



UNIVERSITÀ DEGLI STUDI DI MILANO
Facoltà di Scienze e Tecnologie
Dipartimento di Informatica

DEEPMIRNA: A NOVEL APPROACH TO MICRORNA TARGET PREDICTION USING DEEP LEARNING

Relatore: Prof. Sebastiano Vigna

Correlatore: Prof. Paolo Boldi

Tesi di:
Simone Sinigaglia
Matricola: 902960

Anno Accademico 2018/2019

I would like to thank ...

Contents

1	MicroRNAs and their importance in living beings	1
1.1	What are microRNAs?	1
1.1.1	Transcription and processing of miRNAs	2
1.1.2	RNA-induced silencing complex (RISC)	2
1.2	Why are they important?	3
1.3	Why miRNAs target prediction is a difficult task?	4
1.4	acaso	4
1.4.1	Folders	4
1.4.2	Tables	4
1.5	In Closing	5
2	MicroRNAs target prediction computational methods	7
2.1	Introduction	7
2.2	MiRNA target prediction	8
2.2.1	Features and methodologies	8
3	Experimental setup	13
3.1	Introduction	13
3.2	Preparing for training	14
3.2.1	Data preprocessing	16
3.2.2	Dataset preparation	18
3.2.3	Training dataset	21
3.3	The testing stage	23
3.3.1	Candidate site selection methods: CSSM	24
3.3.2	Longest Common Subsequence implementation	25
3.3.3	Filtering predictions	27
3.3.4	Obtaining the final classification	28
4	Neural network design and implementation	30
4.1	External libraries	30

4.2	Choosing the right model	30
4.3	The feed-forward network	31
4.4	The Convolutional Neural Network aka CNN	34
4.4.1	CNN architecture	35
5	Experimental results	36
5.1	Introduction	36
5.2	Neural Network evaluation	36
5.3	The role of the candidate site selection method	37
5.4	The importance of site accessibility filtering	38
A	Frequently Asked Questions	40
A.1	How do I change the colors of links?	40

Chapter 1

MicroRNAs and their importance in living beings

1.1 What are microRNAs?

MicroRNAs (abbreviated miRNAs) are a family of ≈ 22 -nucleotide small non-coding RNAs that regulates gene expression at the post-transcriptional level [4]. This means that they act by binding to partially complementary sites on target genes, which had been previously transcribed from the DNA of the cell, to induce cleavage or repression of productive translation, preventing this way the target gene to be able to exit the cell and start the translational process that produces peptides and proteins.

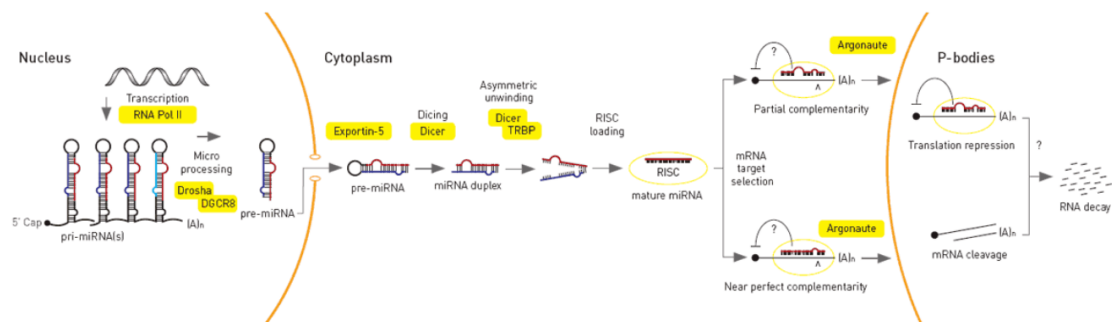


Figure 1: MiRNAs genesis and functionalities

1.1.1 Transcription and processing of miRNAs

As shown in figure 1 miRNAs genes are transcribed by the RNA polymerase II as large primary transcripts (pri-miRNA) that are processed by a protein complex containing the enzyme Drosha, to form an approximately 70 nucleotide precursor miRNA (pre-miRNA). This precursor is subsequently transported to the cytoplasm where it is processed by a second enzyme, called DICER, to form a mature miRNA of approximately 22 nucleotides. The mature miRNA is then incorporated into a ribonuclear particle to form the RNA-induced silencing complex, RISC, which mediates gene silencing.

It's important to note that, generally, only one of the two strands of the stem loop is incorporated into the silencing process, and it's selected on the basis of its thermodynamic instability and weaker base-pairing on the 5' end relative to the other strand. The latter, called the passenger strand due to its lower levels in the steady state, is usually denoted with an asterisk (*) and is normally degraded. However, in some cases, both strands of the duplex are viable and become functional miRNAs that target different mRNA populations. (see figure 2)

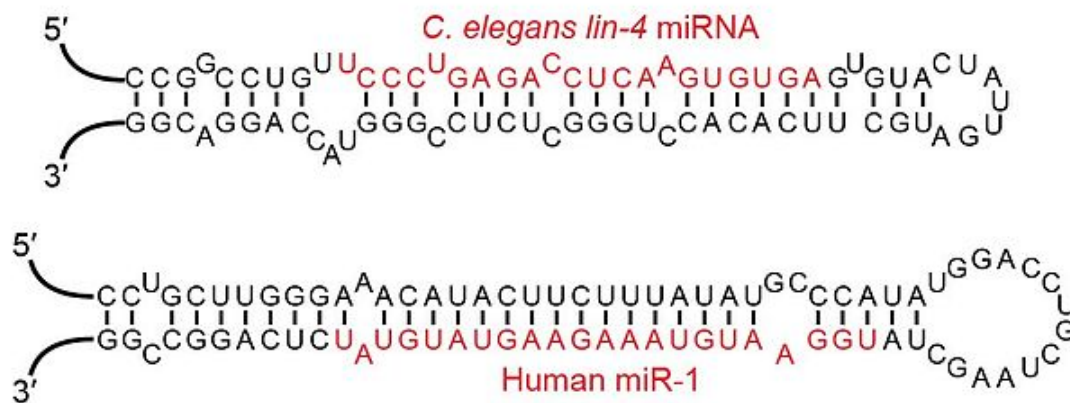


Figure 2: Examples of miRNA stem-loops. In red is shown the mature miRNA

1.1.2 RNA-induced silencing complex (RISC)

As mentioned before the mature miRNA is part of an active RNA-induced silencing complex (RISC). This process represents their main functionality in both animals and plants. The RISC it is a key process in gene silencing and can act in two

different ways as depicted in the right-hand side of picture 1: via mRNA degradation or by preventing mRNA translation. It has been demonstrated that given complete complementarity between the miRNA and target mRNA sequence, Ago2 can cleave the mRNA and lead to direct mRNA degradation. In the presence of only partial complementarity instead, silencing is achieved by preventing translation [25].

1.2 Why are they important?

MiRNAs are particularly abundant in many mammalian cell types and appear to target about 60% of the genes of humans and other mammals [24].

Many miRNAs are evolutionarily conserved, which implies that they have important biological functions [24]. For example, 90 families of miRNAs have been conserved since at least the common ancestor of mammals and fish, and most of these conserved miRNAs have important functions.

The discovery of the first miRNA over 20 years ago has ushered in a new era in molecular biology. There are now over 2000 miRNAs that have been discovered in humans and it is believed that they collectively regulate two third of the genes in the genome.

The repressive action of miRNAs has a huge impact on many biological processes such as cell cycle control and several developmental and physiological processes including stem cell differentiation, cardiac and skeletal muscle development, neurogenesis, insulin secretion, cholesterol metabolism, aging, immune responses and viral replication. [10]

In addition to their important roles in healthy individuals, microRNAs have also been implicated in a number of diseases including a broad range of cancers, heart and neurological diseases. In fact it has been discovered that their expression patterns are highly specific in respect to external stimuli, developmental stage or tissue and this can be used to diagnose diseases in which the expression levels of miRNAs are known to change considerably [39]. Consequently, miRNAs are intensely studied as candidates for clinical diagnosis and predictors of drug response [27].

1.3 Why miRNAs target prediction is a difficult task?

1.4 acaso

1.4.1 Folders

This template comes as a single zip file that expands out to several files and folders. The folder names are mostly self-explanatory:

Appendices – this is the folder where you put the appendices. Each appendix should go into its own separate .tex file. An example and template are included in the directory.

Chapters – this is the folder where you put the thesis chapters. A thesis usually has about six chapters, though there is no hard rule on this. Each chapter should go in its own separate .tex file and they can be split as:

- Chapter 1: Introduction to the thesis topic
- Chapter 2: Background information and theory
- Chapter 3: (Laboratory) experimental setup
- Chapter 4: Details of experiment 1
- Chapter 5: Details of experiment 2
- Chapter 6: Discussion of the experimental results
- Chapter 7: Conclusion and future directions

1.4.2 Tables

Tables are an important way of displaying your results, below is an example table which was generated with this code:

```
\begin{table}
\caption{The effects of treatments X and Y on the four groups studied.}
\label{tab:treatments}
\centering
\begin{tabular}{l l l}
\toprule
\thead{Groups} & \thead{Treatment X} & \thead{Treatment Y} \\
\midrule
1 & 0.2 & 0.8\end{tabular}
```


Table 1: The effects of treatments X and Y on the four groups studied.

Groups	Treatment X	Treatment Y
1	0.2	0.8
2	0.17	0.7
3	0.24	0.75
4	0.68	0.3

```

2 & 0.17 & 0.7\\
3 & 0.24 & 0.75\\
4 & 0.68 & 0.3\\
\bottomrule\\
\end{tabular}
\end{table}

```

You can reference tables with `\ref{<label>}` where the label is defined within the table environment. See `Chapter1.tex` for an example of the label and citation (e.g. Table 1).

There are many different \LaTeX symbols to remember, luckily you can find the most common symbols in The Comprehensive \LaTeX Symbol List.

You can write an equation, which is automatically given an equation number by \LaTeX like this:

```

\begin{equation}
E = mc^{\{2\}}
\label{eqn:Einstein}
\end{equation}

```

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by \LaTeX . If you don't want a particular equation numbered, use the unnumbered form:

```
\[ a^{\{2\}}=4 \]
```

1.5 In Closing

You have reached the end of this mini-guide. You can now rename or overwrite this pdf file and begin writing your own `Chapter1.tex` and the rest of your thesis. The easy work of setting up the structure and framework has been taken care of for you. It's now your job to fill it out!

Good luck and have lots of fun!

Guide written by —
Sunil Patel: www.sunilpatel.co.uk
Vel: LaTeXTemplates.com

Chapter 2

MicroRNAs target prediction computational methods

2.1 Introduction

Earlier in Chapter 1 we described how miRNAs play a fundamental role in gene regulation. It is common belief that the final and probably most relevant step in their regulatory pathway is targeting [39]. Targeting is intended as the binding of the mature miRNA to the messenger RNA via the RNA Induced Silencing Complex (see figure 3). Hence, valid targets need to be identified for miRNAs in order to properly understand their role in cellular pathways.

