View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Sabanci University Research Database

MICRORNA TARGET PREDICTION BY CONSTRAINT PROGRAMMING

by

Tuğrul Tekbulut

Submitted to The Graduate School of Engineering and Natural Sciences

in partial fulfillment of

the requirements for the degree of

Master of Science

Sabancı University

Spring 2006

MICRORNA TARGET PREDICTION BY CONSTRAINT PROGRAMMING

Approved by

Doç. Dr. Uğur Sezerman _____
Thesis Supervisor

Prof. Dr. Neş'e Bilgin_____

Yrd. Doç Dr. Yücel Saygın _____

Date of Approval: _____

© Tuğrul Tekbulut 2006

All Rights Reserved

ABSTRACT

MicroRNAs (miRNAs) are small regulatory RNAs of about 22 nucleotide long sequences that perform important functions such as larval development switches, cell proliferation and differentiation, apoptosis, fat metabolism, control of leaf and flower development. MicroRNA sequences are highly conserved across even unrelated species, a fact which suggests a key role in the evolutionary development. MicroRNAs are transcribed in the nucleus and perform their functions in the cytoplasm by binding to the complementary target mRNAs. MicroRNAs modulate gene expression either by suppressing translation or by mRNA cleavage and degradation. Plant microRNAs bind to their target mRNA on the coding region, almost perfectly, and perform their function by the cleavage of the mRNA, while animal microRNAs, bind imperfectly to their target mRNA, on the 3' UTR region, and perform their functions by suppressing translation. MicroRNAs are discovered by both mutational studies and by computational methods. Hundreds of microRNAs have been cloned and sequenced in several organisms including humans, but to date, only few of them have known functions. The experimental techniques to understand the functions of miRNAs are time consuming and expensive which makes computational methods necessary. The identification of targets of plant microRNAs is straightforward due to near-perfect binding, but the imperfect binding of animal miRNAs to target mRNAs makes the computational target prediction rather difficult. In this thesis a new method is proposed for microRNA target prediction in animals using Constraint Logic Programming. With the established method a package micTar was developed to identify targets in *Drosophila* genome.

ÖZET

MikroRNA'lar (kısaca miRNA) gen anlatımının düzenlenmesinde önemli işlevleri olan, ortalama uzunluğu 22nt olan küçük RNA'lardır. MikroRNA'ların larval gelişiminde, hücre gelişmesinde ve farklılaşmasında, sineklerde yağ metabolizmasında, bitkilerde yaprak ve çiçek gelişmesinde önemli işlevleri keşfedilmiştir. MikroRNA dizilerinin birbirinden çok uzak olan canlılarda bile büyük ölçüde korunmuş olmaları önemli evrimsel işlevleri olduğuna işaret etmektedir. MikroRNA'lar hücre çekirdeğindeki transkripsiyon sonrası sitoplazmaya geçerek, hedefledikleri komplementer mRNA'lara bağlanarak ya mRNA'nın kesilerek yokedilmesiyle, ya da protein translasyonunun engellenmesiyle işlevlerini görürler. Bitki mikroRNA'ları hedeflerine mRNA'ların protein kodlayan bölgesinden bağlanır ve mRNA'yı keserek işlev görür. Hayvanlarda ise mRNA'nın 3' ucundaki kod taşımayan bölgelerine oldukça karmaşık bir şekilde bağlanan mikroRNA'lar protein sentezinin baskılanmasına neden olmaktadır. MikroRNA'lar ya suni mutasyon çalışmaları ya da biyoinformatik yöntemleriyle keşfedilmektedir. Bugüne kadar insan dahil çeşitli canlılardan klonlanan mikroRNA'ların sayısı bini geçmiş olmakla birlikte çok azının işlevleri tanımlanabilmiştir. MikroRNA hedeflerinin biyoinformatik yöntemleriyle keşfedilmesi yönünde son bir yılda yoğun bir çalışma ve yayın olmuştur. Hayvanlardaki bağlanmaların karmaşıklığı biyoinformatik yöntemlerle miRNA hedeflerinin belirlenmesini güçleştirmektedir. Bu tezde hayvanlardaki mikroRNA'ların hedeflerinin belirlenmesinde denenmemiş bir yöntem olan Kısıtlı Mantık Programlama yöntemi denenmektedir. Sözü geçen metodla bir micTar yazılım paketi geliştirilmiş ve *Drosophila* genomuna uygulanmıştır.

