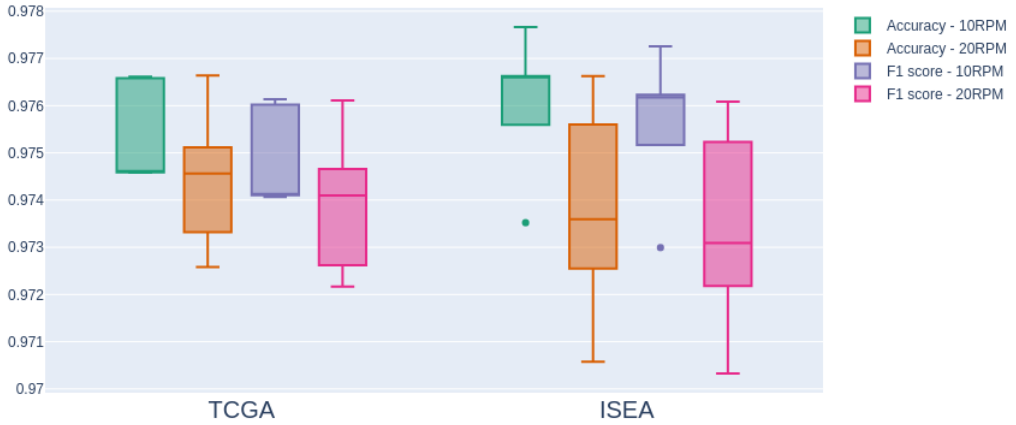


Figure 6.4. KIRC vs. KICH binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: I5P, I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.98 (+/- 0.02)
  - F1-Score: 0.98 (+/- 0.02)
  - Loss: 0.145 (+/- 0.029)
- **Test Set**:
  - Accuracy: 0.95
  - F1-Score: 0.95
  - Loss: 0.126

Confusion matrix		P.	
		KICH	KIRC
A.	KICH	6	1
	KIRC	2	50

KIRP vs. KICH

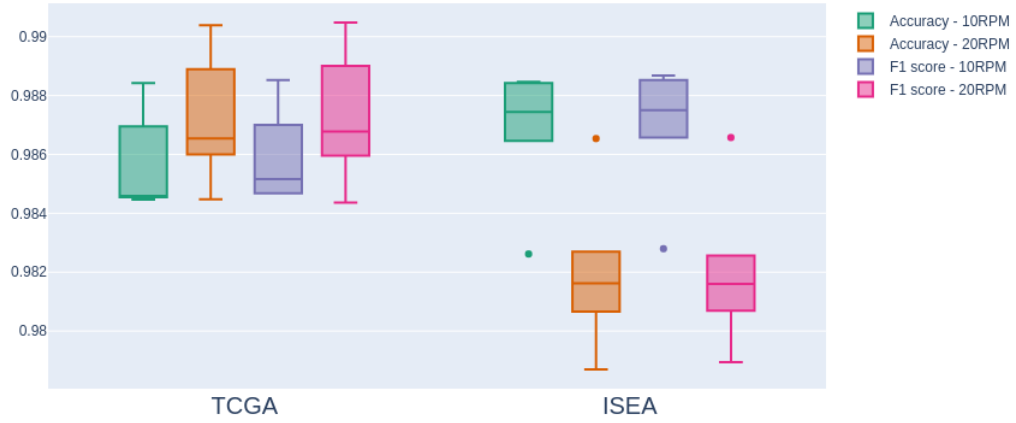


Figure 6.5. KIRP vs. KICH binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: TCGA
- Expression Filter: 20 RPM
- Columns:
  - Isoforms: EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.99 (+/- 0.01)
  - F1-Score: 0.99 (+/- 0.01)
  - Loss: 0.175 (+/- 0.067)
- **Test Set**:
  - Accuracy: 0.97
  - F1-Score: 0.97
  - Loss: 0.09

Confusion matrix		**P.**	
		KIRP	**KICH**
A.	**KIRP**	32	0
	KICH	1	6

KIRC vs. Normal

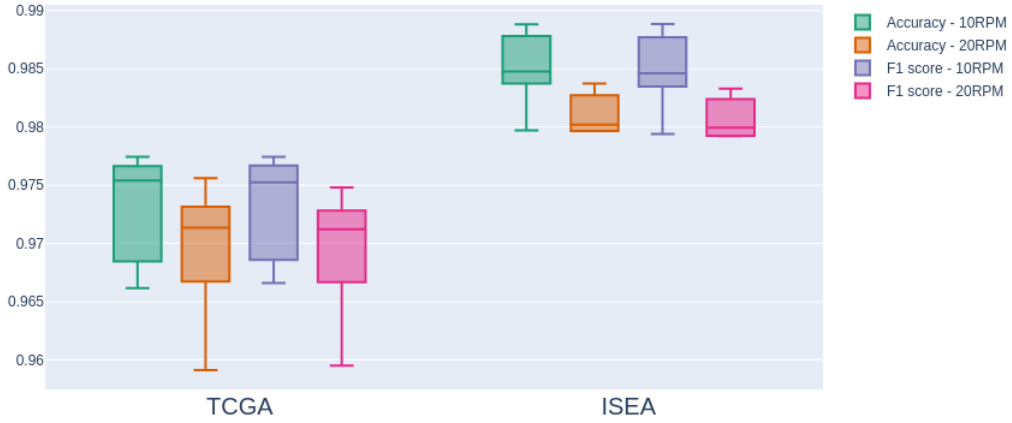


Figure 6.6. KIRC vs. Normal binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: None
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.99 (+/- 0.01)
  - F1-Score: 0.99 (+/- 0.01)
  - Loss: 0.074 (+/- 0.026)
- **Test Set**:
  - Accuracy: 0.98
  - F1-Score: 0.98
  - Loss: 0.052