However, many of the discovered miRNAs do not yet have identified targets. This is especially the case in animals where the miRNA does not bind to its target with a nearly perfect matching as it does in plants [31]. Experiments have proved that a single miRNA has the potential to regulate hundreds of target mRNAs and multiple miRNAs may compete for the regulation of the same mRNA [2], however target validation is difficult, expensive, and time consuming. Thus, having considered all these facts, it is of crucial importance to have accurate computational miRNA target predictions.

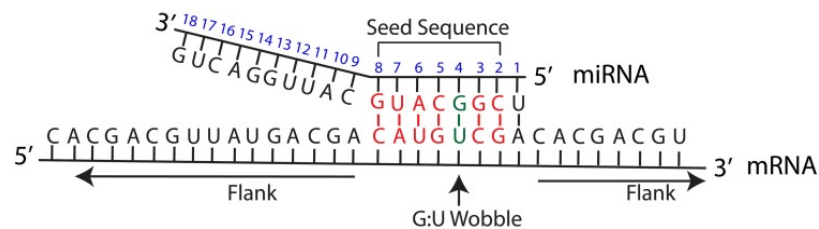


Figure 3: Example of miRNA targeting.

2.2 MiRNA target prediction

Before miRNA target prediction tools were available, possible miRNA binding sites were determined manually. These target sites were later confirmed by laborious and inefficient techniques such as site-directed mutagenesis and other experimental methods. The identification of the first targets for the let-7 and lin-4 miRNAs led to the idea that miRNAs have a pattern in targeting genes which could be used to develop target prediction algorithms [29].

Originally gene targeting by miRNAs was believed to be the result of their binding to the 3'UTR of the target mRNA [2], however recent studies [12] have confirmed gene regulation as a result of the binding of the miRNA to the coding region as well as to the 5'UTR. Furthermore, computational evidence suggests that regulation via the binding of the miRNA to the coding region differs in comparison to the binding pattern seen at the 3'UTR. In particular, it's suggested that miRNAs target the coding regions of mRNAs with short 3'UTRs [34].

Another key factor in target prediction is that 3'UTRs are prone to change under different conditions which might result in the elimination of the target site. Binding in the coding region on the other hand may instead present an evolutionary advantage for the cell as it could help in the preservation of the miRNA binding site [28].

2.2.1 Features and methodologies

While many miRNA targets have been computationally predicted only a limited number have been experimentally validated. Moreover, although a variety of miRNA target prediction algorithms are implemented, results amongst them are generally inconsistent and correctly identifying functional miRNA targets remains a challenging task.

The average performance of target prediction tools, which typically identify approximately 80% of known miRNA targets, indicates that the mechanisms associated with miRNA-regulated processes remain poorly understood. Thus, there is a room for novel approaches to improve the knowledge of the rules that govern their targeting process [18]

The various methodologies implemented use several different approaches and analyze a wide range of features for this task. Almost all target prediction methods are rule-based or adopt machine learning methodology with varying success. Rule-based systems incorporate various human-crafted descriptors to represent miRNA:gene target binding (e.g. type of pairs in the site, binding stability, or conservation of the target site among species). Machine learning techniques also use those descriptors, but as input features to machine learning models. The limitation

of both these approaches is indeed the process of feature selection and representation, which is constrained by the use of human selected descriptors to model a process that is not fully understood.

The most common characteristics used in miRNA targets identification are [32]:

- seed region complementarity
- free energy
- site accessibility
- conservation

The seed region

Targeting patterns are very different between plants and animals. Plants, in fact, show a near perfect complement between their miRNA and the respective target mRNA. On the other hand animal miRNAs bind their targets with only partial complementarity. In particular, a region of about 6 to 8 nucleotides in length at the beginning of the miRNA is of crucial importance in the targeting. This short subsequence is called *seed region* and it comprises the nucleotides between the second at the eighth (the seed sequence in Figure 3) starting from the 5' end. The seed region is very important because it binds to the target mRNA leading to the regulation of the gene in question [36].

Undoubtedly the seed region is one of the most commonly used miRNA traits for target prediction. This seed-centric view, in fact, has been supported by structural studies [35] and a widely cited report [8] that investigated the importance of other (non-canonical) regions within a miRNA concluding that their contributions had relatively low relevance compared to the (canonical) seed region. More recent experiments, however, have highlighted a role for the entire miRNA, suggesting that a more flexible methodology is needed [14].

Free energy

The free energy, also called hybridization energy, is defined as the energy released by the pairing between the miRNA and mRNA and it can be used as the measure of the stability of the bond. In fact a stable bond is considered more likely to be a functional target of the miRNA. However, since measuring this quantity directly is difficult, usually the change of free energy during a reaction is considered (ΔG). Reactions with a negative ΔG have less energy available to react in the future, hence they result in systems with an increased stability. By predicting how the miRNA and its candidate target hybridize, regions of high and low free energy

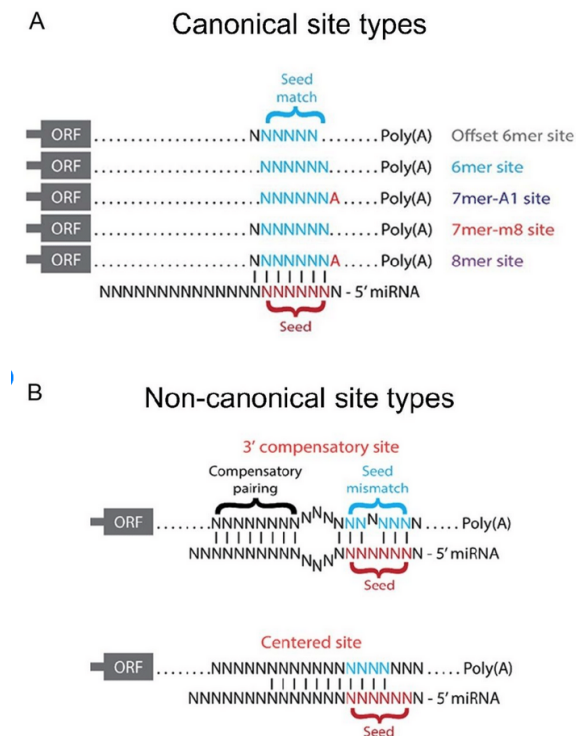


Figure 4: Example of canonical and non-canonical binding sites.

can be inferred (Figure 5) and the overall ΔG can be used as an indicator of how strongly bound they are [42].



Figure 5: A hairpin loop is shown with the loop corresponding to a region of high free energy (a positive ΔG) and the stem corresponding to a region of low free energy (a negative ΔG)

Site accessibility

Site accessibility is the measure of the ease with which a miRNA can locate and hybridize with its target. After transcription, in fact, a mRNA assumes a certain secondary structure which can interfere with the miRNA ability to bind to its target site. To understand why this is important, we need to consider that the miRNA:mRNA hybridization involves a two-step process in which a miRNA firstly binds to a short accessible region of the mRNA and only after, while the secondary structure of the mRNA unfolds, completes the binding. It is likely that secondary structures contribute to target recognition, because there is an energetic cost to freeing base-pairing interactions within mRNA in order to make the target accessible for miRNA binding. Hence, to assess the likelihood that a mRNA is a target of a given miRNA, the predicted amount of energy required to make the site accessible (the so called site accessibility energy SAE) should be taken into consideration [17].

The SAE can be computed as the difference between the free energy cost of opening the mRNA and free energy gained from the intermolecular interaction (Figure 6).

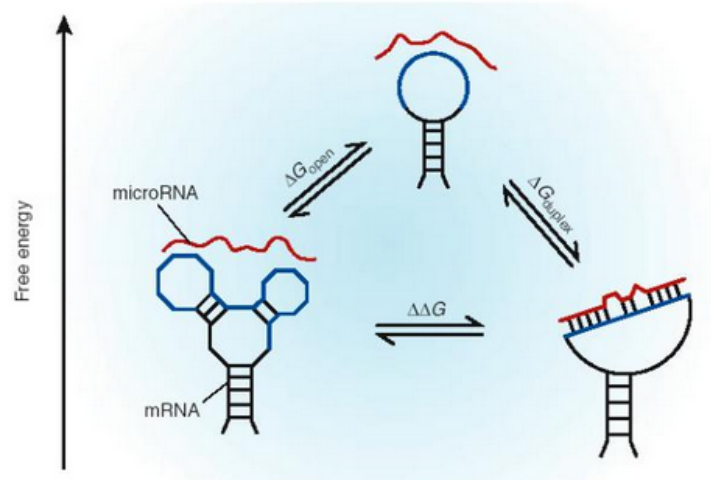


Figure 6: Binding of a miRNA to its target mRNA is depicted as a 2-step process. Portion of the mRNA structure must be open before miRNA:mRNA base pairing can be established.

Conservation

Conservation refers to the maintenance of a sequence across species. According to many reports [39] looking at conserved targets between different species helps reducing the number of false positive results. However, other more recent studies highlighted the fact that this may also increase the number of false negatively identified targets [24].

Chapter 3

Experimental setup

3.1 Introduction

The majority of prediction tools are based on the assumption that it is the miRNA seed region that contains almost all the important interactions between a miRNA and its target and their focus are on these canonical sites.

In this thesis, we aim at using deep learning techniques to investigate the role of those non-canonical sites and pairing beyond the canonical seed region in miRNA targets. Recent increases in computational power have permitted the rise of methods that can dispense with human-crafted features, making it possible to deal directly with raw data and autonomously learn and identify patterns to appropriately represent it. In particular, deep learning has been shown to be an effective method for classification tasks in domains with complex feature representation [22].

Deep Learning (DL) has already been applied to the miRNA target prediction problem. Cheng et al. [6] used convolutional neural networks to analyze matrices of miRNA:site features, but the selected features were still human-crafted descriptors and thus the method faces similar problems as rule-based and ML approaches. A more recent work, DeepTarget [23], relied on recurrent neural networks to identify potential binding sites and assess their functionality. However this work is still oriented to the identification of canonical sites and relies on a limited small data set for the training phase. Another approach using DL is DeepMirTarSdA [40], that explore the use of stacked de-noising auto-encoders (SdA) to predict human miRNA-targets at the site level, but the network obtained is huge and the small availability of input data (about 8000 samples between positives and negatives) results in a model that performs well on the data being used but generalizes quite poorly.

In this thesis we present DeepMiRNA, a miRNA target prediction tool that attempts to take advantage of the learning capacity of a neural network to extract abstract patterns from raw input data. Unfortunately, to the best of our knowledge, there is no suitable raw-data representational method for miRNA-target prediction. This is very likely the consequence of the very small quantity of validated data available for this task. For this reason, rather than making assumption about suitable descriptor, we created a set of rules to help finding the best candidate binding sites leaving the classification decision to the neural network (see CSSM in the next section).

More precisely, DeepMiRNA scans the 3'UTR of the gene identifying potential target sites according to the chosen rules. It then uses the previously trained network to identify the relevant patterns by directly examining the whole mature miRNA transcript, rather than focusing on the seed region and analyzing precomputed descriptors. In the following text we denote a miRNA binding site with MBS.

