Fall 2010

A MicroRNA TARGET PREDICTION ALGORITHM

Rupinder Singh
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the [Bioinformatics Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Singh, Rupinder, "A MicroRNA TARGET PREDICTION ALGORITHM" (2010). *Master's Projects*. 15.
DOI: <https://doi.org/10.31979/etd.kw6x-sebw>
https://scholarworks.sjsu.edu/etd_projects/15

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

A MicroRNA TARGET PREDICTION ALGORITHM

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Rupinder Singh

Dec 2010

©2010

Rupinder Singh

ALL RIGHTS RESERVED

The Undersigned Writing Project Committee Approves the Writing Project Titled

A MicroRNA TARGET PREDICTION ALGORITHM

by

Rupinder Singh

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

Dr. Sami Khuri
Professor of Computer Science Department,
San Jose State University, San Jose, California

Date

Dr. Chris Pollett
Professor of Computer Science Department,
San Jose State University, San Jose, California

Date

Dr. Robert Chun
Professor of Computer Science Department,
San Jose State University, San Jose, California

Date

ABSTRACT

A MicroRNA TARGET PREDICTION ALGORITHM

by Rupinder Singh

MicroRNA target prediction using the experimental methods is a challenging task. To accelerate the process of miRNA target validation, many computational methods are used. Computational methods yield many potential candidates for experimental validation. This project is about developing a new computational method using dynamic programming to predict miRNA targets with more accuracy. The project discusses the currently available computational methods and develops a new algorithm using the currently available knowledge about miRNA interactions.

ACKNOWLEDGMENT

I would like to thank my project advisor, Dr. Sami Khuri for his exceptional guidance and support. I would also like to thank Prof. Chris Pollett and Prof. Robert Chun for their valuable time and feedback.

TABLE OF CONTENTS

1 Introduction	10
2 Background	13
2.1 MicroRNA Biogenesis.....	13
2.2 MicroRNA Target prediction	16
2.2.1 Complementarity searching based methods	18
2.2.2 Thermodynamic based methods	18
2.2.3 Other methods.....	19
2.3 Current MicroRNA target prediction tools	19
2.3.1 miRanda	19
2.3.2 TargetScan and TargetSacnS	20
2.3.3 PicTar	20
2.3.4 DIANA-microT	21
2.3.5 Support vector machine (SVM).....	21
2.4 MicroRNA Target databases	21
2.4.1 Tarbase.....	21
2.4.2 MiRDB	22
2.4.3 MiRecords	22
2.5 Prior work.....	23
3 Method.....	24
3.1 Data collection	24
3.2 miRNA and 3'UTR alignment	25
3.3 Position Specific Scoring Matrix.....	27
3.4 Target Prediction	28
4 Implementation.....	30
4.1 Script 1: MicroRNA-mRNA interaction information.....	30
4.2 Script 2: Retrieve sequences from mirBase and GenBank.....	31
4.3 Script 3: Align miRNA and 3'UTR	32

4.4 Script 4: Filtering with minimum free energy	36
4.5 Script 5: Build Position Specific Scoring Matrix (PSSM)	37
4.6 Script 6: Predicting miRNA targets	38
5 Results	40
6 Conclusion	46
Bibliography	47
Appendices	49
Appendix A: Source code for miRNA target predictor	49

LIST OF FIGURES

Figure 1. MicroRNA biogenesis in the animal cells.....	14
Figure 2. Various configurations for miRNA-mRNA duplex.....	15
Figure 3. Secondary structure of miRNA-mRNA interaction.	17
Figure 4. Overview of developed method.....	29
Figure 5. Contents of main.txt file	31
Figure 6. MicroRNA and mRNA output files.....	32
Figure 7. Scoring matrix.....	33
Figure 8. Backtracking	34
Figure 9. Output file for local alignments	35
Figure 10. Output file for intractions with free energy	37
Figure 11. MicroRNA specific PSSM	38
Figure 12. Output file for predicted interactions	39

LIST OF TABLES

Table 1. Basic scoring scheme	25
Table 2. Characters for consensus sequence.....	15
Table 3. M. Zuker's energy parameters.	36
Table 4. Comparison of MIMAT0000068 (hsa-miR-15a) interactions.....	40
Table 5. Comparison of MIMAT0002867 (hsa-miR-520h) interactions.....	42
Table 6. Comparison of MIMAT0000255 (hsa-miR-34a) interactions.....	43

1 Introduction

MicroRNAs (miRNAs) are ~22 nucleotide RNA sequences that bind to multiple target mRNAs, resulting in their silencing. MicroRNAs targets ~60% of all genes and are abundantly present in all human cells. MicroRNAs play an important role in the development of diseases like cancer. The function of miRNA depends on the gene it targets. Knowledge of miRNA function and its target is the first step towards the development of drug or therapy to cure the diseases. There are many experimentally validated miRNAs, but it is a difficult task to identify new miRNA targets experimentally. Since each miRNA can target several mRNA sequences, this will require the aid of computational target prediction tools. These tools can provide numerous targets for experimental validation.

It is known from experimentally validated miRNAs and their targets that miRNAs and their targets have high complementarity between them. Using this as a building block, many algorithms have been developed to predict miRNA targets. Some common approaches are based on dynamic programming, machine learning algorithms, Hidden Markov Model (HMM) and the free energy of the miRNA-target duplex. Results from these algorithms yield many false-positive target sequences (Yoon & Micheli, 2006; Maziere & Enright, 2007; Zhang, Pan, Wang, Cobb, & Anderson, 2006). The goal of this project is to develop an algorithm, using dynamic programming,

with a position specific scoring scheme that can predict the miRNA targets with lower false-positive rates.

Due to its simplicity dynamic programming approaches to predict miRNA targets are particularly attractive. One paper to consider this approach is “A combined computational-experimental approach predicts human microRNA targets” (Kiriakidou, et al., 2004). DIANA-microT (Kiriakidou, et al., 2004) target prediction uses the simple scoring scheme to align miRNA and 3'UTR sequences. This scoring scheme scores interactions at each position equally. It is known from experimentally validated miRNA-target interactions that complementarity at some positions is more important than others. To overcome this limitation, in this project, we develop a dynamic programming algorithm which uses a Position Specific Scoring Matrix (PSSM) as its scoring scheme. This approach assigns high scores to important positions in miRNA-target interactions and low scores to unimportant positions in miRNA-target interactions. This in turn helps to reduce the number of false-positives target sequences returned by our algorithm.

Section 2 of this report discusses the biogenesis of miRNA and their targets, computational approaches to miRNA target prediction and currently available miRNA target prediction tools. Section 3 discusses the design of the developed method. Section 4 discusses the implementation of the developed method. Results from the developed algorithm are compared with the other

miRNA targets prediction tools in Section 5. Section 6 concludes the report and identifies future directions.

2 Background

2.1 MicroRNA Biogenesis

In 1993, Victor Ambros, Rosalind Lee and Rhonda Feinbaum discovered that in *C.elegans* lin-4 regulates the translation of lin-14 mRNA by producing a small RNA product called MicroRNA (miRNA) (Lee, Feinbaum, & Ambros, 1993). They also noticed that these miRNAs, produced from lin-4, had antisense complementarity to multiple sites in the 3'UTR of the lin-14 gene. MicroRNAs are ~22 nucleotides long, non-coding RNAs, that bind to the 3'UTR region of mRNA to inhibit transcription or to induce cleavage (Friedman, Farh, Burge, & Bartel, 2009; Lee, Jeon, Lee, Kim, & Kim, 2002). MicroRNAs are transcribed by polymerases II and polymerases III (Bartel, 2004). The majority of miRNAs are transcribed from independent transcription units, but some are also transcribed from introns of pre-mRNAs (Bartel, 2004). Some miRNA genes are found in a cluster of 2-7 genes, separated by a few nucleotides, having a highly similar expression profile suggesting that transcription of these miRNAs is controlled by common promoters (Lee, Jeon, Lee, Kim, & Kim, 2002). MicroRNA genes found between the introns are not transcribed by their own promoter, but they are processed from the introns (Bartel, 2004). Some miRNAs are expressed at different stages of development; some are expressed in different cells (Bartel, 2004).