Confusion matrix		P.	
		Normal	**KIRC**
A.	**Normal**	13	0
	KIRC	1	51

KIRP vs. Normal

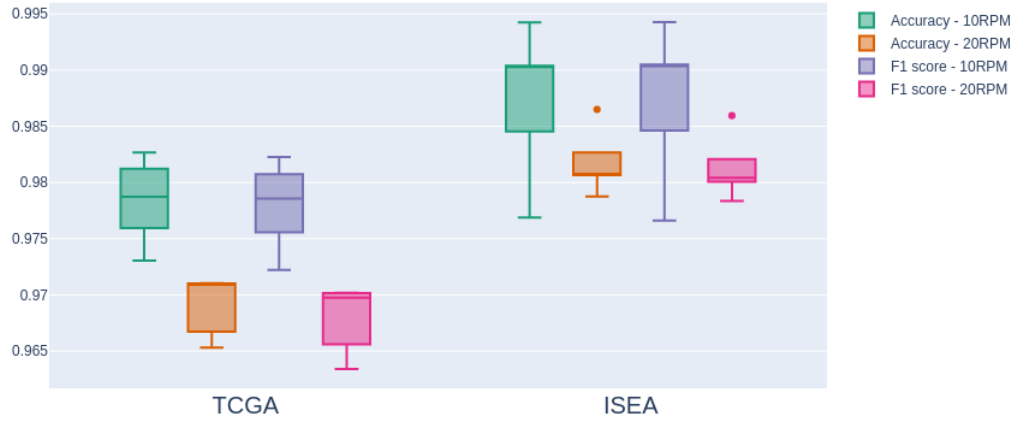


Figure 6.7. KIRP vs. Normal binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: I5P, I3P, I5P-I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.99 (+/- 0.01)
  - F1-Score: 0.99 (+/- 0.01)
  - Loss: 0.191 (+/- 0.047)
- **Test Set**:
  - Accuracy: 1.0
  - F1-Score: 1.0
  - Loss: 0.100

Confusion matrix		P.	
		Normal	**KIRP**
A.	**Normal**	13	0
	KIRP	0	32

KICH vs. Normal

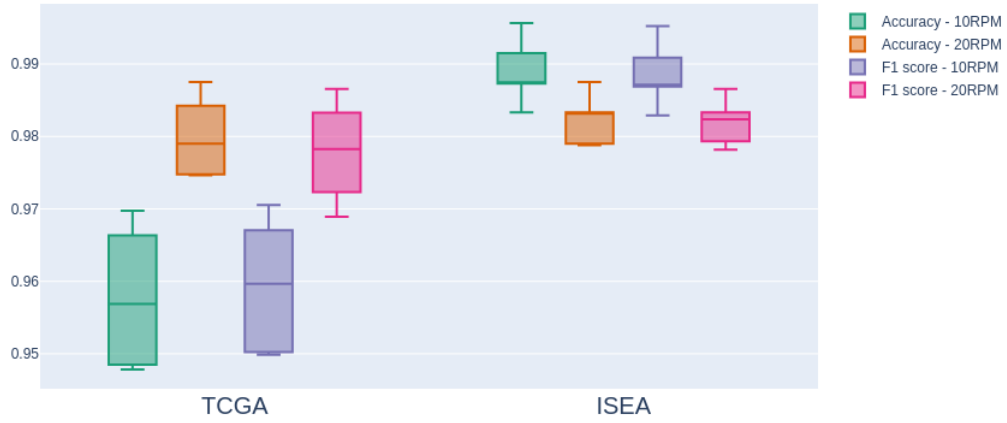


Figure 6.8. KICH vs. Normal binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: I5P, I3P, I5P-I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 1.00 (+/- 0.01)
  - F1-Score: 1.00 (+/- 0.01)
  - Loss: 0.384 (+/- 0.071)
- **Test Set**:
  - Accuracy: 1.0
  - F1-Score: 1.0
  - Loss: 0.286

Confusion matrix		**P.**	
		Normal	**KICH**
A.	**Normal**	13	0
	KICH	0	7

6.2.2 Multiclass approach

In this section the results of the 3 classifiers from the multiclass approach are reported. The classifier **KIRC vs. KICH vs. KIRP** is referred to the classifiers that distinguish a cancer sample into one of the three subtypes, while **KIRC vs. KICH vs. KIRP vs. Normal** to the classifier that distinguish a sample into one of the 4 classes. The results are reported in the same order of the previous section.