3.2 Preparing for training

Two of the fundamental properties in deep neural network theory states that:

1. with sufficient data samples and a correct network design a NN can approximate any mathematical function
2. a NN has the capacity to automatically learn the relevant features of complex data structures by means of its hidden layers [22]

For these reasons, in our approach we sought to minimize potential biases introduced by handcrafted features by working with the miRNA and the mRNA transcripts, feeding them directly to the neural network.

The DeepMiRNA working pipeline for the identification of functional targets for miRNAs can be summarized as follows (Figure 7: a 30-nucleotide sliding window with stride of 5 nucleotides is used to scan the 3'UTR of a given gene. Both values (size and stride) has been empirically computed during the training stage (see the results section for more informations). For each mRNA 30-nucleotide long subsequence, the VIENNA RNACofold package [15] is used to compute the stability of the binding between the miRNA transcript and the fragment. If the computed value is below a predetermined threshold, the primary structure of the mRNA and the miRNA are examined to see if the criteria defined by the candidate site selection rules (CSSR) are met. If so, the duplex is vectorized and fed into the network for classification. The prediction is then further refined using an a posteriori filter that computes the site accessibility of the mRNA region surrounding

the predicted binding site. This last step has revealed an important role in false positive reduction and it's used only for positive predictions.

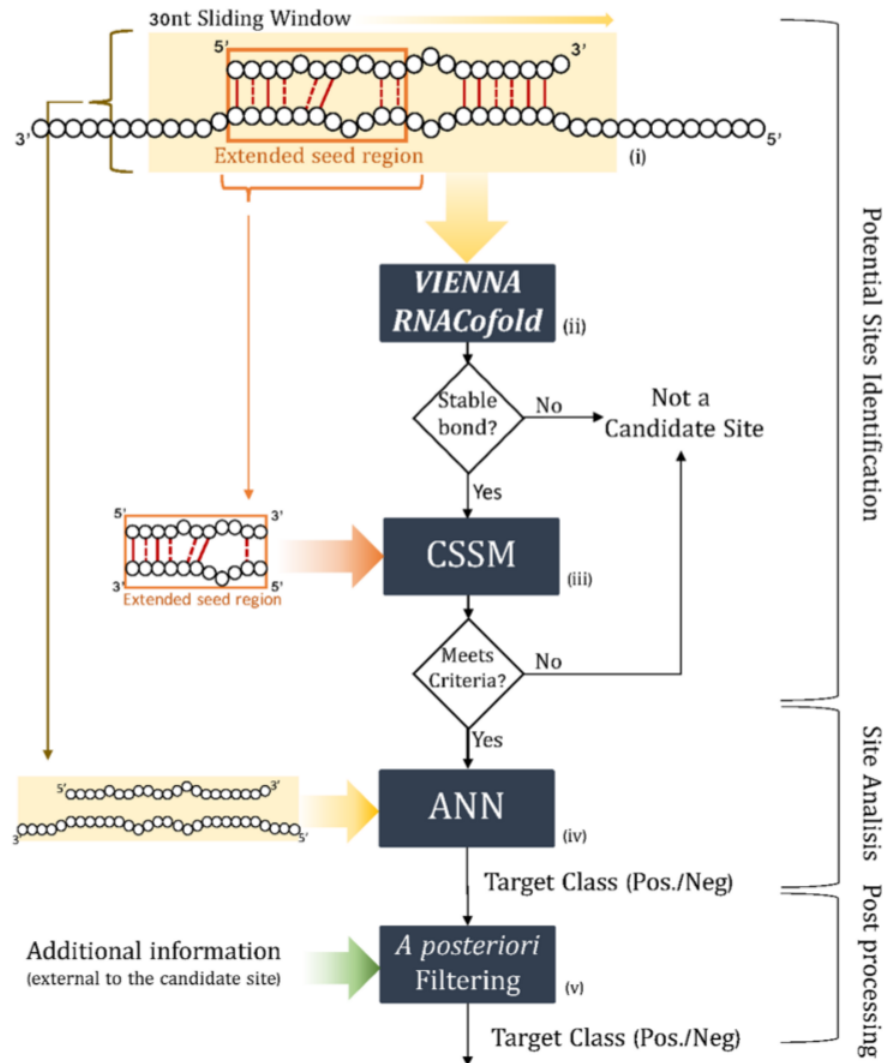


Figure 7: **DeepMiRNA pipeline.** (i) a 30nt sliding window with a 5nt step is used to scan the gene transcript; (ii) The Vienna Cofold software is used to compute the binding stability; (iii) if the bond is predicted to be stable a partial complementarity according to the rules defined in the CSSR is verified; (iv) if all previous checks passed the duplex is fed to the NN for prediction; (v) for positive predictions a filter is used to compute the site accessibility of the miRNA: if the energy needed to access the site is above a certain threshold the prediction is changed to negative.

3.2.1 Data preprocessing

A key factor for successful application of any Machine Learning technique and one of the most important aspect to consider before feeding the network with data is how we prepare the input data and the targets.

The main aspect to consider during this process is the use of suitable techniques to make the data more amenable for the NN: this includes vectorization, normalization and handling of missing values.

vectorization

In order to be accepted by the neural network all inputs must be tensors, that is they must be expressed by numerical arrays with suitable shape and dimension. The process to encode categorical data, in our case the transcript sequences, into tensor is called *vectorization*. In this thesis we actually employed two different techniques for this operation:

- one hot encoding of the sequences [1].
- sequence embedding using a Word2Vec approach [11].

One hot encoding is a process by which categorical data such as strings are converted into binary vectors. In our case, each nucleotide is translated to a binary vector of size 4, corresponding to the four possible nucleotide values as described in Table 2

The main problem using this method is that not all duplexes have the same size. This is in particular due to the different miRNA's transcript length (ranges from 18 to 30), The network, instead, requires that all inputs have the same shape. Hence, in order to meet this requirement, every miRNA sequence, when needed, has been padded with 'empty' letters to reach the maximum size length (in this case 30). Regarding the site transcript, each fragment has size 40: 30 corresponding to the

Table 2: One Hot encoding of a nucleotide.

Nucleotide	Encoding
A	[1, 0, 0, 0]
C	[0, 1, 0, 0]
G	[0, 0, 1, 0]
U or T	[0, 0, 0, 1]
Empty	[0, 0, 0, 0]

window size plus 5 additional nucleotides upstream and downstream. These additional nucleotides seek to capture any influence that the flanking sequence may exert on the target [24]. With these adjustments each duplex is represented by a binary vector of (fixed) size 280.

The second vectorization method uses a Word2Vec approach. Whereas the vectors obtained through one-hot encoding are binary and sparse (mostly made of zeros), sequence embeddings are usually dense, very low-dimensional floating-point vectors.

Word2Vec [11] is a Natural Language Processing (NLP) methodology to map words into numeric vectors based on their context. Being the context defined as the words surrounding the word to encode. For DNA sequences, however, there is no clear definition for words, so usually a k -mer (that is a set of k continuous nucleotides) is used to define a word (see Figure 8). Therefore, in case of biological sequences the context is defined as the set of n adjacent k -mers (being n a parameter to validate). For this thesis use the software available at <https://github.com/pnnpnpn/dna2vec> to train the model used to encode the k -mers. This encoding has two important advantages compared to one hot encoding:

1. each k -mer of length comprised between 3 and 8 is mapped to an equal size vector of size 100.
2. similar k -mers are mapped to close points in the features space according to a specific distance metric (usually Euclidean distance).

In our case each variable length miRNA sequence has been split into 4 different size k -mers each mapped into a 100-dimension vector, while each fixed size site transcript (plus the flanking nucleotides) has been split into 5 8-mers. This way each duplex is mapped into a 9×100 matrix obtained concatenating the resulting 9 vectors. It's important to note that this vectorization requires a different design and implementation of the neural network to use as we will describe in the next section.

normalization and missing values

Another crucial part of data preprocessing concerns their normalization. In general, it's not safe to feed the network with data that takes relatively large values or that are heterogeneous (i.e have very different values ranges). In our case, however, the vectorization process guarantees that the numerical values resulting from the encoding are both small and homogeneous. Regarding missing or incomplete data, we found a very small quantity of them in the datasets retrieved, hence we simply decided to discard them.

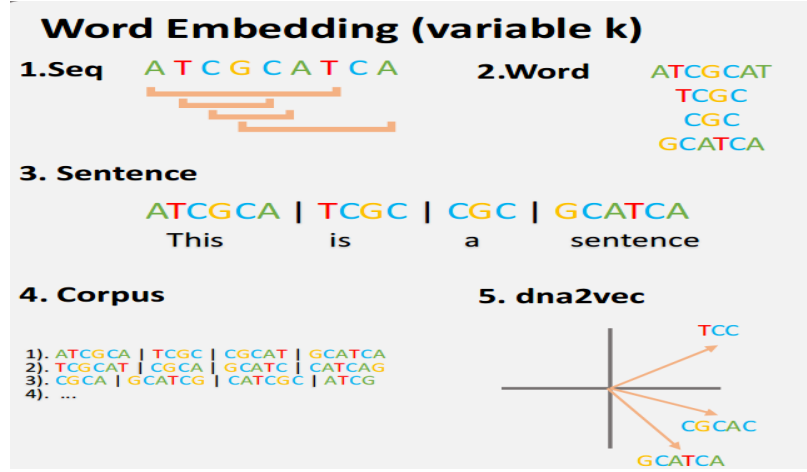


Figure 8: Dna2Vec mapping example using different length k-mers

3.2.2 Dataset preparation

One of the biggest challenge to face in any Machine Learning application is to get a good set of significant data. The main purpose is to have a sufficiently variable and representative dataset that allows the model to generalize well on new and unseen examples. This phase is probably the most important together with the network design as it's crucial for achieving good performances. It is of fundamental importance that the data is representative for the problem we want to solve. It does not matter how big is the quantity of data we obtain if that is not aligned with the goal we want to pursue. One should focus on finding data with features that matter to what we're trying classify or predict and discard unrelated features. Hence, the first step of a classification process should be proper data collection, and until we achieve this, we will find ourselves constantly coming back to this step [20].

For the miRNA target prediction task, this requires a comprehensive dataset of verified positive and negative targets that encompass both canonical and non-canonical examples. While there are multiple data repositories providing information regarding experimentally validated positive miRNA targets[38] [16] there are significantly fewer experimentally verified negative targets. This represents a major concern for ML-based approaches that require similar numbers of labeled examples for both classes.

In this thesis we only considered human data and we used the version 8 of the Diana TarBase¹ and the release 7.0 of miR TarBase² datasets. Moreover, in order to annotate the datasets with lacking informations such as molecules transcripts,

¹<http://diana.imis.athena-innovation.gr/DianaTools/index.php>

²<http://mirtarbase.mbc.nctu.edu.tw/php/index.php>

Table 3: Example of Diana TarBase entries.

gene name	miRNA name	functionality
AB002442	hsa-miR-192-5p	POSITIVE
DAD3R	hsa-miR-34a-5p	NEGATIVE
E2IG4	hsa-miR-200c-3p	POSITIVE

we downloaded gene sequences from Ensembl³ and miRNA transcripts from miR-Base⁴

Diana TarBase

The Diana TarBase is the most widely used database for validated miRNA interactions. The latest version at time of writing is *v8* and asking for a bulk download from their website is possible to obtain a dataset of 526658 positive and 63559 negative validated targets. This dataset provides informations about miRNA:mRNA interactions in the form of miRNA name, gene name and functionality as it's illustrated in Table 3. Unfortunately there is no information about transcript sequences of the molecules or binding sites location.