to

my lovely ones :

Leyla, Ece and Oya

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to express my gratitude to Doç. Dr. Uğur Sezerman for his encouragement, guidance and support during my studies and for introducing me to the world of small RNAs. Special thanks to Doç. Dr. Pierre Flener for discussing with me the problems in constraint programming and special guidance on the subject. I thank both Dr. Pierre Flener and Dr. Justin Pearson for allowing me to use the edit distance global constraint that they had co-authored. I am indebted to Prof. Neş'e Bilgin who took her time and energy to teach me the secrets of molecular biology all through the summer of 2004, and to my friend, Doç.Dr. Göktürk Üçoluk who helped me to crack some Prolog programming problems. I would like to thank Swedish Institute of Computer Science for donating me the Sicstus Prolog license during the course of this work. Finally, I would like to thank Prof. Dr. Kemal Inan, the Dean of the Engineering and Natural Sciences for his encouragement and support, and all the administrative staff of Sabancı University.

GLOSSARY

microRNA: A small ~22nt non protein coding endogenous RNA which plays an important function in post-transcriptional gene regulation.

siRNA: A small ~22nt interfering double stranded RNA originating from internal or external sources, binds to its target with perfect match and an important role by cleaving it target mRNA .

nt: abbreviation for nucleotide.

bp: abbreviation for base pair.

ORF : acronym for Open Reading Frame.

RNAi: Short for RNA interference. Phenomenon of gene regulation by cleavage of target mRNAs by foreign or endogenous double-stranded small RNA.

Seed : Minimum of 4 nucleotide Watson-Crick pair between miRNA and the target mRNA to the 5'side of miRNA.

Full Seed : 7 or 8 nucleotide Watson-Crick pair between miRNA and the target mRNA to the 5'side of miRNA.

UTR: UnTranslated Region. Regions of mRNA which does not carry protein coding information.

3' UTR: Untranslated regions on the 3' end of mRNA.

miRNP: microRiboNucleoProtein. microRNA-mRNA-Protein complex .

EGFP: Enhanced Green Fluorescent Protein.

Propagation: (Constraint Programming) Elimination of impossible values from the domains of variables.

Reification: (Constraint Programming) A constraint with an attached Boolean variable; utilized to combine complex constraints.

CHAPTER 1

INTRODUCTION

MicroRNAs (miRNAs) small RNA molecules of length approximately 22 nt, encoded in the genomes of plants and animals that seem to play important roles in gene regulation. MicroRNAs regulate gene expression by binding to their matching mRNAs and they modulate the protein translation either by cleavage of the target mRNA or by suppressing protein translation in the ribosome.

Although the discovery of the first miRNA occurred more than a decade ago [1], only recently, the importance of this class of small, regulatory RNAs has been appreciated [2]. Several hundred miRNAs have been cloned and sequenced from mouse, human, *Drosophila*, C. elegans, and Arabidopsis samples and, around 200-300 unique miRNA genes are estimated to be present in the genomes of both humans and mice. The sequences of many of the miRNAs are homologous between species which implies that miRNAs are involved in evolutionally conserved and critical regulatory pathways.

There are different miRNA pathways in plants and animals. In plants, miRNAs tend to be perfectly complementary to their targets which are mostly located in protein coding regions of mRNAs. The plant miRNAs perform their function by cleavage and degradation of mRNA like in siRNA pathway in RNAi [6]. In animals, the miRNA targets are mostly located in non-coding 3'UTR regions and the function is performed by blocking the

translation initiation. Many of the recently cloned miRNAs are found to be differentially expressed in particular cell types which suggest an important function in cell differentiation.

miRNAs are discovered either by cloning methods or by computational methods. Since miRNAs are expressed differentially in space and time, cloning methods will not be able to locate all miRNA expressing genes which makes development of computational methods a necessity. But to understand their function, is even more difficult with experimental techniques and computational methods must be developed. There is an explosion of algorithms developed to find the targets of miRNAs with widely differing results which paradoxically require experimental verification. Since the rules of algorithms are derived from very small number of experimentally known targets of miRNAs, as the number of experimentally known targets is increasing there will be better chances to improve the algorithms.