KIRC vs. KICH vs. KIRP

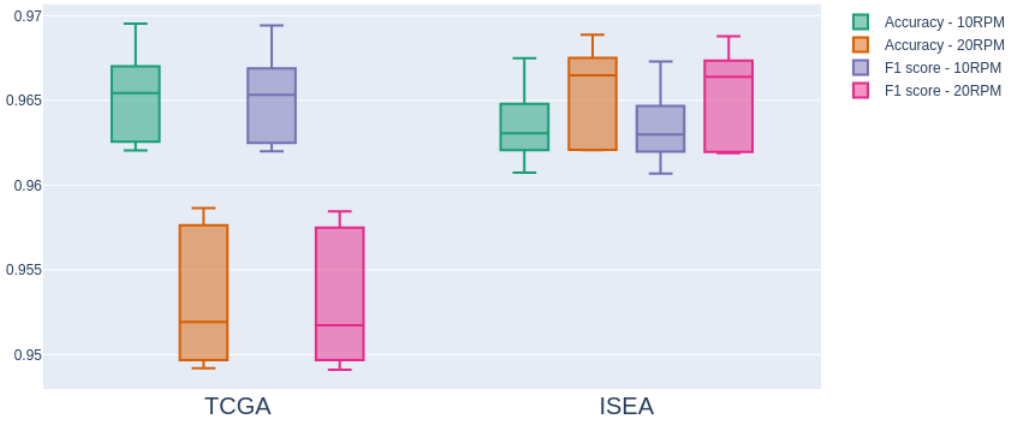


Figure 6.9. KIRC vs. KICH vs. KIRP multiclass classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: TCGA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: I5P, I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.96 (+/- 0.01)

  - F1-Score: 0.96 (+/- 0.01)
  - Loss: 0.328 (+/- 0.084)
- **Test Set**:
  - Accuracy: 0.93
  - F1-Score: 0.93
  - Loss: 0.279

Confusion matrix		**P.**		
		KIRP	**KICH**	**KIRC**
A.	**KIRP**	31	0	1
	KICH	0	6	1
	KIRC	2	2	48

KIRC vs. KICH vs. KIRP vs. Normal

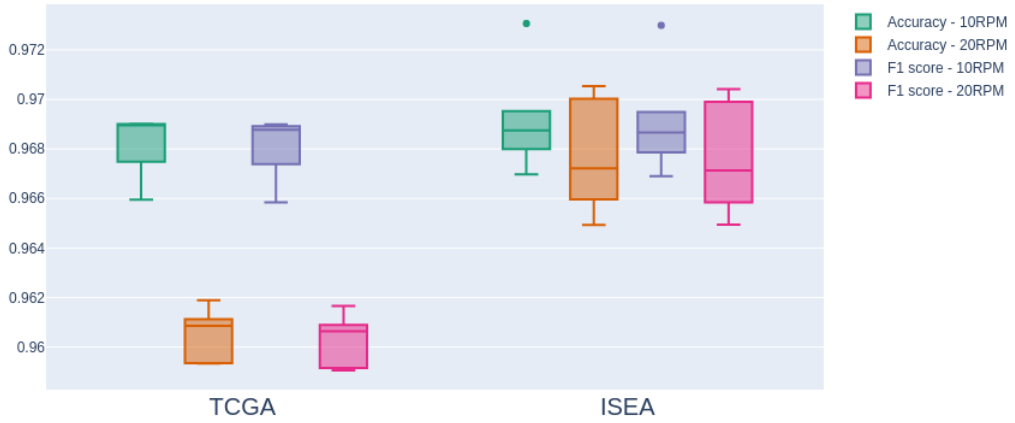


Figure 6.10. KIRC vs. KICH vs. KIRP vs. Normal multiclass classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: None
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.97 (+/- 0.01)
  - F1-Score: 0.97 (+/- 0.01)
  - Loss: 0.184 (+/- 0.030)
- **Test Set**:
  - Accuracy: 0.93
  - F1-Score: 0.93
  - Loss: 0.214

Confusion matrix		P.			
		Normal	KIRP	KICH	KIRC
A.	Normal	13	0	0	0
	KIRP	0	30	0	2
	KICH	0	0	6	1
	KIRC	0	2	2	48

6.2.3 Classification results using a Machine Learning tool

This section reports the results of the same classification tasks using a different well known machine learning method, the Support Vector Machine (SVM) [29][35]. The aim of this approach is to check whether a different tool can obtain comparable results from the same input data but rearranged to the new input dimensionality, that is a monodimensional vector for each input. For this reason a flattening operation as the one reported in figure 5.7 is done to every sample.
The SVMs were trained with the same training set and tested with the same procedure proposed so far, choosing a linear kernel as kernel function and the default parameters reported in the Scikit-Learn library [28].
For the comparison I report the results for those classifiers that had worse performance compared to the others in both approaches, one from binary and one from multiclass, that are **KIRP vs. KIRC** and **KIRC vs. KICH vs. KIRP**.