Mir TarBase

This database contains interactions of about 2000 miRNA and around 20000 genes. In total the downloaded dataset comprises 380091 positive and 382 negative duplexes available from the Download section of the website. It's possible to find it under the name *hsa_MIT.xlsx*. It's important to note that every single miRNA can interact with hundreds of different genes. Again, exactly as with the Diana TarBase dataset, no sequence or binding sites transcripts are provided.

Ensembl

From Ensembl[41] is possible to obtain the transcripts of all human genes. In this thesis we focused on the 3'UTR and for the download we used the BioMart tool available from the main menu. More specifically, from the BioMart webpage we followed these steps:

- from the drop down menu choose database: *Ensembl Genes 96*;

³<https://ensembl.org/index.html>

⁴<http://mirbase.org>

- then choose dataset: *human genes GRCh38.p12*;
- from the left-hand side of the page click attributes and select *sequences*;
- from this submenu select *3'UTR* and then click on *Header Information*;
- now from the header submenu select the following attributes: *Gene stable ID*, *Gene name*, *Transcript stable ID*, *3'UTR start* and *3'UTR end*;
- lastly click on *Results* from the top left and export the generated fasta file selecting the box *Unique results only* and pressing *Go*.

Note that a single gene identified with a unique ID may produce multiple RNA's transcripts each denoted with its own transcript ID. The actual transcript observed will depend on the tissue, developmental time point, and environmental or hormonal factor. Typically there's a single major transcript expressed in a given cell at a given time, but not always. Unfortunately in both the Diana TarBase and miR TarBase the transcript ID of the miRNA:mRNA pair is not present, hence, according to what is used in [40], we kept the transcript with the longest sequence.

mirBase

The mirBase database allowed us to retrieve all known miRNAs transcripts. For the homo sapiens species there currently around 2000 miRNAs sequences and in order to collect them we followed these steps:

- from the homepage click *Browse* from the top menu;
- click on *human* and a preview of the data will appear on screen as depicted in figure 9;
- on the bottom of the page select sequence type *Mature sequence* and output format *Unaligned fasta format*;
- press *Select all* and then *Fetch sequences*;
- the query result will be printed on the next page where it can be copied and pasted to a regular text file and saved as a fasta file.

As a preliminary step, the Diana and Mir TarBase data were parsed to remove inconsistent entries that were marked both as positive and negative targets due to contradictory results in different experimental validations and combine entries that were validated more than once by different verification methods. This produced a final dataset of 646206 positive (+) and 38464 negative (-) miRNA:mRNA

Homo sapiens miRNAs

ID	Accession	RPM	Chromosome	Start	End	Strand	Confidence
hsa-let-7a-1	MI0000060	145261	chr9	94175957	94176036	+	✓
hsa-let-7a-2	MI0000061	142652	chr11	122146522	122146593	-	✓
hsa-let-7a-3	MI0000062	142757	chr22	46112749	46112822	+	✓
hsa-let-7b	MI0000063	83403	chr22	46113686	46113768	+	✓
hsa-let-7c	MI0000064	135622	chr21	16539828	16539911	+	✓
hsa-let-7d	MI0000065	4649	chr9	94178834	94178920	+	✓

Figure 9: A sample of the mirBase database.

interactions containing bindings for about 16000 different genes and around 2000 miRNAs. This data was then split into two equal parts for the training and testing phases.

3.2.3 Training dataset

Structuring a proper training dataset is an essential aspect of effective deep learning models but one that is particularly hard to solve. Part of the challenge comes from the intrinsic relationship between a model and the corresponding training dataset. If the performance of a model is below expectations, it is often hard to determine whether the causes are related to the model itself or to the composition of the training dataset.

The purpose of this stage is to train and validate the neural network that has the responsibility for distinguishing between functional (positive) and non-functional (negative) target sites. A positive experimentally validated gene can exhibit tens of potential positive binding sites but not necessarily all them are actually functional. Hence, the training set must be composed of miRNA:MBS pairs rather than miRNA:mRNA duplexes. However, the retrieved data, while consistent and sufficiently various, do not contain such informations. Searching the Internet we were able to find only 2 publicly available validated datasets providing information regarding experimentally identified miRNA binding site locations: the Helwak [14] and the Grosswendt [12] datasets.

Positive binding sites

The two above datasets contain miRNA:MBS locations obtained through PAR-Clip [13] and CLASH [21] experiments, however the binding site identified might not be functional. Hence, in order to consider a site as positive we cross-reference them with the Diana TarBase and miR TarBase. In particular we considered a given duplex miRNA:MBS as functional if:

- form a stable bond, that is it has a free energy below a predetermined threshold according to Vienna Cofold. The threshold used was -10 kcal/mol.
- correspond to a miRNA:gene pair marked as functional in either the Diana or the miR dataset.

This process produced a total of 33142 positive miRNA:MBS duplexes for the training stage.

Negative binding sites

The small number of non-functional experimentally validated binding sites makes the construction of a representative negative dataset a very difficult task. Some of the examined existing tools overcome this issue by creating 'mock' targets [30] continuously shuffling the sequence of a real functional binding site until the resulting transcript does not appear in any positive miRNA target repository. This method, however, may lead the neural network to learn the function used to create the negative examples resulting in a model trained to discriminate between artificial and real data rather than discerning functional targets from non-functional. Besides, there is no guarantee that the generated sequence is indeed a true negative because it has not been experimentally validated.

The solution we implemented opted for using the validated miRNA:mRNA pairs to extract the negative binding sites and add them the ones provided by the CLASH and CLIP experiments. The idea is that any sequence of approximately 30 nucleotides in the mRNA of a negatively validated miRNA:mRNA pair may represent a potential negative target site. However, in order to avoid the introduction of noise including such sequences, we decided to keep only subsequences of mRNA where the associated miRNA has the potential to form a stable bond. In order to check this requirement we used a sliding window of 30 nt along the entire 3'UTR region. For each negative miRNA:mRNA pair we kept only MBS with the a binding energy below a certain threshold, chosen according to RNACoFold tool from the ViennaRNA package [15]. This process resulted in 32284 negative target sites.

Training stage

Once the positive and the negative datasets have been created, we shuffled and merged them to create a unique dataset. The whole process of creation of the training set is illustrated in figure 10. In order to train and validate the network we used an 8 fold cross-validation approach keeping 66% of the data for training and the rest for validation and test set. During this split we made sure to exclude miRNA:MBS pairs that shared miRNA and MBS with data instances in the training dataset.

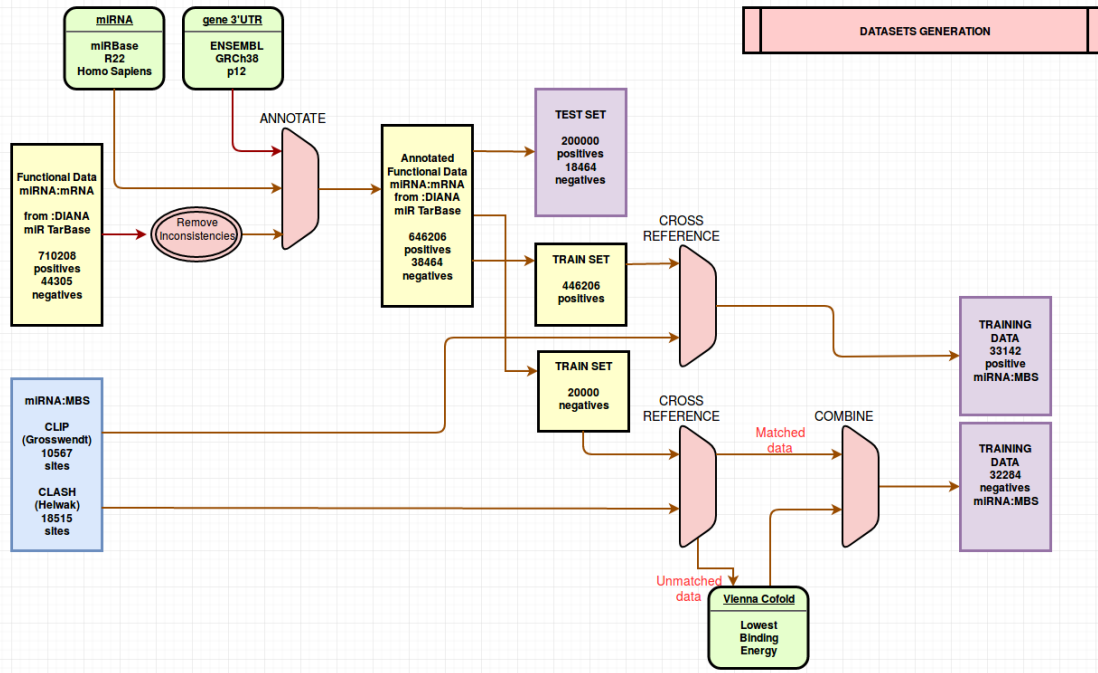


Figure 10: The datasets generation process.

3.3 The testing stage

There is a big difference regarding the procedure adopted for the testing stage compared to the training phase. Here the purpose is to predict if a given miRNA targets a certain gene, hence the testing data consists of pairs containing the miRNA and the whole 3'UTR transcript, rather than a specific MBS. This step is essential to be able to evaluate the whole pipeline process of DeepMiRNA.

To correctly evaluate the network performance we used the experimentally verified miRNA:mRNA pairs excluded from the training set. We now describe all the

steps involved in the test stage.

3.3.1 Candidate site selection methods: CSSM

Often, in the academic world, people tend to affirm that, in order to properly train a neural network, one needs to have a huge amount of data available. While we partially disagree with this common belief, convinced that quality and variety prevail over quantity, we still agree that in the case of miRNA targets prediction the available experimentally validated data is still not sufficiently representative of the task and the candidate site selection step effectively narrows the search space to simplify the NN classification task.

For this reason, the selection of candidate sites in a mRNA becomes a key step for a miRNA target sites predictor because it helps identifying which regions within the mRNA have the potential to accommodate a binding site. We found out that most of the publicly available algorithms follow a similar approach: they scan the gene's 3'UTR looking for sites that are partially complementary to the miRNA transcript; if a site meets certain criteria, it is considered to be a candidate site and is subjected to further analysis.

In the light of new recent results, stating that functional miRNA targets can arise either by a single strong binding site (i.e. 6 consecutive complementary nucleotides) or by multiple weak binding sites [14], we believe it's essential to consider the whole 3'UTR sequence using CSSM willing to accept both canonical and non-canonical sites. These methods are less conservative and allow accepting bulges, mismatches or wobble pairs in the seed region (see figure 11).