Figure 1. MicroRNA biogenesis in the animal cells

The biogenesis of miRNAs is shown in Figure 1 (McDanel, 2009). The microRNA gene is transcribed into primary miRNA (pri-miRNA) by polymerases, which is then processed by a nuclear RNase type III enzyme, Drosha, to produce a 60-70nt long stem loop precursor miRNA (pre-miRNA) (Nam, Shin, Han, Lee, Kim, & Zhang, 2005). Drosha cleaves both strands of

the stem near the base of primary stem loop (Bartel, 2004). The precursor is then exported to the cytoplasm by the nuclear export factor Exportin 5 and the Ran-GTP cofactor and trimmed by Dicer into miRNA:miRNA* duplex (Nam, Shin, Han, Lee, Kim, & Zhang, 2005; Lindow & Gorodkin, 2007). Drosha processes one end of miRNA:miRNA* duplex in nucleus while other end is processed by Dicer in cytoplasm (Bartel, 2004). One strand of miRNA:miRNA* duplex is identified by the RNA-induced silencing complex (RISC) while the other strand is generally degraded (McDanel, 2009). The miRNA, along with RISC, targets the specific 3'UTR of mRNA transcript.

Figure 2. Various configurations for miRNA-mRNA duplex (A) Near-perfect binding site for one miRNA (B) Multiple target sites for one miRNA (C) Strong binding site for one miRNA (D) Multiple target sites for multiple miRNAs

The miRNA binds to the mRNA and it either causes the mRNA cleavage or it inhibits the translation. mRNA cleavage mostly occurs in plants

while translational repression occurs mostly in animals (Yoon & Micheli, 2006). Single mRNA can contain multiple miRNA targets for different miRNAs or for the same miRNA (Figure 2) (Yoon & Micheli, 2006).

It is also known that miRNAs are highly conserved among different species. In addition to the conserved miRNAs, there are lots of non-conserved species specific miRNAs (Zhang, Pan, Wang, Cobb, & Anderson, 2006). These non-conserved miRNAs may control the specific characteristics that are unique to those species.

2.2 MicroRNA Target prediction

Prediction of miRNA targets in plants is ingenious because of near perfect complementarity between miRNA and mRNA, but is more challenging in animals. Interaction between miRNA and mRNA lacks the perfect complementarity, which results in many different computational approaches to predict miRNA targets. The miRNA target prediction principles used by most of the approaches are almost similar (Yoon & Micheli, 2006). Some of the prediction criteria are:

- The miRNA and 3'UTR region of mRNA have complementarity between them, especially between the seed region, position 2-7 of miRNA, of miRNA and mRNA. Complementarity between miRNA and mRNA can be

of three types: 5'-dominant canonical, 5'-dominant seed and 3'-compensatory (Figure 3) (Maziere & Enright, 2007).

- The thermodynamics of miRNA and mRNA interaction can be computed by currently available RNA folding packages and is used in many prediction algorithms (Yoon & Micheli, 2006).
- The 3'UTR target regions of many miRNAs are highly conserved over many species (Friedman, Farh, Burge, & Bartel, 2009).

Figure 3. Secondary structure of miRNA-mRNA interaction. (a) Good or perfect complementarity at both the 5' and 3' ends of the miRNA. (b) Perfect seed region complementarity at 5' end of the miRNA, but poor 3' complementarity. (c) Seed region has a mismatch or wobble but 3' end has excellent complementarity.

MicoRNA target prediction approaches can be classified into three categories.

1. Complementarity searching based methods.
2. Thermodynamic based methods.
3. Other methods.

2.2.1 Complementarity searching based methods

These types of approaches try to identify initial potential targets using complementarity and then improve them by using other features like thermodynamics, binding site structure and conservation. Stark and co-workers initially implemented this strategy for predicting miRNA targets in *Drosophila melanogaster* (Stark, Brennecke, Russell, & Cohen, 2003). miRanda, TargetScan and PicTar are based on the same strategy but address the limitation of the previous approach.

2.2.2 Thermodynamic based methods

Algorithms in this category use the favorable thermodynamic structure as an initial indicator and use other properties of miRNA-mRNA interaction to filter miRNA targets. DIANA-microT and RNAHybrid fall in this category.

2.2.3 Other methods

Large-scale analysis published by Xie et al. suggested that 3'UTRs have some common motifs of short length (Xie, et al., 2005). Some of these motifs are complement to the seed region of known miRNA, others might be the target of unknown miRNAs. This method of analyzing whole genome to predict miRNA targets can only be applied to predict conserved targets. Machine learning methods are also used to predict miRNA targets. Kim et al. (Kim, Nam, Rhee, Lee, & Zhang, 2006) implemented the miTarget classifier using the support vector machine (SVM). Performance of machine learning algorithms is affected by shortage of training data.

2.3 Current MicroRNA target prediction tools

2.3.1 miRanda

miRanda (Enright, John, Gaul, Tuschl, Sander, & Marks, 2003) identifies potential binding sites by looking for high complementarity between miRNA and mRNA. The scoring method used by the algorithm favors complementarity between 5' end of the miRNA and 3' end of mRNA. The resulting binding sites are then evaluated thermodynamically, using the Vienna RNA folding package. The false-positive rate is between 24% and 39% with the basic parameter settings.

2.3.2 TargetScan and TargetScanS

This method (Lewis, Shih, Jones-Rhoades, Bartel, & Burge, 2003) requires perfect complementarity to the seed region of a miRNA. After checking complementarity in the seed region, it then checks the complementarity in other regions. It tries to filter many false-positives in the beginning of the prediction process. It also uses conservation criteria for filtering. Predicted binding sites are then evaluated thermodynamically, using the Vienna RNA folding package. The estimated false-positive rate is between 22% and 31%. TargetScanS is the improved version of TargetScan. It limits the miRNA complementarity to a six nucleotide seed and a match at position 1 of 3' end.

2.3.3 PicTar

This algorithm (Krek, et al., 2005) uses a group of orthologous 3'UTRs from multiple species and then scans it for those displaying seed match to miRNAs. Matched alignments are then filtered according to their thermodynamic stability. Predicted targets are then scored using Hidden Markov Model (HMM) maximum-likelihood fit approach.

2.3.4 DIANA-microT

This method (Kiriakidou, et al., 2004) uses a 38 nucleotide window that progressively moves across a 3'UTR sequence. The free energy of the potential binding sites is calculated. It demands 3' complementarity to the miRNA instead of 5' end complementarity.

2.3.5 Support vector machine (SVM)

SVM (Kim, Nam, Rhee, Lee, & Zhang, 2006) is based on 41 features, based on interaction between miRNA and mRNA. These features can be classified into three elements: structural features, thermodynamic features and position based features. This method builds a statistical model and tries to predict miRNA targets by fitting miRNA and their targets in the model. This method requires large negative training data, which is currently unavailable for miRNA-mRNA interaction.

2.4 MicroRNA Target databases

2.4.1 Tarbase

TarBase (Sethupathy, Corda, & Hatzigeorgiou, 2006) database contains a manually curated collection of experimentally tested miRNA targets, in humans/mice, fruit flies, worms, and zebrafish. It contains positive

tested and negative tested miRNA targets. Each positive target site has information about miRNA and its gene, the nature of experiment and source from where data was extracted. It is also linked with other databases such as Gene Ontology (GO) and UCSC Genome Browser. Currently it has 1333 entries for all species.

2.4.2 MiRDB

MiRDB (Wang, 2008) is an online database system based on miRNA target prediction and functional annotation. Compared to other databases miRNA functional annotations in miRDB are presented with a primary focus on mature miRNAs. It also has wiki editing interface that allows anyone to make contributions on miRNA functional annotation. It has 2295 miRNA with targets.

2.4.3 MiRecords

miRecords (Xiao, Zuo, Cai, Kang, Gao, & Li, 2009) is a database of experimentally validated and computationally predicted miRNA targets. The validated targets component of the miRecords contains high-quality manually curated experimentally validated miRNA-target interactions with systematic documentation of experimental support for each interaction. All experimental evidences about miRNA-targets are divided into three levels: the target gene

level, the target region level and the target site level. The predicted targets component of miRecords store predicted miRNA targets produced by 11 miRNA target prediction programs. Currently, this database includes 1135 records of validated miRNA-target interactions between 301 miRNAs and 902 target genes.

2.5 Prior work

DIANA-microT miRNA target prediction tool is based on the dynamic programming. It uses the dynamic programming to predict the potential binding sites and calculates the free energy at each step. The difference between our approach and the DIANA-microT is the scoring scheme. Our approach uses the Position specific scoring matrices as compared to the simple scoring scheme of DIANA-microT.

3 Method

This section illustrates the developed algorithm used to predict the targets of miRNA. It is divided into four parts and each part is listed individually.