KIRP vs. KIRC

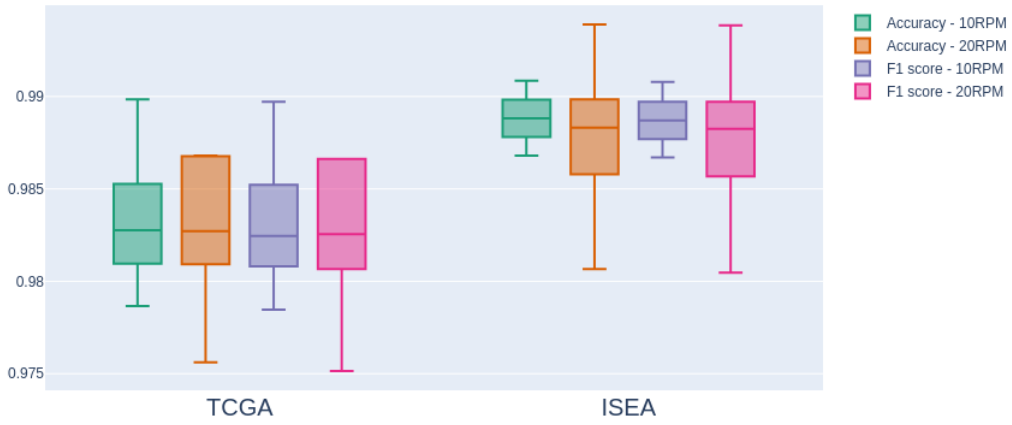


Figure 6.11. KIRP vs. KIRC binary classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 20 RPM

- Columns:
  - Isoforms: I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.99 (+/- 0.01)
  - F1-Score: 0.99 (+/- 0.01)
- **Test Set**:
  - Accuracy: 0.98
  - F1-Score: 0.98

Confusion matrix		**P.**	
		KIRP	**KIRC**
A.	**KIRP**	32	0
	KIRC	2	50

KIRC vs. KICH vs. KIRP

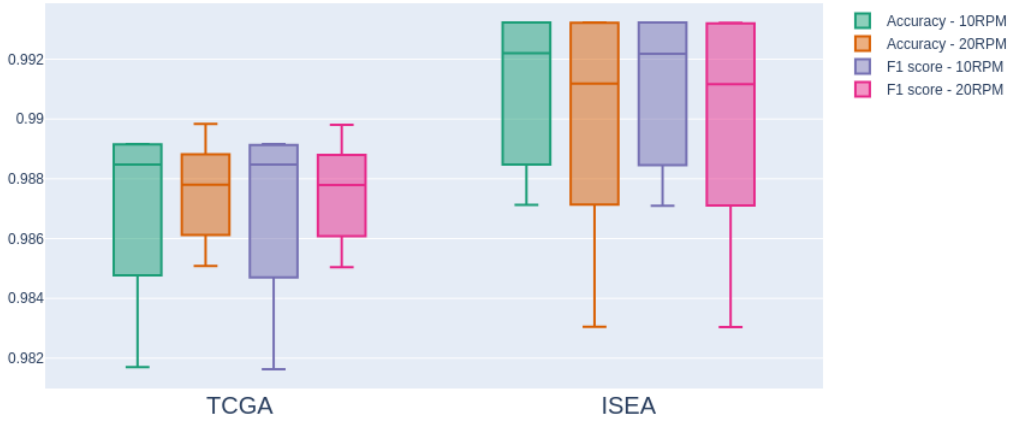


Figure 6.12. KIRC vs. KICH vs. KIRP multiclass classifier results (created with [15]).

The best performances were reported by the following classifier:

- Alignment type: ISEA
- Expression Filter: 10 RPM
- Columns:
  - Isoforms: I5P, I3P, EX
  - logRPM: Yes

Cross validation results and performance on test set:

- **Cross Validation** (mean (+/- standard deviation)):
  - Accuracy: 0.99 (+/- 0.01)
  - F1-Score: 0.99 (+/- 0.01)
- **Test Set**:
  - Accuracy: 0.98
  - F1-Score: 0.98

Confusion matrix		P.		
		KIRP	**KICH**	**KIRC**
A.	**KIRP**	32	0	0
	KICH	0	6	1
	KIRC	1	0	51

6.3 Discussion of the results

From the classification results the following facts can be highlighted: The results from both binary and multiclass approach report an overall accuracy and F1-score grater than 0.9 in both cross-validation and test set with a maximal deviation of 0.03 between the two test procedures. Distinguish normal samples from cancer samples reports the highest performances in terms of accuracy and F1-score in both cross-validation and test set. KIRC vs. Normal classifier reports also the lowest values of Loss (0.074 in cross-validation and 0.052 in Test Set). Overall, datasets from ISEA slightly outperformed TCGA in cross-validation metrics 5 cases out of 6 from the binary approach. These results lead to the following considerations:

- The high classification performances may be related to the apparently good separability of the samples in the 4 classes shown in the PCA graphs for both the alignment procedures (figure 6.1 and figure 6.2).

- The relatively higher separability of the 4 classes showed in the ISEA graph (figure 6.2) may justify the better classification results from ISEA datasets in the binary approach compared to TCGA.

- The relatively lower scattering of the samples in the 4 classes showed in the ISEA graph (figure 6.2) may also justify the apparently lower variability of the results as shown in the boxplots in figures 6.3, 6.5, 6.6, 6.8.