The CSSM adopted in DeepMiRNA use a very similar approach to miRAW [33]: we consider a 30-nucleotide window to scan the 3'UTR and we extend the typical 7mer seed region (figure 11a) considering the first 10 nucleotides of the miRNA transcript (figure 11d) to look for partial complementarity.

In particular, we consider a site to be a potential candidate site if there is a minimum number of base pairs, both Watson-Crick and Wobble, within the extended seed region. In this thesis we investigated three different configurations:

1. CSS-6.0:10: a candidate site must contain at least 6 base pairs between the extended seed region composed of the first 10 nucleotides;
2. CSS-7.0:10: a candidate site must contain at least 7 base pairs between the extended seed region composed of the first 10 nucleotides;
3. CSS-7.1:10: a candidate site must contain at least 7 base pairs between the extended seed region composed of the 9 nucleotides comprised from the second and the tenth.

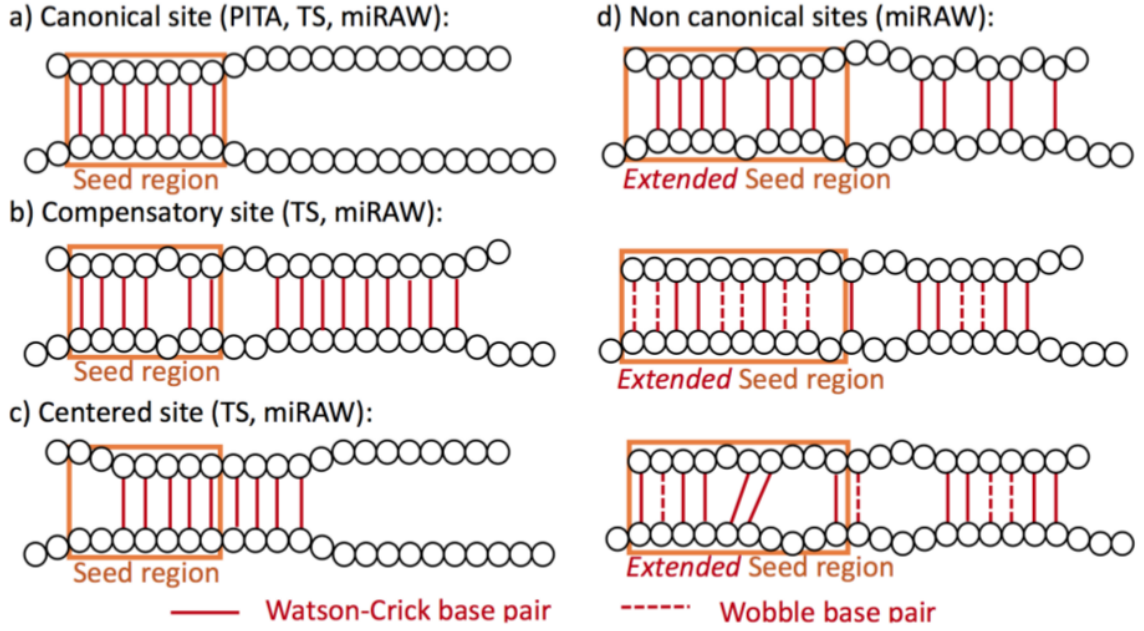


Figure 11: Example of canonical and non-canonical sites used in three available tools: Pita, TargetScan and miRAW.

In each case, base pairs do not need to be consecutive in order to accommodate the presence of gaps and bulges. The task of finding non-consecutive complementary nucleotides can be mapped to the problem of finding the longest common subsequence (LCS) between two strings as illustrated in figure 12.

Those configurations accept both standard canonical MBSs as well as a broader range of non-canonical target site structures including the vast majority of experimentally validated sites from Diana TarBase and CLIP/CLASH binding site datasets. Moreover, while these relaxed conditions for the seed region generate a much larger number of candidate sites and potentially an increased quantity of false positives, the decision of whether a site represents a functional target is delegated to the neural network. This way, we ensure that minimal assumptions, and hence bias, are incorporated into the analysis.

3.3.2 Longest Common Subsequence implementation

The computational cost of the testing stage is considerably higher than that of the training phase. The reasons lie mainly in the computation of the potential candidate sites, involving check of complementarity and calculation of the minimum free energy, and the cost of the filtering stage (see next section). While there is very

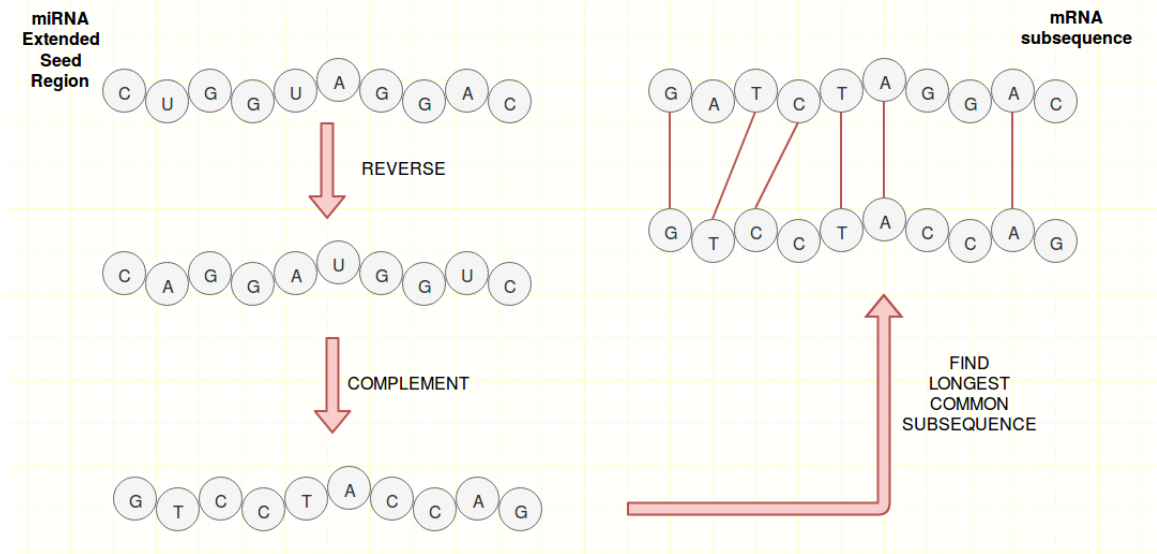


Figure 12: Mapping partial non-consecutive complementarity to longest common subsequence search.

little to do with the calculation of the interaction energy of the duplexes, which completely relies on the Vienna Cofold library, for the complementarity check we tried different approaches to reduce the computational cost.

As mentioned above the seek of non-consecutive complementary nucleotides between two sequences can be seen as a special case of the problem of finding the longest common subsequence (LCS) between two strings.

The LCS version we implemented in this thesis is done using *Cython*[5]. Cython is a programming language that can be seen as a superset of Python that aims to give C-like performances with code that is mostly written using native Python.

In fact, Cython actually works by producing Python modules where the code, however, is first translated and compiled into C.

For this reason, Cython code must, unlike Python, be compiled. This usually happens in two stages:

- a *.pyx* file is compiled by Cython into a *.c* file;
- the *.c* file is then compiled to a *.so* file which can be imported and used directly in any Python session.

The algorithm we implemented uses dynamic programming and has a worst case cost of $\mathcal{O}(n \times m)$ where n and m are the sequences lengths. The main difference between the classical implementation, that works by thoroughly filling a

$n \times m$ table and eventually uses backtracking to recover the final result, and our approach, consists in how we handle the table construction. In fact, unlike the original implementation, we update the table only when a match is found, thus reducing the whole computational cost.

However, for our task, we found out that the number of updates required is, in average, around the 70% of the table size. This is most likely due to the reduced alphabet size (four letters) of the sequences. For this reason the total cost of the function is comparable with the regular implementation.

In average the Cython version of the algorithm is 20 times faster than the original Python implementation and this justifies its use in the DeepMiRNA pipeline.

3.3.3 Filtering predictions

In chapter 2 we highlighted the importance of site accessibility for miRNA targets prediction. In fact, many studies [14] [32] have proved genes accommodate site accessibility by preferentially positioning targets in highly accessible regions [17] thus demonstrating that target accessibility is a critical factor in miRNA function.

Moreover, the use of a greedier CSSM approach gives rise to an increase on the number of potential candidate sites for each miRNA:mRNA duplex. For example, on average, where a strict canonical approach identifies ≈ 3 -4 sites per duplex, CSS-6.0:10 identifies approximately 32 MBSs while CSS-7.0:10 about 21 and CSS-7.1:10 around 15. As a consequence the chance of obtaining more false positives is very likely to increase.

Furthermore, the neural network itself may not be able to completely discern false from true positive duplexes. This is due, in particular, to two main factors: first of all the number of validated negative MBSs is extremely low; in fact, despite having a good number of negatively validates miRNA:mRNA pairs, we only have a small quantity of negatively validated binding sites. Thus, we had to artificially select the negative sites for the training stage as explained in the previous section. Second, functional and non-functional sites are very similar in terms of complementarity with the pairing miRNA and, hence, we believe it's important to also consider the characteristics of the secondary structure of the duplex to improve the accuracy of the prediction and reduce the number of false positives.

To this purpose, we investigated the possibility of using an a-posteriori filter that considers the secondary structure and compute the site accessibility of the target site. The site accessibility has been computed as follows:

- for each potential site identified using the CSSM, we considered the region of 200 nucleotides surrounding the MBS and we call it *folding chunk*. For example, if the MBS has length 30 and comprises the region between nucleotide

100 and 129, we consider for the computation of the site accessibility the region between nucleotide 15 and 214. If either on the left or on the right-hand side of the MBS there were not enough nucleotides we considered a shorter fold;

- the free energy ΔG_{free} of the folding region is computed, using Vienna Co-fold, to check the amount of energy released during the reaction;
- next we computed the opening energy ΔG_{open} , that is the energy needed to unfold the mRNA and allow the miRNA binding;
- the site accessibility energy $\Delta\Delta G$ is given by the difference between the free energy released with the binding and the opening energy (see figure 13). The bigger the value the more accessible the site:

$$\Delta\Delta G = \Delta G_{free} - \Delta G_{open} \quad (1)$$

In order to use this feature as a filter we set a site accessibility threshold meaning that sites with a low value are discarded while only MBS with a site accessibility greater than the threshold are considered as functional.

One may wonder why we used this feature as an a-posteriori filter rather than using it as another candidate selection rule. While the final result is the same, computing the secondary structure of a folding chunk, and hence its accessibility, is a pretty difficult task involving a lot more computation than asking the network for a prediction. Thus, we decided to use this feature as the last pipeline step.