3.1 Data collection

We used the miRecords (Xiao, Zuo, Cai, Kang, Gao, & Li, 2009) database to collect the candidate data. This database contains both, experimentally validated and computationally predicted miRNA-mRNA interactions. We used only experimentally validated miRNA-mRNA interactions to improve the quality of miRNA target prediction. This database does not contain miRNA and mRNA sequences but only contains the reference to mirBase and GenBank database. MiRNA sequences are downloaded from mirBase database in fasta format. GenBank provides web based API called 'Entrez Programming Utilities', which is used to retrieve DNA sequences and information of coding region. These DNA sequences are then processed and 3'UTR regions are extracted from the sequences. Some of the DNA sequences lack the definitive 3'UTR boundary. All miRNA-mRNA interactions involving incomplete 3'UTR are removed and 954 out of 1068 miRNA-mRNA interactions are selected for further processing.

3.2 miRNA and 3'UTR alignment

Each miRNA-mRNA interaction is aligned using the dynamic programming. For the initial alignment, scoring scheme is given in Table 1.

Relation between miRNA and mRNA	Score
Complementarity between miRNA and mRNA	3
GU wobble	1
Gap in miRNA or in mRNA	-2
Otherwise	-1

Table 1. Basic scoring scheme

Using this scoring scheme a scoring matrix is built. From this scoring matrix, local alignments of miRNA and mRNA are computed. For computing local alignments, the last row of the scoring matrix is scanned and cells with higher score than the predefined threshold are selected. From each selected cell local alignments are calculated by backtracking. In this method of finding local alignments, scoring complement and GU wobble positively and gap and mismatch negatively is important. Scoring a gap or a mismatch as a positive score causes the method to fail because then in the last row of scoring matrix, score will always increase from left to right which will further cause global alignment of sequences.

In the next step all aligned sequences are filtered to reduce the false-positive rate. There are three types of filters which are applied in sequential order. The first filter selects the only those miRNA-mRNA interactions, which have perfect complementarity at seed region (position 2-7) of miRNA. The second filter is applied on the remaining miRNA-mRNA interactions. Those interactions are passed in this filter which has at most 1 mismatch in the seed region of miRNA but excellent complementarity at 3' end. To be qualified as excellent complementarity at 3' end, at least 9 out of 10 positions of miRNA should be complement to the mRNA. This parameter is adjustable in the program. After this filtering step, all interactions which does not pass through these filters are discarded. Then third filter is applied on the selected interactions. This filter selects the top 10 interactions with the minimum free energy. This filter is used to test the thermodynamic stability of miRNA-mRNA duplex.

To test the stability of miRNA-mRNA interaction, a consensus sequence representing the structure of miRNA-mRNA duplex is created. In this consensus sequence structure information is represented as follows:

Structure	Character representation
Complementarity or GU wobble	M
Internal loop	I
Bulge	B

Hairpin	H
---------	---

Table 2. Characters for consensus sequence

A generated consensus sequence is used to calculate the free energy of each miRNA-mRNA interaction using the M. Zuker's energy parameters.

3.3 Position Specific Scoring Matrix

Filtered miRNA-mRNA interactions are used to build the position specific scoring matrix (PSSM). Two types of position specific scoring matrices are built. The first type of scoring matrix is specific to one particular miRNA. The second type of PSSM is a general matrix which includes combined information from each individual miRNA specific PSSM. PSSM has four rows each for complementarity, mismatch, GU wobble and gap. The number of columns is equal to the length of miRNA. In the case of general PSSM, the number of columns is equal to 22. Whenever the length of miRNA is greater than the 22 nucleotide, 10 nucleotides from the 5' end of miRNA and 12 nucleotides from the 3' end are used to build the general PSSM. For the shorter miRNAs, columns with a zero score are added to adjust the length of miRNA in the middle part of the matrix.

The position specific scoring matrices are built by scanning the miRNA-mRNA interaction from the 5' end of mRNA towards the 3' end of

mRNA. For each base pairing at particular position, a corresponding cell from the matrix is incremented. Also only gaps in mRNAs are counted, but gaps in miRNAs are ignored.

3.4 Target Prediction

In this phase we try to predict the miRNA target based on the knowledge captured in position specific scoring matrices. This phase requires the complete database of 3'UTR regions. For this purpose, we used the PACdb (Brockman, Singh, Liu, Quinlan, Salisbury, & Graber, 2005) database. PACdb contains 3'UTR database for many species, we only used the database for humans. It contains 6051 3'UTR sequences for humans.

MicroRNA, whose targets are predicted, is aligned against the each sequence in the database. As aligning miRNA with each 3'UTR is inefficient, we tried to limit the alignment with only those 3'UTRs which had almost perfect complementarity to the seed region of miRNA. In this phase, the scoring matrix is built using the position specific scoring matrix. If miRNA is known and specific PSSM exists, then that PSSM is used, otherwise general PSSM is used. Backtracking is done using the same process as in the initial alignment to generate local alignments.

Figure 4. Overview of developed method

4 Implementation

The prediction algorithm is implemented in the perl programming language. The algorithm is divided into smaller scripts. Each script further divides into small subroutines. Interactions and sequence data from external databases are stored in the text files. In this section each individual script and their input and output is discussed.

4.1 Script 1: MicroRNA-mRNA interaction information

This script processes the miRecord database for humans and stores the miRNA-mRNA interaction information in the file called main.txt. This database does not contain the miRNA and mRNA sequences, but it only contains the references to the mirBase and the GenBank database. It stores the miRNA id, miRNA accession number, mRNA name, mRNA accession number and reference to the source article in PubMed. It generates the following output file:

>miRNA id	Accession No	mRNA Name	mRNA Accession No	Ref PMID
hsa-let-7a	MIMAT0000062	RAVER2	NM_018211	17637574
hsa-let-7a	MIMAT0000062	LIN28	NM_024674	16495412
hsa-let-7a	MIMAT0000062	NF2	NM_181833	17220301
hsa-let-7a	MIMAT0000062	KRAS	NM_004985	15766527
hsa-let-7a	MIMAT0000062	NRAS	NM_002524	15766527
hsa-let-7a	MIMAT0000062	TRIM71	NM_001039111	17890240
hsa-let-7a	MIMAT0000062	LOC561754	XM_685160	17535905
hsa-let-7a	MIMAT0000062	HMGA2	NM_003483	17600087
hsa-let-7a	MIMAT0000062	CASP3	NM_032991	18758960
hsa-let-7a	MIMAT0000062	DICER1	NM_177438	18700235
hsa-let-7a	MIMAT0000062	PRDM1	NM_182907	18583325
hsa-let-7a	MIMAT0000062	ITGB3	NM_000212	18679415
hsa-let-7a	MIMAT0000062	RAVER2	NM_018211	17637574
hsa-let-7b	MIMAT0000063	LIN28	NM_024674	15131085
hsa-let-7b	MIMAT0000063	CDK6	NM_001259	17699775
hsa-let-7b	MIMAT0000063	CCND1	NM_053056	18379589
hsa-let-7b	MIMAT0000063	CDC25A	NM_201567	17699775
hsa-let-7b	MIMAT0000063	BCL7A	NM_020993	18026111
hsa-let-7b	MIMAT0000063	HMGA2	NM_003483	17437991
hsa-let-7c	MIMAT0000064	MYC	NM_002467	17877811