Classification with SVM

The overall classification results using SVM led to better performance compared to CNN, in terms of accuracy and F1 scores metrics up to 1 for both cross-validation and test set in the two approaches (binary and multiclass). From one side it confirmed the goodness of the data collected in describing the sample characteristic for the malignancy detection purpose. On the other side, SVM showed a better separability of the samples in the four classes compared to CNN, starting from the same training set. Nonetheless, the SVM and other machine learning tools won't provide any information about the underlying features that led to the final class, which may be crucial in our case to understand which characteristics (miRNAs) of the input data are related to a specific sample condition. Conversely, deep learning tools allow us to reconstruct the relationship between input data and the final class. In our case, the information related to the single miRNA/miRNA group coming from the isoforms can be retrieved in the output neurons of the convolutional

layer. This information is more reliable the more data we give as input to the network.

Chapter 7

Conclusions

Sample characterization by means of its miRNA isoforms profiles reported promising results in detecting malignancy, regardless of the alignment procedure (ISEA or TCGA). Exploiting datasets containing the expression of isoforms related to miRNA-mRNA interaction sites (from ISEA alignment) didn't show significant improvements compared to less informative datasets in terms of classification performances for this particular case of study (kidney cancer). Nonetheless, graphical representations (both PCA in figures 6.1,6.2 and boxplots A.2) showed a relatively higher separability of the samples among the 4 classes using the more informative datasets (ISEA) compared to the less informative (TCGA). This discrepancy may be explained from the limited number of samples available to train the CNN that may have led ISEA dataset to not outperform on TCGA, since CNNs' training strictly depends on the training set dimensionality, the larger, usually, the better. The proposed method of miRNA isoform sample profiling for cancer detection can be applied to study malignancy from different tissues (bronchus, breast, brain, skin and so on) simply extracting the alignment results and producing the miRNA expression matrices as input of the CNN model. Together with malignancy detection, also other types of sample classification can be taken into account. In general, larger datasets may improve the performances of the CNN and, therefore, the reliability to relate miRNAs/miRNA groups to specific biological pathways.
To conclude, the contributions of the proposed method can be summarized by the following points:

- The method introduce an innovative sample characterization using the miRNA isoforms profile. This characterization can be functional to study specific biological pathways since each miRNA/miRNA group belongs to

specific gene regulation pathways that influence the cell final behavior.

- The classification method proposed with CNN can help identify those miRNA/miRNA groups related to a specific sample characteristic chosen as label. With the growing availability of data, the association can become more reliable.

Acknowledgements

I want to thank my supervisor Gianvito Urgese for giving me the possibility to work with such an interesting topic, allowing me to spread my knowledge in the Bioinformatics world introduced to me by Professor Elisa Ficarra, to whom I show my sincere gratitude.
A huge thank goes to Marta Lovino for her invaluable assistance in helping me to get through this work, her patience and dedication were for me models.

To my colleague and friend Matilde goes my gratitude for the best collaboration and mutual support during these academic years, for the hard work along with never-ending laughs, I'm glad to achieve this award with you.
I wish to thank my previous colleague and friend Marco for all the precious support given, both technical and psychological, during my whole academic career.
My gratitude goes to all my friends with whom I shared the best moments in the lasts years and gave me the support and love that I needed as a human being.
I'd love to mention my special crew: Ana, Daria, Piero, Roberto, and Rosanna, but also Denise, Elena, Mattea, Cristina, and Eleonora.
Each of you reminds me of how essential true friendship can be, no matter how far in time and space.

Last but not at least I wish to express my deepest gratitude to my family, especially my parents, for giving me the possibility to study in Turin in the best conditions I could ask. Your support and trust were for me essential to get through my whole academic career and become the person I am today.
I sincerely hope to make you always proud of me, as son and human being.

Bibliography

[1] URL: https://github.com/bcgsc/mirna.

[2] URL: http://mirbase.org/.

[3] URL: http://www.mirbase.org/blog/2011/04/whats-in-a-name/.

[4] URL: https://www.slideshare.net/dalpozz/racing-for-unbalanced-methods-selection.

[5] David P Bartel. «Metazoan micrornas». In: *Cell* 173.1 (2018), pp. 20–51.

[6] John Besser et al. «Next-generation sequencing technologies and their application to the study and control of bacterial infections». In: *Clinical microbiology and infection* 24.4 (2018), pp. 335–341.

[7] Nitesh V Chawla et al. «SMOTE: synthetic minority over-sampling technique». In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[8] François Chollet et al. *Keras*. https://keras.io. 2015.

[9] Andy Chu et al. «Large-scale profiling of microRNAs for the cancer genome atlas». In: *Nucleic acids research* 44.1 (2016), e3–e3.

[10] Gianpiero Di Leva and Carlo M Croce. «miRNA profiling of cancer». In: *Current opinion in genetics & development* 23.1 (2013), pp. 3–11.

[11] Bastian Fromm et al. «A uniform system for the annotation of vertebrate microRNA genes and the evolution of the human microRNAome». In: *Annual review of genetics* 49 (2015), pp. 213–242.

[12] B Hesper and P Hogeweg. «Bioinformatica: een werkconcept». In: *Kameleon* 1.6 (1970), pp. 28–29.

[13] Paulien Hogeweg. «The roots of bioinformatics in theoretical biology». In: *PLoS computational biology* 7.3 (2011).

[14] Pauline Hogeweg. «Simulating the growth of cellular forms». In: *Simulation* 31.3 (1978), pp. 90–96.