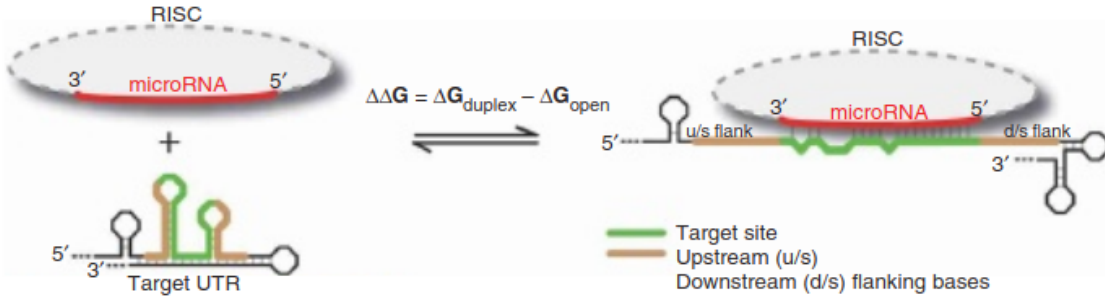


Figure 13: Illustration of site accessibility energy for miRNA:MBS interactions.

3.3.4 Obtaining the final classification

In order to classify a given duplex, we consider that the miRNA targets an mRNA if any of the potential MBSs of the mRNA are functional. The targeting process

implemented in DeepMiRNA requires the neural network to classify at least one not filtered candidate site as functional to consider a miRNA:mRNA as a positive targeting event.

More specifically, given a miRNA m and a gene g a candidate site selection method $sm(m, g)$ determines a set CS of potential MBSs:

$$sm(m, g) = CS \quad (2)$$

To determine if the the miRNA is targeting the gene, each candidate site within the miRNA:mRNA segment is properly encoded and input to the neural network. The result of the targeting prediction $T(m, g)$ corresponds to the disjunction of conjunction between the neural network outputs for all the candidate sites and the site accessibility check:

$$T(m, g) = \bigvee_{cs} nn(m, cs) \wedge \Delta\Delta G(cs) > th_{sa} \quad (3)$$

Where $nn(m, cs)$ denotes the output of the neural network while $\Delta\Delta G(cs) > th_{sa}$ represents the filtering step checking the site accessibility of the candidate is sufficiently high.

The fact that it only requires a single candidate to be predicted as functional makes DeepMiRNA particularly sensitive to false positive. This, however, it's partially mitigated by selecting a higher value for the site accessibility threshold th_{sa} . The best performance has been obtained by setting $th_{sa} = -14kcal/mol$.

Chapter 4

Neural network design and implementation

4.1 External libraries

DeepMiRNA has been developed using *Python3*. The *requirements.txt* file, present in the project root directory, contains names and versions of all libraries used for the implementation. In particular, we used *Pandas* and *Numpy* for the preprocessing step and the datasets preparation together with *RNA* (the Python version of the Vienna Cofold) and *Biopython* to parse .fasta files and convert them to .csv.

Implementation of the neural network was done with *Keras*[7] using *Tensorflow* backend[9]. Keras is an open-source neural-network library capable of running on top of TensorFlow, Theano and other major deep learning frameworks. Keras contains numerous implementations of commonly used NN building blocks such as layers, optimizers or activation functions and it aims at being user-friendly, modular and extensible. However, Keras, which is now fully supported in the TensorFlow's core library, has been conceived more as an interface than a standalone framework.

We opted for the use of this library because it offers a higher-level, more intuitive set of abstractions that makes it easier to develop deep learning models.

4.2 Choosing the right model

Building deep learning applications in the real world is a never-ending process of selecting and refining the right elements of a specific solution. Among those elements, the selection of the correct model and the right structure of the training dataset are, arguably, the two most important decisions that any data scientist

needs to make when designing deep learning solutions. How to decide what deep learning model to use for a specific problem? How do we know whether we are using the correct training dataset or we should gather more data? Those questions are the common denominator across all stages of the life cycle of a deep learning application. Even though there is no magic answer to those questions, there are several ideas that could guide the decision process.

First of all we need to start identifying the correct baseline model, in particular we should select what type of networks suits more the input dataset. In the case of miRNA targets predictions the topological structure of the available data are strictly correlated to the vectorization method selected. If we opt for the one-hot encoding that maps duplexes into fixed-size vectors, we should be thinking of using a feed-forward network with inter layer connectivity. While, if we select the Dna2Vec approach that transforms each duplex into a matrix, then the problem could be tackled using convolutional neural networks (CNN)[22].

The second part concerns the selection of the optimization algorithm to use. The most popular are, arguably, SGD (Stochastic Gradient Descent) and its variation using momentum or learning decay, and Adam. The latter, in particular, is very often used combined with CNNs.

As mentioned in chapter 3 DeepMiRNA uses 2 different neural network for the training stage according to the chosen data representation: a regular feed-forward network for the one-hot encoded sequences and a CNN for the Dna2Vec encoded duplexes.

4.3 The feed-forward network

The feed-forward network we designed for the miRNA target prediction task is a very simple neural network consisting of 5 dense hidden layers, comprising rectifier activation function ReLU nodes, each followed by a dropout layer. Dropout [37] is a regularization technique used to prevent overfitting. In this thesis, the dropout rate has been set to 0.7 meaning that there is a probability $p = 0.3$ that a certain neuron is ignored (i.e set to 0). Basically this implies that their contribution to the activation of downstream (i.e. next layer) neurons is temporally removed on the forward pass and weight update is not applied on the backward pass (see figure 14).

While a neural network learns, neuron weights settle into their context within the network. Those weights are tuned for specific features providing some specialization. However, neighboring neurons become to rely on this specialization, which, if taken too far, can result in a fragile model too specialized on the training data. Hence, if neurons are randomly dropped out of the network during training,

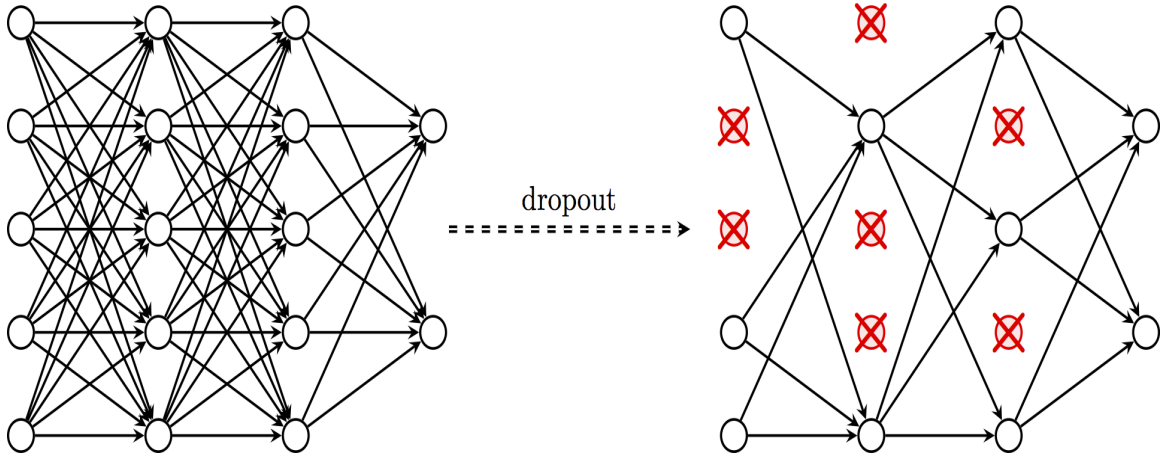


Figure 14: Left: neural network before dropout. Right: neural network after dropout.

other neurons will have to step in and handle the representation required to make predictions for the missing neurons. This is believed to result in multiple independent internal representations being learned by the network. The effect is that the network becomes less sensitive to the specific weights of neurons and results in a model capable of better generalization and less likely to overfit the training data.

The output layer is composed of one sigmoid node that returns a value between 0 and 1 corresponding to the final score prediction. The class of the site is determined by the output of this neuron:

$$class(i) = \begin{cases} 1 & \text{if } o \geq 0.4 \\ 0 & \text{if } o < 0.4 \end{cases}$$

Where i and o denote respectively the input and the output of the network. The prediction threshold value 0.4 has been empirically evaluated.

For the loss function we chose the binary cross entropy with Adam optimizer set with the following parameters:

- learning rate = 0.002;
- beta_1 = 0.9;
- beta_2 = 0.999;
- epsilon = $1e - 8$;
- decay = 0.0

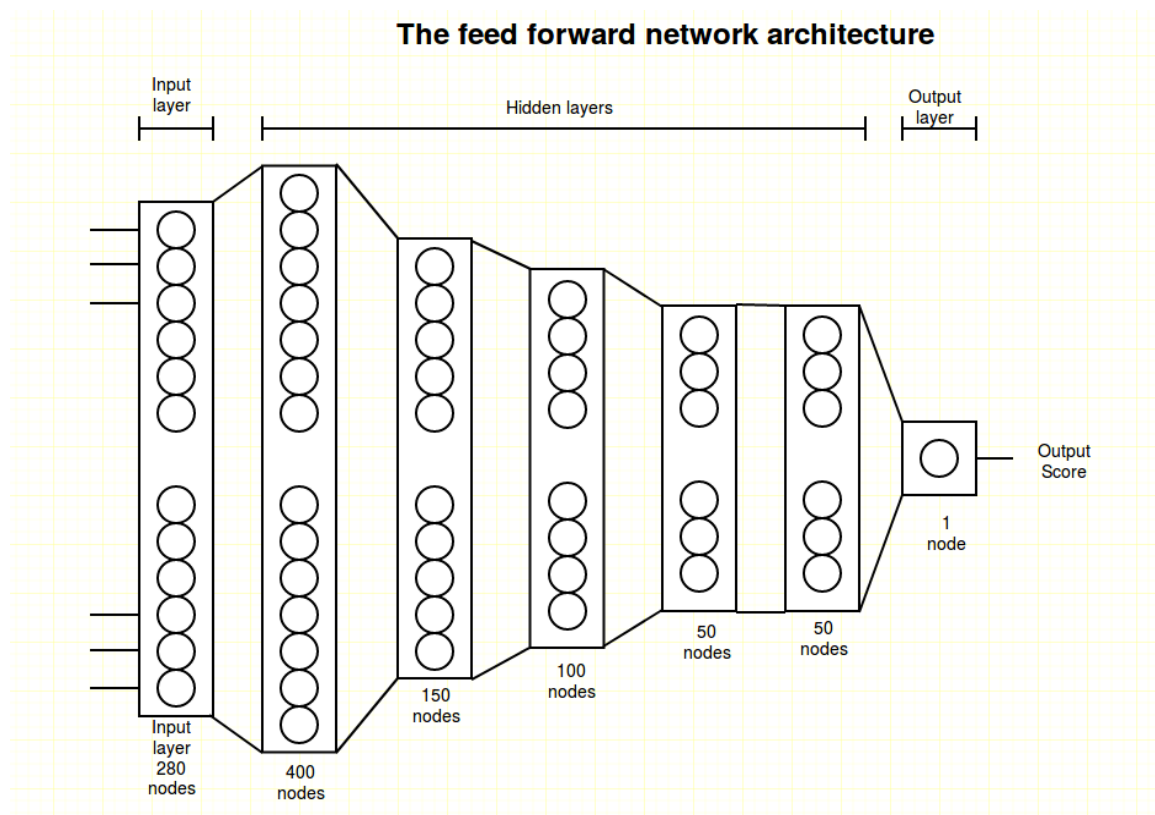


Figure 15: **The feed-forward neural network.** The first hidden layer increases the dimensionality of the input allowing the representation of data in a more complex feature space. There is debate in the machine learning community regarding the need of such over-completion layers, as they do not necessarily improve the accuracy whilst making the learning process slower. Considering that the relatively low number of inputs of the proposed network allows a fast training procedure, we opted to include this higher dimension layer to give the network the chance of identifying more complex patterns.