Figure 5. Contents of main.txt file

4.2 Script 2: Retrieve sequences from mirBase and GenBank

This script uses main.txt as the input file and reads it line by line. It then stores the list of miRNAs and mRNAs. First it extracts the miRNA sequences from the mirBase database and stores them in individual file named '{miRNA_accession_no}_miRna.txt'. It then downloads the mRNA sequences from the GenBank along with CDS information using the web based API called 'Entrez programming utilities'. It processes the each mRNA sequences and extracts the 3'UTR regions from it. It then stores the each 3'UTR in the text files named '{mRNA_accession_no}_mRna.txt'.


```

A >hsa-miR-133b MIMAT0000770
UUUGGUCCCCUUAACACGCUA

B >gi|156071469|ref|NM_001194.3| Homo sapiens hyperpolarization activated cyclic nucleotide-gated potassium channel 2 (HCN2)
GGCCGGCGGCGGCGGCGGCTCCGCTCCGCACTGCCCCGGCGCGCCTCGCCATGGACGCGCGGGGG
CGGCGGGCGGCGGCGGAGAGCCCGGGCGGACCCCGCGCGGCGGCGCGCGCGCGCGCGCGCGCG
CCCCCAACAGCAGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
AGCACCGCGCGCGCGGAGCGTTGCCCGGAGGCGCGGATGAGGGCGCGCGCGCGCGCGCGCGCGCG
CAGCGCGCAGCAGCTCGTGGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
GGCGAGTGGCGCGCGCGAGCGCGAGTGCAGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
TGTCGTTCTCGTGGCGCGGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
CGAGGAGGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
CTCCTGCAGCGCGGCGTCAACAAGTTCTCGCTGCGGATGTTGCGCAGCCAGAAGCGCGTGGAGCGCGAGC
AGGAGCGCGTCAAGTGGCGCGGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
CACCATGCTGCTGTTTCATGGTGGGAACTCATCATCATCCAGTGGGCATCACCTTCTTCAAGGATGAG
ACCACTGCCCCGTGGATCGTGTTCAGCTGGTCTCGGACACCTTCTTCTCATGGACCTGGTGTGAACT
TCCGCACCGGCTTGTGATCGAGGACAAACAGGAGATCATCTGGACCCGAGAAGATCAAGAAGAAGTA
TCTGCGCACGTGGTTCGTGGTGGACTTCGTGTCTCCATCCCGTGGACTACATCTTCTTATCGTGGAG
AAGGGCATTGACTCGAGGTCTACAAGACGGCAGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
GCCTCTGCGGCTGCTGCGCCTCTCAGCGCTGATCGCTACATCCATCAGTGGGAGGAGATCTCCACAT
GACCTATGACCTGGCCAGCGCGGTGATGAGGATCTGCAATCTCAICAGCATGATGCTGCTGCTGCCAC
TGGGACGCGTGCAGTTCCTGGTGCCTATGCTGCAGGACTTCCGCGCAACTGCTGGGTGTCATCA
ATGGCATGGTGAACCACTCGTGGAGTGAAGTGTACTCTTCGCACTCTTCAAGGCCATGAGCCACATGCT
GTGCATCGGTACGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG
GTGGGTGCCACCTGCTACGCCATGTTTCATCGGCCAGCGCACTGCCCTCATCCAGTGGTGGACTCTCGC
GGCGCCAGTACAGGAGAAGTACAAGCAGGTGGAGCAGTACATGTCCTTCCACAAGCTGCCAGCTGACTT
CCGCCAGAAGATCCAGACTACTATGAGCACCCTTACCAGGCGAAGATGTTTGACGAGGACAGCATCTG

```

Figure 6. (A) MicroRNA output file (B) mRNA output file

4.3 Script 3: Align miRNA and 3'UTR

This script uses main.txt as the input file and reads it line by line and extracts miRNA and mRNA accession numbers from it. For a particular line in main.txt, it uses the miRNA and the mRNA accession numbers to search for the corresponding miRNA and mRNA sequences in the same directory where the main.txt is stored. It then reads the miRNA sequence and mRNA sequences from the individual file. The miRNA sequence is aligned against the mRNA sequence using dynamic programming. The scoring matrix is built using the scoring scheme listed in Table 1. This matrix has a number of rows equal to the length of the miRNA and the number of columns equal to the length of the mRNA. One such scoring matrix is shown in figure Figure 7.

A	U	G	G	U	U	A	U	A	A	A	A
0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	-1	-1	3	1	3	3	3	3
2	2	2	2	0	-2	2	2	4	6	6	6
1	3	1	1	3	1	0	3	2	4	5	5
6	4	4	2	1	2	4	2	6	5	7	8
5	7	5	3	3	2	2	5	4	5	5	6
10	8	6	4	6	6	4	5	4	3	4	4
11	9	11	9	7	5	5	3	4	3	2	3
13	11	10	12	10	8	8	6	6	7	6	5
11	16	14	12	15	13	11	11	9	7	6	5
14	14	15	13	15	18	16	14	12	10	8	6
15	17	15	14	16	18	17	19	17	15	13	11
13	16	16	14	15	17	17	18	18	16	14	12
15	14	17	17	15	15	20	18	21	21	19	17
13	14	15	18	16	14	18	19	21	24	24	22
14	12	15	16	17	15	17	17	22	24	27	27
19	17	15	14	19	20	18	20	20	22	25	26
20	18	20	18	17	18	19	18	19	20	23	24
21	19	21	23	21	19	17	18	17	18	21	22
24	24	22	21	26	24	22	20	18	16	19	20
25	23	27	25	24	25	23	21	19	17	17	18
26	26	25	26	26	25	24	24	22	20	18	16
26	29	27	25	29	29	27	27	25	23	21	19
30	28	30	28	27	28	32	30	30	28	26	24

Figure 7. Scoring matrix

After building the scoring matrix, the script scans the last row of the matrix to select those cells which have higher value than the predefined threshold. Then from each selected cell backtracking is done to extract the local alignments of miRNA and 3'UTR. The process of backtracking is shown in Figure 8. It shows only the small part of the complete local alignment between the miRNA and 3'UTR.

A	U	G	G	U	U	A	U	A	A	A	A
0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	-1	-1	3	1	3	3	3	3
2	2	2	2	0	-2	2	2	4	6	6	6
1	3	1	1	3	1	0	3	2	4	5	5
6	4	4	2	1	2	4	2	6	5	7	8
5	7	5	3	3	2	2	5	4	5	5	6
10	8	6	4	6	6	4	5	4	3	4	4
11	9	11	9	7	5	5	3	4	3	2	3
13	11	10	12	10	8	8	6	6	7	6	5
11	16	14	12	15	13	11	11	9	7	6	5
14	14	15	13	15	18	16	14	12	10	8	6
15	17	15	14	16	18	17	19	17	15	13	11
13	16	16	14	15	17	17	18	18	16	14	12
15	14	17	17	15	15	20	18	21	21	19	17
13	14	15	18	16	14	18	19	21	24	24	22
14	12	15	16	17	15	17	17	22	24	27	27
19	17	15	14	19	20	18	20	20	22	25	26
20	18	20	18	17	18	19	18	19	20	23	24
21	19	21	23	21	19	17	18	17	18	21	22
24	24	22	21	26	24	22	20	18	16	19	20
25	23	27	25	24	25	23	21	19	17	17	18
26	26	25	26	26	25	24	24	22	20	18	16
26	29	27	25	29	29	27	27	25	23	21	19
30	28	30	28	27	28	32	30	30	28	26	24

Figure 8. Backtracking

All local alignments are then passed through the two filters. Those alignments are stored for the next stage which passes through either the first or the second filters. All other alignments are discarded. Only those alignments get passed through the first filter which have perfect complementarity between the seed region of miRNA and 3'UTR. Those alignments which do not have perfect complementarity in the seed region and have only one mismatch in the seed region, but near perfect complementarity exist at the 3' end are passed through the second filter.

All passed alignments are then stored in one file for each miRNA. This file includes all local alignments between one particular miRNA and all mRNAs listed in the main.txt file. This file also stores the starting position of the local alignment with respect to the mRNA.

```

192
GACUAUAUUAUC~ACCU~UCUU~
UUGGUGUG~UUGGAUG~AUGGAGU

192 _____ Position
GACUAUAUUAUC~ACCU~UCUU~ mRNA
UUGGUGUG~UUGGAUG~AUGGAGU miRNA

192
GACUAUAUUAUC~ACCUUCUU~
UUGGUGUG~UUGGAUG~AUGGAGU

192
GACUAUAUUAUC~ACCUUCUUUG
UUGGUGUG~UUGGAUG~AUGGAGU

192
GACUAUAUUAUC~ACCUUCUUUGU
UUGGUGUG~UUGGAUG~AUGGAGU~

275
AACAAUGCAACUUUUAUUACUGU~
UUGGUGUGUUGGA~~U~GAUGGAGU

275
AACAAUGCAACUUUUAUUACUGUU~
UUGGUGUGUUGGA~~U~GAUGG~AGU

275
AACAAUGCAACUUUUAUUACUGUU~

```

Figure 9. Output file for local alignments

4.4 Script 4: Filtering with minimum free energy

This script reads the local alignments from the file for a particular miRNA. It first builds the consensus sequence using the characters listed in Table 2. This consensus sequence is used to calculate the free energy of each local alignment using the energy values listed in Table 3. The top 10 alignments with the minimum free energy are stored in the file for further processing.