[15] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[16] Ana Kozomara, Maria Birgaoanu, and Sam Griffiths-Jones. «miRBase: from microRNA sequences to function». In: *Nucleic acids research* 47.D1 (2019), pp. D155–D162.

[17] Yong Sun Lee and Anindya Dutta. «MicroRNAs in cancer». In: *Annual Review of Pathological Mechanical Disease* 4 (2009), pp. 199–227.

[18] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. «Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning». In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: http://jmlr.org/papers/v18/16-365.html.

[19] Heng Li. «Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM». In: *arXiv preprint arXiv:1303.3997* (2013).

[20] Heng Li and Nils Homer. «A survey of sequence alignment algorithms for next-generation sequencing». In: *Briefings in bioinformatics* 11.5 (2010), pp. 473–483.

[21] H Li et al. «Genome Project Data Processing Subgroup. 2009. The Sequence alignment/map (SAM) format and SAMtools». In: *Bioinformatics* 1000.25 (), pp. 2078–9.

[22] Wes McKinney. «Data Structures for Statistical Computing in Python». en. In: *Proceedings of the 9th Python in Science Conference*. 2010, pp. 51–56.

[23] K.Jarrod Millman and Michael Aivazis. «Python for Scientists and Engineers». en. In: *Computing in Science and Engineering* 13 (2011), pp. 9–12.

[24] Corine T Neilsen, Gregory J Goodall, and Cameron P Bracken. «IsomiRs–the overlooked repertoire in the dynamic microRNAome». In: *Trends in Genetics* 28.11 (2012), pp. 544–549.

[25] Travis E. Oliphant. *A guide to NumPy*. et. USA: Trelgol Publishing, 2006.

[26] Travis E. Oliphant. «Python for Scientific Computing». en. In: *Computing in Science and Engineering* 9 (2007), pp. 10–20.

[27] Josh Patterson and Adam Gibson. *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.", 2017.

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

[29] Johan AK Suykens and Joos Vandewalle. «Least squares support vector machine classifiers». In: *Neural processing letters* 9.3 (1999), pp. 293–300.

[30] Gianvito Urgese. «Computational Methods for Bioinformatics Analysis and Neuromorphic Computing». PhD thesis. Politecnico di Torino, 2016. URL: http://hdl.handle.net/11583/2646486.

[31] Gianvito Urgese et al. «isomiR-SEA: an RNA-Seq analysis tool for miRNAs/isomiRs expression level profiling and miRNA-mRNA interaction sites evaluation». In: *BMC bioinformatics* 17.1 (2016), p. 148.

[32] John N Weinstein et al. «The cancer genome atlas pan-cancer analysis project». In: *Nature genetics* 45.10 (2013), p. 1113.

[33] Svante Wold, Kim Esbensen, and Paul Geladi. «Principal component analysis». In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[34] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. 2012. arXiv: 1212.5701 [cs.LG].

[35] Dell Zhang and Wee Sun Lee. «Question classification using support vector machines». In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003, pp. 26–32.

[36] Mark Ziemann, Antony Kaspi, and Assam El-Osta. «Evaluation of microRNA alignment techniques». In: *Rna* 22.8 (2016), pp. 1120–1138.