The resulting architecture can be visualized in figure15 and parameters such as number of layers and neurons have been chosen through cross validation.

Despite being a very simple model, this network showed a better performances on the test set compared to other deeper (that is with more hidden layers) or more complex (i.e with a greater number of nodes) models. Check chapter 5 for further details on the results.

4.4 The Convolutional Neural Network aka CNN

We have explained earlier, that the second encoding function provided by Deep-MiRNA is based on Dna2Vec [19]. In computational biology, one of the ubiquitous representation of long DNA sequence is dividing it into shorter k-mer components. Unfortunately, the straightforward vectorization of k-mer as a one-hot vector is vulnerable to the curse of dimensionality. Worse yet, the distance between any pair of one-hot vectors is equidistant, even though, for example, the sequence *ATGGC* should be closer to *ATGGG* than *CACGA*. This might be particularly problematic when applying the latest machine learning algorithms to solve problems in biological sequence analysis. Dna2Vec is, thus, a distributed representation of biological sequences that allows mapping k-mers to vectors of real numbers. By splitting the miRNA:MBS duplex in a fixed number of variable length k-mers, we were able to encode each pair into a bi-dimensional matrix.

More specifically, each k-mer is embedded into a continuous vector space of 100 dimensions, and each duplex is divided into 9 variable length k-mers: concatenating these vectors we obtain a 9x100 matrix representing the encoded sequence (see figure 16).

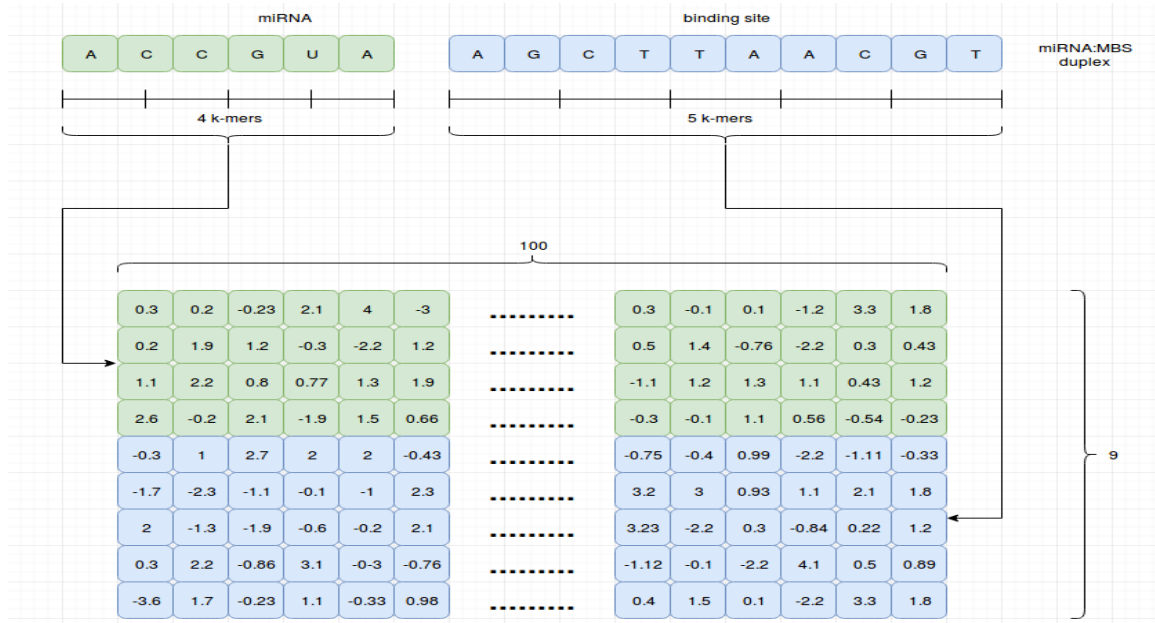


Figure 16: **Sequence Embedding.** In green the miRNA sequence is split into 4 k-mers each mapped in a 100-dimension vector. In blue the same is done with miRNA binding site which is divided into 5 k-mers. The concatenation of the 9 resulting vectors is the matrix distributed representation of the duplex.

With this representation, we believe the best base model to use for our task is represented by a convolutional neural network.

4.4.1 CNN architecture

Chapter 5

Experimental results

5.1 Introduction

In the previous chapters we described how the DeepMiRNA's architecture is based on two main functional blocks: on one side there is the NN, whose purpose is that of analyzing the selected candidate target sites, while on the other there are the CSSM used during the target prediction step and the a-posteriori filter that tries to refine the network prediction.

To assess these two aspects, we first evaluated the outputs of the NN training process through cross validation and then investigated performance using the different candidate site selection methods outlined in chapter 3. These comprised the novel (non-canonical) models implemented for this thesis: CSS-6.10 and CSS-7.10. Finally, we tested DeepMiRNA's performance by comparing it against TargetScan[3], mirDB[26] and miRAW[33], which represent the most commonly used target site predictors based on citations.

5.2 Neural Network evaluation

The best performing feed-forward neural network was trained using the whole training set with this parameters:

- dropout rate = 0.7
- optimizer = Adam as described in chapter 4
- loss function = binary cross entropy
- batch size = 128

- number of epochs = 15

The above parameters were computed using a validation set composed of about 10000 examples (20% of the training set).

Once the best model has been selected, we built a fresh network using the best parameters and we trained it holding out the 25% of the training data for the purpose of testing its performance. This validation presented very good results in terms of predicting both positive and negative sites, with all evaluated metrics resulting in scores well above 0.9 (see figure17).

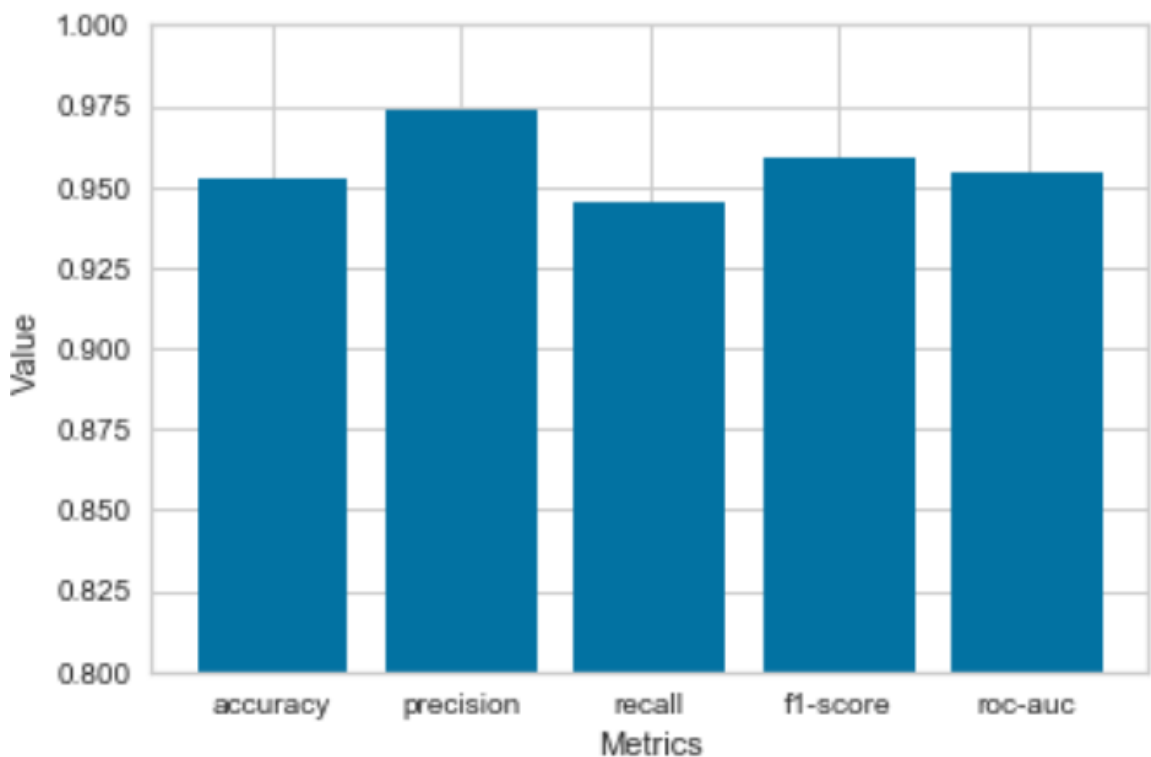


Figure 17: **Network performance metrics.** To this purpose we trained the network with best parameters using 75% of the training data. The remaining 25% has been used as test set. As shown above all metrics are close to 0.95.

5.3 The role of the candidate site selection method

To investigate the impact of the site selection method, we compared the performances of the three different CSSM described in chapter3.

Figure 18 summarizes the obtained results: all methods reach accuracies between 0.76 and 0.8, these values are computed considering the imbalance of the test set, that is they are calculated as the average of the accuracies on positive and negative samples. CSS-6.0:10 seems to achieve slightly better performances for every metric compared to the other two methods. This is the less conservative approach and represents the best trade-off between the higher quantity of false positives generated by the CSSM and the increased rate of false negatives due to the filtering stage. The F1-score is the harmonic average between precision and recall and shows how well both classes are classified by a particular CSSM. The values obtained are very similar to the accuracies and indicate an ability to effectively predict both negative and positive targets especially for CSS-6.0:10 and CSS-7.0:10.