Stacking energies for base pairs						
	A/U	C/G	G/C	U/A	G/U	U/G
A/U	-0.9	-1.8	-2.3	-1.1	-1.1	-0.8
C/G	-1.7	-2.9	-3.4	-2.3	-2.1	-1.4
G/C	-2.1	-2.0	-2.9	-1.8	-1.9	-1.2
U/A	-0.9	-1.7	-2.1	-0.9	-1.0	-0.5
G/U	-0.5	-1.2	-1.4	-0.8	-0.4	-0.2
U/G	-1.0	-1.9	-2.1	-1.1	-1.5	-0.4
Destabilizing energies for loops						
Number of bases	1	5	10	20	30	
Internal	-	5.3	6.6	7.0	7.4	
Bulge	3.9	4.8	5.5	6.3	6.7	
Hairpin	-	4.4	5.3	6.1	6.5	

Table 3. M. Zuker's energy parameters

```

93
AAUGCAUGAUCAAA~UGCAACCUCAC
UUG~GUA~U~GUUGGAUGAUGGAGU~
MMMBMMBMBMMMIIMMMIMMMMMB
6.5

100 ----- Position
GAUCAA AUGCAACCUCACAACCUUGG mRNA
UUGGU~~AUGUUGGA~UGAUGGAGU~ miRNA
MMMMBMBMMMMMMMMBMMIMMMMMB Consensus sequence
-9.6 ----- Free energy value

150
AGCCAUA~AUGUAAACUGCCUC~
UUGGUAUGU~UGGAUGAUGGAGU
MMMMMMBMBMMMIIMMMMMMMB
-5.3

```

Figure 10. Output file for intractions with free energy

4.5 Script 5: Build Position Specific Scoring Matrix (PSSM)

This script reads files generated by script 4. It builds two types of position specific scoring matrices one for each miRNA and one general matrix. The miRNA specific matrix has four rows and columns equal to the length of the miRNA. Each row represents the information about the complementarity, mismatch, GU wobble or gap at a particular position in miRNA. The script reads each alignment nucleotide by nucleotide from the 5' end of mRNA and records this information in the matrix.

Comp	6	32	32	20	6	14	10	28	20	10	14	
Mis	16	2	2	2	2	2	10	6	6	16	8	12
GU	14	2	2	14	28	6	2	2	6	2	2	2
Gap	2	2	2	2	2	16	16	2	6	10	14	6

Figure 11. MicroRNA specific PSSM

A general matrix is also built using the same procedure but the number of columns in the matrix is fixed to 22. In the case of longer miRNAs 12 positions at 5' end and 10 positions at 3' end are selected to build the general PSSM and positions between them are discarded. In the case of shorter miRNAs 10 positions at the 5' end are used as such while the remaining positions are shifted towards the 3' end by introducing the columns with zero values in each cell.

4.6 Script 6: Predicting miRNA targets

This script uses the position specific matrices to predict the targets of miRNAs. The user enters the miRNA sequence in the input.txt file. The script reads the input file and searches for the PSSM. The script uses the miRNA specific PSSM if it exists, otherwise it uses the general PSSM. It aligns the miRNA against the 3'UTR in the PACdb. It builds the scoring matrix using the PSSM. Then it backtracks the local alignments from the last column of the scoring matrix. The major difference between this script and script 3 is that it

uses the PSSM for scoring the base-pairing between the miRNAs and mRNAs.

```

>1:43479774:43475604:43475604:6:ENSG00000176591:43480322:43479774
256
GC~CUAGAGGGAGGUA~UUUGG
UGAGAUUUCCCUUCGUGAAACA
MMBMMMMMMMMMMMMBMMMI
-17

>7:43630967:43633506:43633505:2:ENSG00000164543:43589368:43630966
1064 ———— Position
ACUCUGUAAGUAGGAGGUACUUUG~ ———— mRNA
UGAGAU~UUC~~CCUUCGUGAAACA ———— miRNA
MMMMMBMMBBMMMMMMMMMMMB ———— Consensus sequence
-21.3 ———— Energy Value

>7:43630967:43633506:43633505:2:ENSG00000164543:43589368:43630966
1064
ACUCUGUAAGUAGGAGGUACUUUGUC
UGAGAU~UUC~~CCUUCGUGAAACA~~
MMMMMBMMBBMMMMMMMMMMMBB
-19.2

>14:73467998:73468585:73468584:4:ENSG00000119725:73428632:73467997
314
ACUGUUGAGGGGAGGUUUCCUUUG~
UGA~GAUUCCCUUCG~~~UGAAACA
MMBMMMMMMMMMMMI I IIMMMMB
-17.8

>12:93489342:93485030:93485030:6:ENSG00000057704:93568337:93489342

```

Figure 12. Output file for predicted interactions

5 Results

All the miRNA targets predicted by our algorithm are compared with the currently available tools. miRecords provides a database of predicted targets by all popular prediction tools. Using this database, predicted targets by our algorithm are compared with miRanda, miTarget, PicTar, RNAHybrid and TargetScan and the results are summarized in Table 4, 5 and 6. Due to unavailability of DIANA-microT database, results are not compared with it. PACdb uses the 3'UTR sequences from the Ensembl database while miRecords uses the sequences from the GenBank. Our predicted interaction contains the sequence from the Ensembl database and it introduces the problem of comparing these sequences with the miRecords predicted database sequences. To resolve the issues, we used the BLAST tool to get the GenBank sequence equivalent of Ensembl sequence. Only those sequences are selected which are 100 percent identical. There are still some sequences which do not have GenBank equivalent.

Ensembl ID	RefSeq	Number of interaction predicted by					
		Our Method	miRanda	miTarget	PicTar	RNAHybrid	TargetScan TargetScan
ENSG00000198018	NM_020354	2	1	0	0	1	0

ENSG00000155926	NM_003235	2	0	0	0	1	0
ENSG00000184402	NM_198935	1	0	0	0	1	0
ENSG00000177119	NM_001025356	4	2	0	0	1	0
ENSG00000116285	NM_018948	3	0	0	0	1	0
ENSG00000138696	NM_001203	5	0	0	0	1	0
ENSG00000155099	NM_018710	2	1	1	1	1	1
ENSG00000167968	NM_001374	2	1	1	0	1	0
ENSG00000164695	NM_152284	3	1	0	0	1	0
ENSG00000135218	NM_001001547	2	0	0	0	1	0
ENSG00000183579	NM_032173	3	2	2	0	1	0
ENSG00000183114	NM_207334	2	0	0	0	1	0
ENSG00000128595	-	2					
ENSG00000156804	-	1					
ENSG00000106868	-	2					
ENSG00000203935	-	2					
ENSG00000167081	-	2					
ENSG00000163171	-	1					
ENSG00000188394	-	3					
ENSG00000204399	-	4					

Table 4. Comparison of MIMAT0000068 (hsa-miR-15a) interactions

Ensembl ID	RefSeq	Number of interaction predicted by					
		Our Method	miRanda	miTarget	PicTar	RNAHybrid	TargetScan TargetScanS
ENSG00000164543	NM_004760	4	1	0	0	1	1
ENSG00000119725	NM_021188	4	0	0	0	1	0
ENSG00000057704	NM_020698	2	0	0	0	1	0
ENSG00000130150	NM_152581	2	0	0	0	1	0
ENSG00000124145	NM_002999	1	0	0	0	1	0
ENSG00000150672	NM_001364	2	2	1	0	1	2
ENSG00000106355	NM_012322	2	0	0	0	1	1
ENSG00000147724	NM_015912	1	0	0	0	1	1
ENSG00000163251	NM_003468	1	1	0	0	1	1
ENSG00000174298	NM_153027	3	1	1	0	1	1
ENSG00000172845	NM_003111	2	2	0	0	1	0
ENSG00000154277	NM_004181	2	0	0	0	1	0
ENSG00000100441	NM_015299	2	1	1	0	1	1
ENSG00000157851	NM_020134	2	2	0	0	1	0
ENSG00000126259	NM_032123	2	0	0	0	1	0

ENSG00000176591	-	2					
ENSG00000165055	-	5					
ENSG00000162419	-	3					
ENSG00000204334	-	4					
ENSG00000114125	-	2					
ENSG00000204712	-	2					

Table 5. Comparison of MIMAT0002867 (hsa-miR-520h) interactions

Ensembl ID	RefSeq	Number of interaction predicted by					
		Our Method	miRanda	miTarget	PicTar	RNAHybrid	TargetScan TargetScanS
ENSG00000144021	NM_004804	3	1	0	0	1	0
ENSG00000136531	NM_021007	5	1	0	0	1	0
ENSG00000127946	NM_005338	2	1	0	0	1	0
ENSG00000154096	NM_006288	3	1	0	0	1	0
ENSG00000143033	NM_007358	3	0	0	0	1	0

ENSG00000134438	NM_013435	2	0	0	0	1	0
ENSG00000165802	NM_015537	2	0	1	0	1	0
ENSG00000077684	NM_199320	3	1	0	0	1	0
ENSG00000205795	NM_001037160	2	0	0	0	1	1
ENSG00000058063	NM_014616	2	0	0	0	2	0
ENSG00000109618	NM_016955	3	0	0	0	1	0
ENSG00000100216	NM_020243	2	0	0	0	1	0
ENSG00000155827	NM_019592	1	0	0	0	1	0
ENSG00000080823	-	1					
ENSG00000205742	-	2					
ENSG00000187742	-	2					
ENSG00000198488	-	2					
ENSG00000147507	-	1					
ENSG00000134698	-	3					
ENSG00000144481	-	1					
ENSG00000124493	-	2					
ENSG00000164342	-	2					
ENSG00000010318	-	1					

Table 6. Comparison of MIMAT0000255 (hsa-miR-34a) interactions

All interactions between miRNA and mRNA identified by RNAHybrid are also identified by our method. Compared to RNAHybrid our method identified, in most cases, more targets for same miRNA-mRNA interaction. miRanda misses some miRNA targets which are predicted by our method. miTarget, PicTar and TargetScan only identified some miRNA targets.