Appendix A

Supplementary materials

A.1 Example of a SQL query

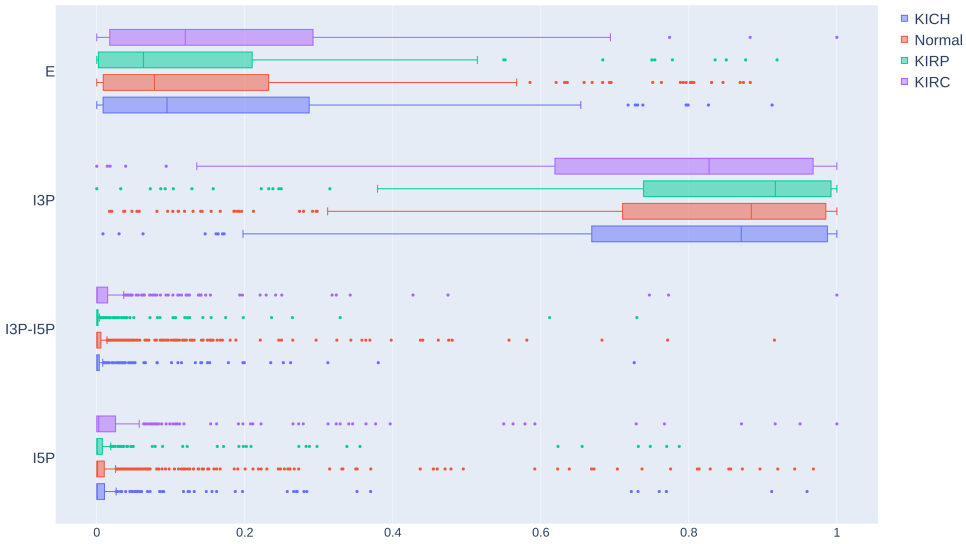
The sample is reported as "xxxx-xxxx-xxxx-xxx"

```
01 | attach 'GeneDB_TI_MIN.db' as db1
02 | Select "MIN".MIN, "I5P-MS-I3P".TC as "I5P-MS-I3P", "I5P-MS".TC as "I5P-MS", "MS-I3P".TC as "
       MS-I3P", "MS".TC as "MS", "MS-IM".TC as "MS-IM", "I5P-MS-IM".TC as "I5P-MS-IM", "MS-IM-
       I3P".TC as "MS-IM-I3P", "I5P-MS-IM-I3P".TC as "I5P-MS-IM-I3P", "I5P-IM-I3P".TC as "I5P-
       IM-I3P", "IM-I3P".TC as "IM-I3P", "I5P-IM".TC as "I5P-IM", "IM".TC as "IM", "I5P".TC as
        "I5P", "I3P".TC as "I3P", "E".TC as "E", "TOT".TC as "TOT"
03 | from (select MIN from db1.TI_MIN order by MIN) "MIN"
04 | left outer join(select distinct MIN,MII from Sample_MII) "MII"
05 | on "MIN".MIN="MII".MIN
06 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 AND
        I3P!=0 AND INS=="T" AND IMS=="F" and sample_isea_db_id=="xxxx-xxxx-xxxx-xxx" group by
       MII) "I5P-MS-I3P"
07 | on "MII".MII="I5P-MS-I3P".MII
08 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 AND
        I3P==0 AND INS=="T" AND IMS=="F" and sample_isea_db_id =="xxxx-xxxx-xxxx-xxx" group by
        MII) "I5P-MS"
09 | on "MII".MII="I5P-MS".MII
10 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 AND
        I3P!=0 AND INS=="T" AND IMS=="F" and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group
       by MII) "MS-I3P"
11 | on "MII".MII="MS-I3P".MII
12 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 AND
        I3P==0 AND INS=="T" AND IMS=="F" and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group
       by MII) "MS"
13 | on "MII".MII="MS".MII
14 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 AND
        I3P==0 AND INS=="T" AND IMS=="T" and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group
       by MII) "MS-IM"
15 | on "MII".MII="MS-IM".MII
16 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 AND
        I3P==0 AND INS=="T" AND IMS=="T" and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group
       by MII) "I5P-MS-IM"
17 | on "MII".MII="I5P-MS-IM".MII
18 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 AND
        I3P!=0 AND INS=="T" AND IMS=="T" and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group
       by MII) "MS-IM-I3P"
19 | on "MII".MII="MS-IM-I3P".MII
20 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 AND
        I3P!=0 AND INS=="T" AND IMS=="T" and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group
       by MII) "I5P-MS-IM-I3P"
21 | on "MII".MII="I5P-MS-IM-I3P".MII
22 | left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 AND
        I3P!=0 AND (IMS=="T" or ISN=="T") AND INS=="F" and sample_isea_db_id == "xxxx-xxxx-
       xxxx-xxx" group by MII) "I5P-IM-I3P"
23 | on "MII".MII="I5P-IM-I3P".MII
```

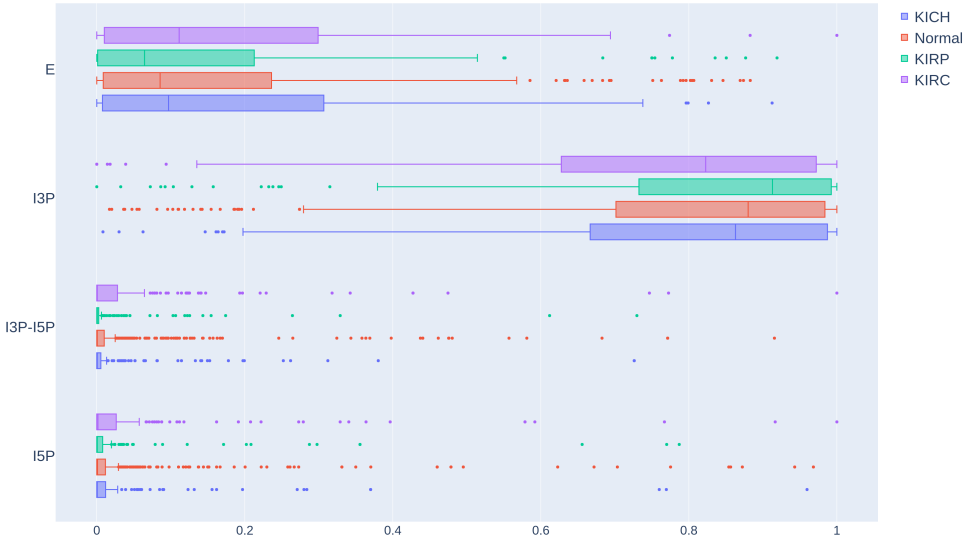
```
24 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 AND
         I3P!=0 AND (IMS=="T" or ISN=="T") AND INS=="F" and sample_isea_db_id == "xxxx-xxxx-
       xxxx-xxx" group by MII) "IM-I3P"
25 |  on "MII".MII="IM-I3P".MII
26 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 AND
         I3P==0 AND (IMS=="T" or ISN=="T") AND INS=="F" and sample_isea_db_id == "xxxx-xxxx-
       xxxx-xxx" group by MII) "I5P-IM"
27 |  on "MII".MII="I5P-IM".MII
28 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 AND
         I3P==0 AND (IMS=="T" or ISN=="T") AND INS=="F" and sample_isea_db_id == "xxxx-xxxx-
       xxxx-xxx" group by MII) "IM"
29 |  on "MII".MII="IM".MII
30 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P!=0 and
         I3P==0 AND IMS=="F" AND ISN=="F" AND INS=="F" and sample_isea_db_id == "xxxx-xxxx-xxxx
       -xxx" group by MII) "I5P"
31 |  on "MII".MII="I5P".MII
32 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where I5P==0 and
         I3P!=0 AND IMS=="F" AND ISN=="F" AND INS=="F" and sample_isea_db_id == "xxxx-xxxx-xxxx
       -xxx" group by MII) "I3P"
33 |  on "MII".MII="I3P".MII
34 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where IEX=="T"
       and sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group by MII) "E"
35 |  on "MII".MII="E".MII
36 |  left outer join (select MII,sum(`TC (Sum)`) AS TC from Sample_MII_iso_inter where MII>=0 and
         sample_isea_db_id == "xxxx-xxxx-xxxx-xxx" group by MII) "TOT"
37 |  on "MII".MII="TOT".MII
```