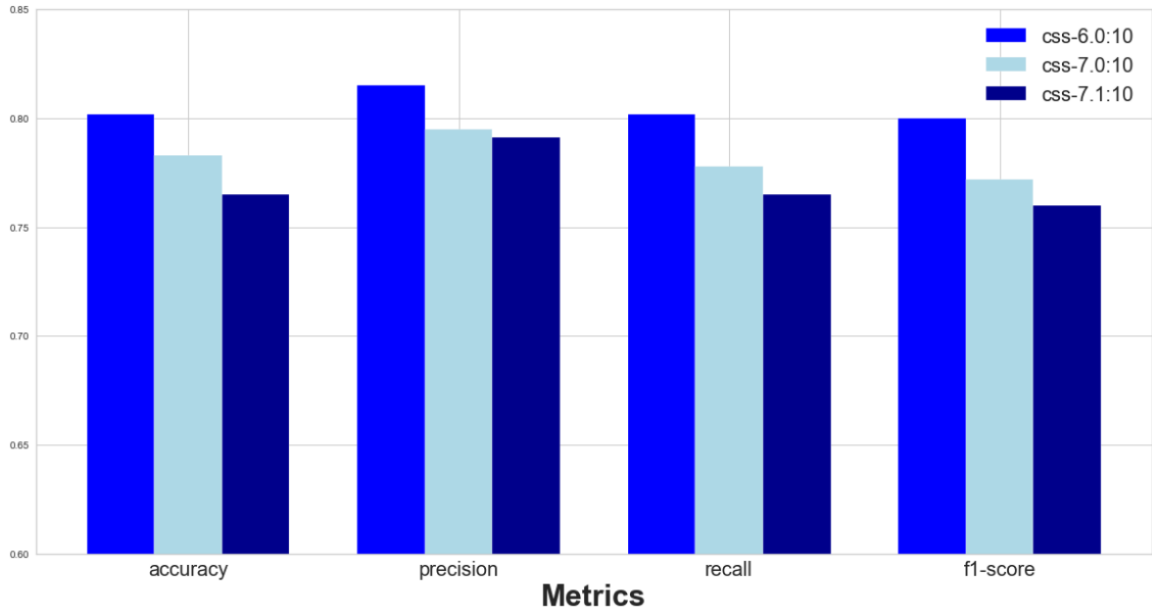


Figure 18: **Evaluation of DeepMiRNA different candidate site selection methods.** Results are evaluated in terms of balanced accuracy, precision, recall and f1-score. The best result was achieved using CSS-6.0:10.

5.4 The importance of site accessibility filtering

In order to measure the effect of site accessibility on miRNA target prediction, we tested the best performing pipeline configuration without filtering the network output. While this considerably reduce the computational cost of the whole process, DeepMiRNA becomes more biased towards the prediction of positive sites

Table 4: **Comparison results with or without the a-posteriori filter**

Accuracies		Precision		Recall		F1-score	
NF	WF	NF	WF	NF	WF	NF	WF

which resulted in high recall but low precision. The obtained results are compared in Table 4.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

```
\hypersetup{urlcolor=red}, or  
\hypersetup{citecolor=green}, or  
\hypersetup{allcolor=blue}.
```

If you want to completely hide the links, you can use:

```
\hypersetup{allcolors=.}, or even better:  
\hypersetup{hidelinks}.
```

If you want to have obvious links in the PDF but not the printed text, use:

```
\hypersetup{colorlinks=false}.
```

Bibliography

- [1] Onehot encoding in python. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>. Accessed: 2019-06-15.
- [2] AGARWAL, V., BELL, G. W., NAM, J.-W., AND BARTEL, D. P. Predicting effective microrna target sites in mammalian mrnas. *elife* 4 (2015), e05005.
- [3] AGARWAL, V., BELL, G. W., NAM, J.-W., AND BARTEL, D. P. Predicting effective microrna target sites in mammalian mrnas. *elife* 4 (2015), e05005.
- [4] BARTHEL, D. ibiology: Introduction to mirnas. <https://www.ibiology.org/genetics-and-gene-regulation/introduction-to-micrnas/>. Accessed: 2019-06-11.
- [5] BEHNEL, S., BRADSHAW, R., CITRO, C., DALCIN, L., SELJEBOTN, D., AND SMITH, K. Cython: The best of both worlds. *Computing in Science Engineering* 13, 2 (2011), 31–39.
- [6] CHENG, S., GUO, M., WANG, C., LIU, X., LIU, Y., AND WU, X. Mirtidl: a deep learning approach for mirna target prediction. *IEEE/ACM transactions on computational biology and bioinformatics* 13, 6 (2015), 1161–1169.
- [7] CHOLLET, F., ET AL. Keras. <https://keras.io>, 2015.
- [8] DU, T., AND ZAMORE, P. D. microprimer: the biogenesis and function of microrna. *Development* 132, 21 (2005), 4645–4652.
- [9] ET AL., M. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [10] FROMM, B., BILLIPP, T., PECK, L. E., JOHANSEN, M., TARVER, J. E., KING, B. L., NEWCOMB, J. M., SEMPERE, L. F., FLATMARK, K., HOVIG, E., ET AL. A uniform system for the annotation of vertebrate microrna genes and the

- evolution of the human microRNAome. *Annual review of genetics* 49 (2015), 213–242.
- [11] GOLDBERG, Y., AND LEVY, O. word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
- [12] GROSSWENDT, S., FILIPCHYK, A., MANZANO, M., AND KLIRONOMOS. Unambiguous identification of miRNA: target site interactions by different types of ligation reactions. *Molecular cell* 54, 6 (2014), 1042–1054.
- [13] HAFNER, M., LANDTHALER, M., BURGER, L., KHORSHID, M., HAUSSE, J., BERNINGER, P., ROTHBALLER, A., ASCANO JR, M., JUNGKAMP, A.-C., MUNSCHAUER, M., ET AL. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by par-clip. *Cell* 141, 1 (2010), 129–141.
- [14] HELWAK, A., KUDLA, G., DUDNAKOVA, T., AND TOLLERVEY, D. Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding. *Cell* 153, 3 (2013), 654–665.
- [15] HOFACKER, I. L. Vienna RNA secondary structure server. *Nucleic acids research* 31, 13 (2003), 3429–3431.
- [16] HSU, S.-D., LIN, F.-M., WU, W.-Y., LIANG, C., HUANG, W.-C., CHAN, W.-L., TSAI, W.-T., CHEN, G.-Z., LEE, C.-J., CHIU, C.-M., ET AL. miRTarBase: a database curates experimentally validated microRNA–target interactions. *Nucleic acids research* 39, suppl_1 (2010), D163–D169.
- [17] KERTESZ, M., IOVINO, N., UNNERSTALL, U., GAUL, U., AND SEGAL, E. The role of site accessibility in microRNA target recognition. *Nature genetics* 39, 10 (2007), 1278.
- [18] KIM, D., SUNG, Y. M., PARK, J., KIM, S., KIM, J., PARK, J., HA, H., BAE, J. Y., KIM, S., AND BAEK, D. General rules for functional microRNA targeting. *Nature genetics* 48, 12 (2016), 1517.
- [19] KIMOTHI, D., SONI, A., BIYANI, P., AND HOGAN, J. M. Distributed representations for biological sequence analysis. *arXiv preprint arXiv:1608.05949* (2016).
- [20] KOTSIANTIS, S., KANELLOPOULOS, D., PINTELAS, P., ET AL. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering* 30, 1 (2006), 25–36.

- [21] KUDLA, G., GRANNEMAN, S., HAHN, D., BEGGS, J. D., AND TOLLERVEY, D. Cross-linking, ligation, and sequencing of hybrids reveals rna–rna interactions in yeast. *Proceedings of the National Academy of Sciences* 108, 24 (2011), 10010–10015.
- [22] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature* 521, 7553 (2015), 436.
- [23] LEE, B., BAEK, J., PARK, S., AND YOON, S. deeptarget: end-to-end learning framework for microrna target prediction using deep recurrent neural networks. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (2016), ACM, pp. 434–442.
- [24] LEWIS, B. P., BURGE, C. B., AND BARTEL, D. P. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets. *cell* 120, 1 (2005), 15–20.
- [25] LIM, L. P., LAU, N. C., GARRETT-ENGELE, P., GRIMSON, A., SCHELTER, J. M., CASTLE, J., BARTEL, D. P., LINSLEY, P. S., AND JOHNSON, J. M. Microarray analysis shows that some micrnas downregulate large numbers of target mrnas. *Nature* 433, 7027 (2005), 769.
- [26] LIU, W., AND WANG, X. Prediction of functional microrna targets by integrative modeling of microrna binding and target expression data. *Genome biology* 20, 1 (2019), 18.
- [27] LU, M., ZHANG, Q., DENG, M., MIAO, J., GUO, Y., GAO, W., AND CUI, Q. An analysis of human microrna and disease associations. *PloS one* 3, 10 (2008), e3420.
- [28] LYTLE, J. R., YARIO, T. A., AND STEITZ, J. A. Target mrnas are repressed as efficiently by microrna-binding sites in the 5′utr as in the 3′utr. *Proceedings of the National Academy of Sciences* 104, 23 (2007), 9667–9672.
- [29] MAZIERE, P., AND ENRIGHT, A. J. Prediction of microrna targets. *Drug discovery today* 12, 11-12 (2007), 452–458.
- [30] MENOR, M., CHING, T., ZHU, X., GARMIRE, D., AND GARMIRE, L. X. mir-mark: a site-level and utr-level classifier for mirna target prediction. *Genome biology* 15, 10 (2014), 500.
- [31] PASQUINELLI, A. E. Micrnas and their targets: recognition, regulation and an emerging reciprocal relationship. *Nature Reviews Genetics* 13, 4 (2012), 271.

- [32] PETERSON, S. M., THOMPSON, J. A., UFKIN, M. L., SATHYANARAYANA, P., LIAW, L., AND CONGDON, C. B. Common features of microRNA target prediction tools. *Frontiers in genetics* 5 (2014), 23.
- [33] PLA, A., ZHONG, X., AND RAYNER, S. miraw: A deep learning-based approach to predict microRNA targets by analyzing whole microRNA transcripts. *PLoS computational biology* 14, 7 (2018), e1006185.
- [34] RECZKO, M., MARAGKAKIS, M., ALEXIOU, P., GROSSE, I., AND HATZIGEORGIOU, A. G. Functional microRNA targets in protein coding sequences. *Bioinformatics* 28, 6 (2012), 771–776.
- [35] SCHIRLE, N. T., SHEU-GRUTTADAURIA, J., AND MACRAE, I. J. Structural basis for microRNA targeting. *Science* 346, 6209 (2014), 608–613.
- [36] SMITH, S. M., AND MURRAY, D. W. An overview of microRNA methods: expression profiling and target identification. In *Molecular Profiling*. Springer, 2012, pp. 119–138.
- [37] SRIVASTAVA, N., HINTON, G., AND KRIZHEVSKY, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [38] VLACHOS, I. S., PARASKEVOPOULOU, M. D., KARAGKOUNI, D., GEORGAKILAS, G., VERGOULIS, T., KANELLOS, I., ANASTASOPOULOS, I.-L., MANIOU, S., KARATHANOU, K., KALFAKAKOU, D., ET AL. Diana-tarbase v7. 0: indexing more than half a million experimentally supported mirna: mrna interactions. *Nucleic acids research* 43, D1 (2014), D153–D159.
- [39] WATANABE, Y., TOMITA, M., AND KANAI, A. Computational methods for microRNA target prediction. *Methods in enzymology* 427 (2007), 65–86.
- [40] WEN, M., CONG, P., ZHANG, Z., LU, H., AND LI, T. Deepmirtar: a deep-learning approach for predicting human mirna targets. *Bioinformatics* 34, 22 (2018), 3781–3787.
- [41] YATES, A., AKANNI, W., AMODE, M. R., BARRELL, ET AL. Ensembl 2016. *Nucleic acids research* 44, D1 (2015), D710–D716.
- [42] YUE, D., LIU, H., AND HUANG, Y. Survey of computational algorithms for microRNA target prediction. *Current genomics* 10, 7 (2009), 478–492.