The reason for identifying more targets for same miRNA-mRNA interaction is the presence of overlapping targets. Our method does not detect the overlapping targets and reports them as different targets.

6 Conclusion

We developed a miRNA target prediction algorithm based on dynamic programming and position specific scoring scheme. Our algorithm was able to match with RNAHybrid. We were only recording the top 50 stable interactions between the miRNA and mRNA. There was a significant difference between it and the tools that were using the conservation criteria. Our method might have missed some of the conserved miRNA targets because we did not consider it while predicting miRNA targets. The other problem we encountered was that it was difficult to compare the results with the other available tools because of different databases.

The quality of target prediction with this method can be further improved by adding the conservation criteria. Further analysis of miRNA-mRNA interactions to uncover more position based features can yield more accurate target prediction algorithms.

Bibliography

- Bartel, D. P. (2004). MicroRNAs: Genomics, Biogenesis, Mechanism, and Function. *Cell*, 116 (2), 281 - 297.
- Brockman, J., Singh, P., Liu, D., Quinlan, S., Salisbury, J., & Graber, J. (2005). PACdb: PolyA Cleavage Site and 3'-UTR Database. *Bioinformatics*, 21 (18), 3691-3693.
- Enright, A., John, B., Gaul, U., Tuschl, T., Sander, C., & Marks, D. (2003). MicroRNA targets in Drosophila. *Genome Biology*, 5 (1), R1.
- Friedman, R. C., Farh, K. K.-H., Burge, C. B., & Bartel, D. P. (2009). Most mammalian mRNA are conserved targets of microRNAs. *Genome Res*, 19 (1), 92 - 105.
- Kim, S.-K., Nam, J.-W., Rhee, J.-K., Lee, W.-J., & Zhang, B.-T. (2006). miTarget: microRNA target gene prediction using a support vector machine. *BMC Bioinformatics*, 7 (1), 411.
- Kiriakidou, M., Nelson, P. T., Kouranov, A., Fitziev, P., Bouyioukos, C., Mourelatos, Z., et al. (2004). A combined computational-experimental approach predicts human microRNA targets. *Genes Dev*, 18, 1165 - 1178.
- Krek, A., Grun, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., et al. (2005). Combinatorial microRNA target predictions. *Nat Genet*, 37 (5), 495 - 500.
- Lee, R. C., Feinbaum, R. L., & Ambros, V. (1993). The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, 75 (5), 843 - 854.
- Lee, Y., Jeon, K., Lee, J. T., Kim, S., & Kim, V. N. (2002). MicroRNA maturation: stepwise processing and subcellular localization. *EMBO*, 21 (17), 4663 - 4670.
- Lewis, B. P., Shih, I.-h., Jones-Rhoades, M. W., Bartel, D. P., & Burge, C. B. (2003). Prediction of Mammalian MicroRNA Targets. *Cell*, 115 (7), 787 - 798.
- Lindow, M., & Gorodkin, J. (2007). Principles and Limitations of Computational MicroRNA Gene and Target Finding. *DNA and Cell Biology*, 26 (5), 339-351.

- Maziere, P., & Enright, A. J. (2007). Prediction of microRNA targets. *Drug Discovery Today* , 12 (11), 452 - 458.
- McDanel, T. G. (2009). MicroRNA: mechanism of gene regulation and application to livestock. *Journal of animal science* , 87, E21-E28.
- Nam, J.-w., Shin, K.-r., Han, J., Lee, Y., Kim, V. N., & Zhang, B.-t. (2005). Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Res* , 3570--3581.
- Sethupathy, P., Corda, B., & Hatzigeorgiou, A. G. (2006). TarBase: A comprehensive database of experimentally supported animal microRNA targets. *RNA* , 12, 192-197.
- Stark, A., Brennecke, J., Russell, R. B., & Cohen, S. M. (2003). Identification of Drosophila MicroRNA Targets. *PLoS Biol* , 1 (3), e60.
- Wang, X. (2008). miRDB: a microRNA target prediction and functional annotation database with a wiki interface. *RNA* , 14 (6), 1012-1017.
- Xiao, F., Zuo, Z., Cai, G., Kang, S., Gao, X., & Li, T. (2009). miRecords: an integrated resource for microRNA–target interactions. *Nucleic Acids Research* , 37, D105–D110.
- Xie, X., Lu, J., Kulbokas, E. J., Golub, T. R., Mootha, V., Lindblad-Toh, K., et al. (2005). Systematic discovery of regulatory motifs in human promoters and 3'UTRs by comparison of several mammals. *Nature* , 338 - 345.
- Yoon, S., & Micheli, G. D. (2006). Computational identification of microRNAs and their targets. *Birth Defects Research Part C: Embryo Today: Reviews* , 78 (2), 118 - 128.
- Zhang, B., Pan, X., Wang, Q., Cobb, G. P., & Anderson, T. A. (2006). Computational identification of microRNAs and their targets. *ScienceDirect* , 395–407.

Appendices

Appendix A: Source code for miRNA target predictor

```
use POSIX;
$dir = 'C:/home/rupinder/Project/database/';
my $inputFile = $dir."input/input.txt";
my $databaseFile = $dir."pac/human.fa";
my $miMap = $dir."map.txt";
my $miFileName = "";
my $globalScore = $dir."score/global.txt";
my $sOutputFile = $dir."result/soutput.txt";
my $gOutputFile = $dir."result/goutput.txt";
my @sMatrix = ();
my @gMatrix = ();
my $stopInterNo = 50;
my @sTopList = ();
my @gTopList = ();
my $flagSMatrix = 0;
$negInfinity = -99999999999999999999;
my $count = 0;
my $miGap = 0;
my $miRnaLength = 0;
$allowed3sideMisMatch = 1;
%energyMap = ('AU' => 0, 'CG' => 1, 'GC' => 2, 'UA' => 3, 'GU' => 4, 'UG' => 5,
              'I' => 0, 'B' => 1, 'H' => 2,
              '1' => 0, '5' => 1, '10' => 2, '20' => 3, '30' => 4);
@bpEnergy = ([ -0.9, -1.8, -2.3, -1.1, -1.1, -0.8],
             [-1.7, -2.9, -3.4, -2.3, -2.1, -1.4],
             [-2.1, -2.0, -2.9, -1.8, -1.9, -1.2],
             [-0.9, -1.7, -2.1, -0.9, -1.0, -0.5],
             [-0.5, -1.2, -1.4, -0.8, -0.4, -0.2],
             [-1.0, -1.9, -2.1, -1.1, -1.5, -0.4]);
@loopEnergy = ( [5.3, 5.3, 6.6, 7.0, 7.4],
                [3.9, 4.8, 5.5, 6.3, 6.7],
                [4.4, 4.4, 5.3, 6.1, 6.5]);
for(my $i=0; $i<$stopInterNo; $i++){
    for(my $j=0; $j<6; $j++){
        $sTopList[$i][$j] = '-';
        $gTopList[$i][$j] = '-';
    }
}
unless ( open(INPUT, $inputFile) ) {
    print "Cannot open file \"$inputFile\"\n\n";
```

```

    goto error;
}
my @input = <INPUT>;
chomp @input;
close INPUT;
my $miRna = $input[0];
chomp $miRna;
$miRna =~ s/\s//g;
$miRnaLength = length($miRna);
unless ( open(DATABASE, $databaseFile) ) {
    print "Cannot open file \"$databaseFile\"\n\n";
    goto error;
}
unless ( open(MIMAP, $miMap) ) {
    print "Cannot open file \"$miMap\"\n\n";
    goto error;
}
while($mapLine = <MIMAP>){
    chomp $mapLine;
    my @mapArray = split(/\t/, $mapLine);
    my $miA = $mapArray[1];
    chomp $miA;
    if($miA eq $miRna){
        $miFileName = $dir."score/".$mapArray[0]."_score.txt"; #specific
        mirna score file
        $sOutputFile = $dir."result/".$mapArray[0]."_soutput.txt";
        last;
    }
}
$miRna = reverse($miRna);
$miRna = seqComplement($miRna);
if($miFileName ne ""){
    my $sMatrixRef = readMatrix($miFileName);
    @sMatrix = @$sMatrixRef;
    $flagSMatrix = 1;
}
my $gMatrixRef = readMatrix($globalScore);
@gMatrix = @$gMatrixRef;
while($annotation = <DATABASE>){
    print "\nProcessing: ", $annotation;
    $mRna = <DATABASE>;
    chomp $mRna;
    $mRna =~ s/\s//g;
    my $pTest = preFilter($mRna, $miRna);
}