A.2 Input data boxplots

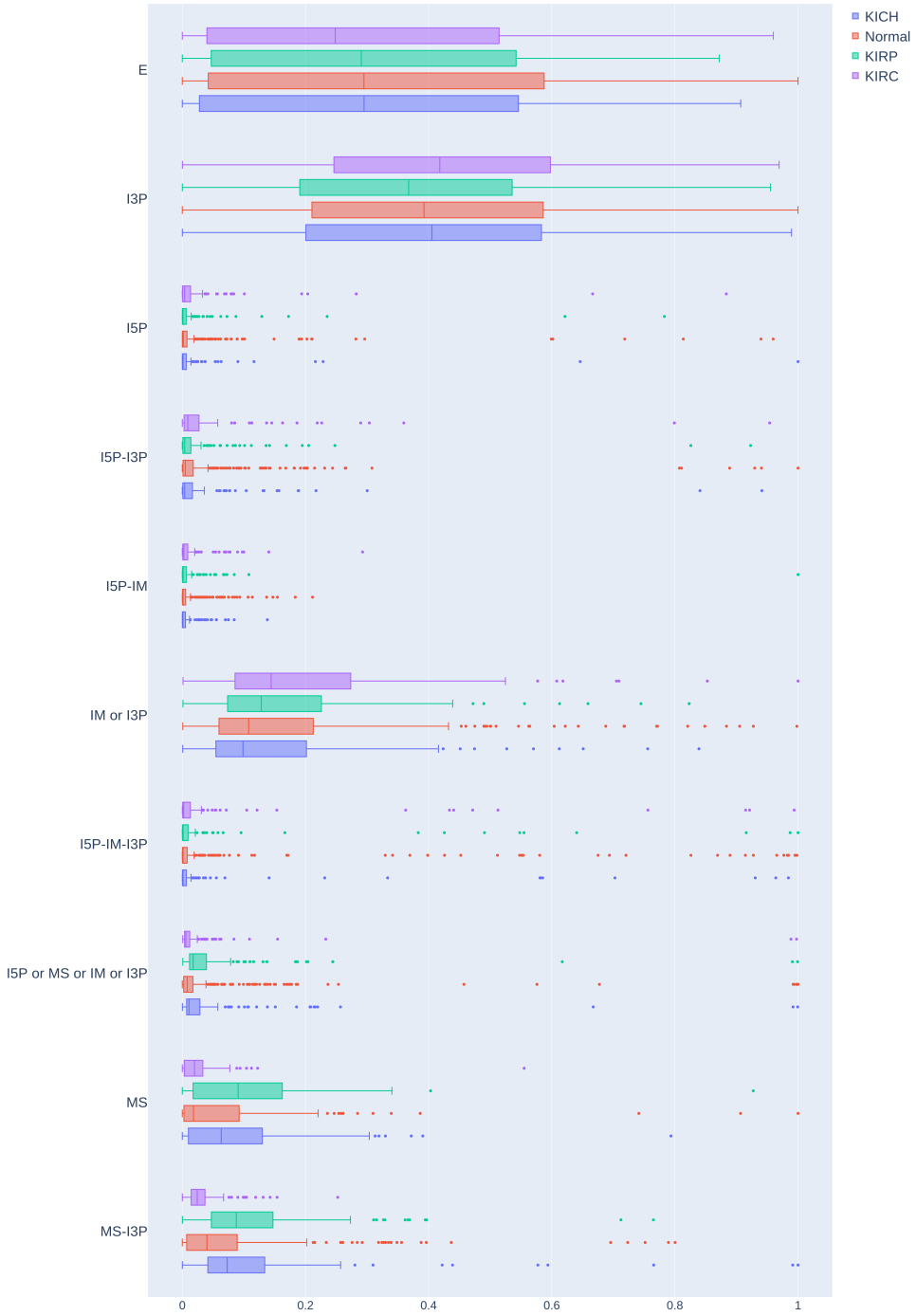
TCGA alignment - 20 RPM



TCGA alignment - 10 RPM



ISEA alignment - 20 RPM



ISEA alignment - 10 RPM