```

```

#my $pTest = 1;
if($pTest == 1){
if($flagSMatrix == 1){
    $matrixRef = initialize($mRna,$miRna);
    $alignedMatrixRef = alignment($matrixRef,\@sMatrix);
    backTrace($alignedMatrixRef,\@sTopList,$annotation);
}else{
    $matrixRef = initialize($mRna,$miRna);
    $alignedMatrixRef = alignment($matrixRef,\@sMatrix);
    backTrace($alignedMatrixRef,\@gTopList,$annotation);
}
}
}
if($flagSMatrix == 1){
    output(\@sTopList,$sOutputFile);
}

sub preFilter{
    my $mRna = shift(@_);
    my $miRna = shift(@_);
    my $flag = 0;
    my $seed = substr($miRna,-7,6);
    if($mRna =~ m/$seed/){
        $flag = 1;
    }else{
        for(my $i=0;$i<6;$i++){
            my $first = substr($seed,0,$i);
            my $last = substr($seed,$i+1);
            if($mRna =~ m/($first)\D($last)/){
                $flag = 1;
            }
        }
    }
    return $flag;
}

sub readMatrix{
    my $file = shift(@_);
    my @matrix = ();
    unless ( open(FILE,$file) ) {
        print "Cannot open file \"$file\"\n\n";
        goto error;
    }
    my @sContent = <FILE>;

```

```

close FILE;
for my $l (0..$#sContent){
    my @IParts = split(/\t\t/, $sContent[$l]);
    for my $p (1..$#IParts){
        $matrix[$l][$p-1] = $IParts[$p];
    }
}
\@matrix;
}

sub max {
    my $len = scalar @_ ;
    my $max = $_[0];
    for(my $i=1;$i<$len;$i++){
        my $element = $_[ $i ];
        if($element > $max){
            $max = $element;
        }
    }
    return $max;
}

sub min{
    my $len = scalar @_ ;
    my $min = $_[0];
    for(my $i=1;$i<$len;$i++){
        my $element = $_[ $i ];
        if($element < $min){
            $min = $element;
        }
    }
    return $min;
}

sub seqComplement{
    my $seq = shift(@_);
    $seq =~ tr/AUCGaucg/UAGCUAGC/;
    return $seq;
}

sub wobble{
    my $seq = shift(@_);
    $seq =~ tr/UGACugac/GUzzGUzz/;
    return $seq;
}

```

```

}

#intialize matrix.
sub intialize {
    my $mRna = shift(@_);
    my $mRnaLen = length($mRna);
    my $miRna = shift(@_);
    my $miRnaLen = length($miRna);
    my @matrix = ();
    for(my $i=0;$i<$miRnaLen+2;$i++){
        for(my $j=0;$j<$mRnaLen+2;$j++){
            for(my $k=0;$k<4;$k++){
                $matrix[$i][$j][$k] = '^';
            }
        }
    }
    $matrix[0][0][0] = '*';
    $matrix[0][1][0] = '-';
    $matrix[1][0][0] = '-';
    $matrix[1][1][0] = 0;
    for(my $j=2;$j<$mRnaLen+2;$j++){
        $matrix[0][$j][0] = substr($mRna,$j-2,1);
        $matrix[1][$j][0] = 0;
    }
    for(my $i=2;$i<$miRnaLen+2;$i++){
        $matrix[$i][0][0] = substr($miRna,$i-2,1);
        $matrix[$i][1][0] = 0;
    }
    return \@matrix;
}

```

```

#highest,match/mismatch/GU,miGap,mGap
sub alignment {
    my $m = shift(@_);
    my $s = shift(@_);
    my @matrix = @$m;
    my $iLen = $#matrix+1;
    my $jLen = ${$matrix[0]}+1;
    for(my $i=2;$i<$iLen;$i++){
        for(my $j=2;$j<$jLen;$j++){
            my $mChar = $matrix[0][$j][0];
            my $miChar = $matrix[$i][0][0];
            if($mChar eq $miChar){

```

```

$matrix[$i][$j][1] = $matrix[$i-1][$j-1][0] + $s->[0][$i-2];

    }else{
    if(($mChar eq 'g' || $mChar eq 'G') && ($miChar eq 'a' || $miChar eq 'A')){
        $matrix[$i][$j][1] = $matrix[$i-1][$j-1][0] + $s->[2][$i-2];
    }
    elsif(($mChar eq 'u' || $mChar eq 'U') && ($miChar eq 'c' || $miChar eq
'C')){
        $matrix[$i][$j][1] = $matrix[$i-1][$j-1][0] + $s->[2][$i-2];
    }
    else{
        $matrix[$i][$j][1] = $matrix[$i-1][$j-1][0] + $s->[1][$i-2];
    }
    }
    $matrix[$i][$j][2] = $matrix[$i-1][$j][0] + $miGap;

$matrix[$i][$j][3] = $matrix[$i][$j-1][0] + $s->[3][$i-2];
$matrix[$i][$j][0] = max($matrix[$i][$j][1],$matrix[$i][$j][2],$matrix[$i][$j][3]);

    }
}
return \@matrix;
}

sub backTrace{
    my $mRef = shift(@_);
    my $s = shift(@_);
    my $annotation = shift(@_);
    my @matrix = @$mRef;
    my $iLen = $#matrix+1;
    my $jLen = $#{$matrix[0]}+1;
    my @tempArray = ();
    for(my $j=2;$j<$jLen;$j++){
        push(@tempArray,$matrix[$iLen-1][$j][0])
    }
    my $maxScore = max(@tempArray);
    my $minScore = min(@tempArray);
    my $threshold = ($maxScore - $minScore)*0.4;
    $threshold = ceil($threshold);
    for(my $j=2;$j<$jLen;$j++){
        if($matrix[$iLen-1][$j][0] > $threshold){
            traceOne($j,$mRef,$s,$annotation);
        }
    }
}

```

```

}

sub traceOne{
    my $startIndex = shift(@_);
    my $mRef = shift(@_);
    my $s = shift(@_);
    my $annotation = shift(@_);
    my $mRnaThirdPart = substr($mRna,$startIndex-1);
    my $j = $startIndex;
    my $flag = 1;
    while($flag == 1){
        $flag =
    traverse($mRef,$startIndex,$j,$mRnaThirdPart,$s,$annotation);
    }
}

sub traverse{
    my($mRef,$startIndex,$j,$mRnaThirdPart,$s,$annotation) = @_;
    my $i = $#{$mRef};
    my $iDouble = 0;
    my $jDouble = 0;
    my $flag = 0;
    my $mRnaSecondPart = "";
    my $mRnaFirstPart = "";
    my $miRnaPart = "";
    while($i > 1 && $j > 1){

        if($mRef[$i][$j][0] == $mRef[$i][$j][1]){
            if(($mRef[$i][$j][0] == $mRef[$i][$j][2]) || ($mRef[$i][$j][0]
== $mRef[$i][$j][3])){
                $iDouble = $i;
                $jDouble = $j;
            }
            #match/mismatch/GU case
            $mRnaSecondPart = $mRnaSecondPart.$mRef[0][$j][0];
            $miRnaPart = $miRnaPart.$mRef[$i][0][0];
            $i = $i-1;
            $j = $j-1;
        }
        elseif($mRef[$i][$j][0] == $mRef[$i][$j][2]){
            if($mRef[$i][$j][0] == $mRef[$i][$j][3]){

                $iDouble = $i;
                $jDouble = $j;
            }
        }
    }
}

```



```

    }
    #miGap
    $mRnaSecondPart = $mRnaSecondPart.'~';
    $miRnaPart = $miRnaPart.$mRef[$i][0][0];
    $i = $i-1;
  }
  else{
    #mGap
    $mRnaSecondPart = $mRnaSecondPart.$mRef[0][$j][0];
    $miRnaPart = $miRnaPart.'~';
    #print "\nmi$j:",$mRef[0][$j][0];
    $j = $j-1;
  }
}
if(($iDouble != 0) && ($jDouble != 0)){
  $flag = 1;
  if($mRef[$iDouble][$jDouble][0] ==
$mRef[$iDouble][$jDouble][1]){
    $mRef[$iDouble][$jDouble][1] = $negInfinity;
  }
  else{
    $mRef[$iDouble][$jDouble][2] = $negInfinity;
  }
}
$mRnaSecondPart = reverse($mRnaSecondPart);
$mRnaFirstPart = substr($mRna,0,$j-1);
$miRnaPart1 = ('~' x length($mRnaFirstPart)).reverse($miRnaPart).('~' x
length($mRnaThirdPart));
$miRnaPart = seqComplement($miRnaPart);
$miRnaPart = reverse($miRnaPart);
$mRnaJoin = $mRnaFirstPart.$mRnaSecondPart.$mRnaThirdPart;
my $aa = reverse($miRnaPart);
my $cSeq = cSeq($mRnaSecondPart,$miRnaPart);
my $e = computeEnergy($mRnaSecondPart,$miRnaPart,$cSeq);
recordInteraction($s,$annotation,$j,$mRnaSecondPart,$miRnaPart,$cSeq
,$e);
return $flag;
}

sub filter{
  my $cSeq = shift(@_);
  my $flag = 0;
  my $seedFlag = checkSeed($cSeq);
  if($seedFlag == 1){

```

```

        $flag = 1;
    }else{
        $flag = check3side($cSeq);
    }
    return $flag;
}

```

#checks complementary in seed region

```

sub checkSeed{
    my $seq = shift(@_);
    $seq = reverse($seq);
    my $flag = 1;
    for(my $i=1;$i<7;$i++){
        if(substr($seq,$i,1) ne 'M'){
            $flag = 0;
            last;
        }
    }
    return $flag;
}

```

#check compl. on 3' region

```

sub check3side{
    my $seq = shift(@_);
    my $flag = 0;
    my $count = 0;
    my $countSeed = 0;
    my $seq2 = reverse($seq);
    for(my $i=1;$i<7;$i++){
        if(substr($seq2,$i,1) eq 'M'){
            $countSeed++;
        }
    }
    if($countSeed == 5){
        for(my $i=0;$i<10;$i++){
            if(substr($seq,$i,1) eq 'M'){
                $count++;
            }
        }
    }
    if($count >= (10-$allowed3sideMismatch)){
        $flag = 1;
    }
    return $flag;
}

```

```

}

#returns consensus sequence for structure
sub cSeq{
    my $mRnal = shift(@_);
    my $miRnal = shift(@_);
    my $cSeq = "";
    my $loopLen = 0;
    my $internalLoop = 0;
    my $bulge = 0;
    for(my $i=0;$i<length($mRnal);$i++){
        #print "\nsub:",substr($mRnal,$i,1);
        if(substr($mRnal,$i,1) eq seqComplement(substr($miRnal,$i,1)) ||
substr($mRnal,$i,1) eq wobble(substr($miRnal,$i,1))){
            if($internalLoop == 1){
                $cSeq = $cSeq.'I'x$loopLen;
                $cSeq = $cSeq.'M';
                $internalLoop = 0;
                $bulge = 0;
                $loopLen = 0;
            }elseif($bulge == 1){
                $cSeq = $cSeq.'B'x$loopLen;
                $cSeq = $cSeq.'M';
                $internalLoop = 0;
                $bulge = 0;
                $loopLen = 0;
            }else{
                $cSeq = $cSeq.'M';
            }
        }
        elsif(substr($mRnal,$i,1) eq '~' || substr($miRnal,$i,1) eq '~'){
            $bulge = 1;
            $loopLen++;
        }
        else{
            $internalLoop = 1;
            $loopLen++;
        }
    }
    if($i == length($mRnal)-1){
        if($internalLoop == 1){
            $cSeq = $cSeq.'I'x$loopLen;
            $internalLoop = 0;
            $bulge = 0;
            $loopLen = 0;
        }
    }
}

```

```

        }elseif($bulge == 1){
            $cSeq = $cSeq.'B'x$loopLen;
            $internalLoop = 0;
            $bulge = 0;
            $loopLen = 0;
        }
    }
}
return $cSeq;
}

sub makeHairpin{
    my $cSeq = shift(@_);
    my $mRnal = shift(@_);
    my $miRnal = shift(@_);
    my $i=0;
    my $seq = "";
    while(substr($cSeq,$i,1) ne 'M' && $i < length($mRnal)){
        $i++;
    }
    if($i != 0){
        if($i == 1){
            if(substr($mRnal,0,1) ne '~' && substr($miRnal,0,1) ne '~'){
                $seq = 'H'x$i;
                $seq = $seq.substr($cSeq,$i);
            }else{
                $seq = $cSeq;
            }
        }else{
            $seq = 'H'x$i;
            $seq = $seq.substr($cSeq,$i);
        }
    }else{
        $seq = $cSeq;
    }
    return $seq;
}

sub recordInteraction{
    my $stopListRef = shift(@_);
    my $annotation = shift(@_);
    my $j = shift(@_);
    my $mRnal = shift(@_);
    my $miRnal = shift(@_);

```



```

for(my $i=0;$i<length($miRna)-1;$i++){
    if(substr($cSeq,$i,1) eq 'M'){
        my $lastCh = substr($cSeq,$i-1,1);
        if($i != 0 && $lastCh ne 'M'){
            $energy = $energy+loopEnergy($lastCh,$loopLen);
            $loopLen = 0;
        }
        if(substr($cSeq,$i+1,1) eq 'M'){
            my $firstPair =
substr($mRna,$i,1).substr($miRna,$i,1);
            my $secondPair =
substr($mRna,$i+1,1).substr($miRna,$i+1,1);
            $energy =
$energy+$bpEnergy[$energyMap{$firstPair}][$energyMap{$secondPair}];
        }
    }
    else{
        if(substr($mRna,$i,1) ne '~'){
            $loopLen++;
        }
        if(substr($miRna,$i,1) ne '~'){
            $loopLen++;
        }
    }
}
if($loopLen != 0){
    $energy = $energy+loopEnergy(substr($cSeq,-2,1),$loopLen);
    $loopLen = 0;
}
return $energy;
}

sub loopEnergy{
    my $lastCh = shift(@_);
    my $loopLen = shift(@_);
    my $energy = 0;
    $loopLen = limit($loopLen);
    $energy = $loopEnergy[$energyMap{$lastCh}][$energyMap{$loopLen}];
    return $energy;
}

sub limit{
    my $int = shift(@_);
    my $intR = 0;

```

```

    if($int == 1){
        $intR = 1;
    }elseif($int <= 5){
        $intR = 5;
    }elseif($int <= 10){
        $intR = 10;
    }elseif($int <= 20){
        $intR = 20;
    }else{
        $intR = 30;
    }
    return $intR;
}

sub output{
    my $stopList = shift(@_);
    my $file = shift(@_);
    for(my $i=0;$i<$count;$i++){
        if($stopList->[$i][0] ne '-'){
            outputInteraction($stopList->[$i][0],$stopList->[$i][1],$stopList-
>[$i][2],
                            $stopList->[$i][3],$stopList->[$i][4],$stopList->[$i][5],$file);
        }
    }
}

sub outputInteraction{
    my $annotation = shift(@_);
    my $j = shift(@_);
    my $mRnal = shift(@_);
    my $miRnal = shift(@_);
    my $cSeq = shift(@_);
    my $e = shift(@_);
    my $interFilename = shift(@_);
    open ABC, ">>$interFilename";
    print ABC $annotation;
    print ABC $j,"\n";
    print ABC $mRnal,"\n";
    print ABC $miRnal,"\n";
    print ABC $cSeq,"\n";
    print ABC $e,"\n\n";
    close ABC;
}

```

```

sub printMatrix {
    my $ref = shift(@_);
    my @eMatrix = @$ref;
    my $iLen = $#eMatrix+1;
    my $jLen = ${eMatrix[0]}+1;
    for(my $i=0;$i<$iLen;$i++){
        for(my $j=0;$j<$jLen;$j++){
            print $eMatrix[$i][$j];
            if($j != $jLen-1){
                print "\t\t";
            }
        }
        print "\n";
    }
}

sub printMatrix2 {
    my $m = shift(@_);
    my @matrix = @$m;
    my $iLen = $#matrix+1;
    my $jLen = ${matrix[0]}+1;
    $outputFilename = $dir."output/output.txt";
    unless ( open(OUTPUTFILE, ">$outputFilename") ) {
        print "Cannot open file \"$outputFilename\"\n\n";
        goto error;
    }

    for(my $i=0;$i<$iLen;$i++){
        for(my $j=0;$j<$jLen;$j++){
            my @array = ();
            for(my $k=0;$k<4;$k++){
                push(@array,$matrix[$i][$j][$k]);
            }
            my $join = join(':',@array);
            print OUTPUTFILE $matrix[$i][$j][0];
            if($j != $jLen-1){
                print OUTPUTFILE "\t\t";
            }
        }
        print OUTPUTFILE "\n";
    }
}

close OUTPUTFILE;
}

print "\nDone";

```