This thesis is organized in seven chapters and two appendices. A review of miRNA biology is given in Chapter 2. This background enables the reader to understand how the rules for microRNA target findings are derived. The experimental verification of the rules is explained in Chapter 3, Principles of microRNA-Target Relationship. Chapter 4 briefly compares some microRNA target finding algorithms. Chapter 5 gives the details of the approach of this thesis to develop a new method for microRNA target identification. In Chapter 6, the results obtained with the package developed are given, and, they are compared to two other known algorithms. Conclusions and recommendations for further development are discussed in Chapter 7. A comprehensive list of microRNA target finding bibliography is provided. Appendix A is a short introduction to constraint programming. Appendix B gives details how a bioinformatics problem like sequence alignment can be modeled using constraints.

CHAPTER 2

BIOLOGICAL BACKGROUND

Genomics

It is very surprising that miRNAs are overlooked and left undiscovered for many years. One of the reasons might be the "Central Dogma of Molecular Biology" which mainly focuses on the protein coding regions, and, naming the rest of the genome as "junk". The miRNA genomic studies show that these small genes generally exist in regions distant from protein coding regions but sometimes appear in tandem or in the introns of protein coding genes [3]. The miRNAs within the intron sequences do not have their own promoters and transcription factors. They share them with the primary transcript of the host gene.

Since the miRNAs are differentially expressed in different cell types it is not easy to detect them only by cloning [4]. The computational miRNA identification tools have been designed that search for sequences in conserved non protein coding regions that can potentially form stem and loop hairpin precursors. Computational methods have enabled the discovery of many miRNAs which have been later verified experimentally.

miRNA Biogenesis

miRNAs are transcribed as parts of longer RNA molecules [2]. Two RNA polymerases play a role in pri-miRNA transcription [5]: pol II and pol III.

Pol II produces all mRNAs and some non-coding RNAs, and four of the small nuclear RNAs (snRNAs) of the spliceosome, whereas pol III produces some of the shorter non-coding RNAs, including tRNAs, 5S ribosomal RNA, and the U6 snRNA. Naturally, miRNAs processed from the introns of protein-coding host genes are transcribed by pol II. As of today, it is suggested that all miRNA primary transcripts must be capped transcripts which are polymerized by pol II, due to the following observations:

(1) The length of pri-miRNAs are more than 1 kb, which is longer than typical pol III transcripts.

(2) These pri-miRNAs contain long runs of uridine residues, which would prematurely terminate pol III transcription.

(3) Many miRNAs are differentially expressed during development, an observation for pol II but not for pol III transcripts.

(4) When open reading frame of a reporter protein is placed downstream from the 5′ portion of miRNA genes, it leads to a robust reporter protein expression [4].

miRNA and siRNA

miRNAs and siRNAs are two different types of small regulatory RNAs. While miRNAs are endogenous, siRNAs are mostly exogenous processed from foreign double stranded RNA duplexes. miRNAs silence the target gene by binding to the 3' UTR of target mRNA causing suppression of protein synthesis on the ribosome,. siRNAs act like plant miRNAs, targeting the coding region of the target mRNA and perform their function by cleavage of the mRNA3.

miRNAs are generally transcribed from genomic loci distinct from other recognized genes, whereas siRNAs often derive from mRNAs, transposons, viruses or heterochromatic DNA. siRNAs do not form local hairpin structures, they are processed from long bimolecular RNA duplexes or extended hairpins. From each miRNA precursor, only one miRNA duplex is generated while a multitude of siRNA duplexes are generated from each siRNA precursor leading to many different siRNAs. miRNA sequences are conserved in related species, whereas endogenous siRNA sequences are rarely conserved [5,6].

Maturation

After transcription, the long RNA precursor is processed by the dsRNA-specific ribonuclease, Drosha, within the nucleus, into hairpin RNAs of 70-100 nucleotides. (Figure 1).

The hairpin RNAs are transported to the cytoplasm by a protein complex called Exportin; and, there, they are digested by a second, double-strand specific ribonuclease, Dicer, which shaves away the bulb of the hairpin [2,6].

The resultant 17-23nt long single stranded miRNA or siRNA are bound by a ribonucleoprotein complex called RISC (RNA Induced Silencing Complex). After binding to the RISC complex, single-stranded miRNA adapts a conformation that bind to target mRNA which have a significant complementarity. The RISC assembly is mostly comprised of Argonaute family proteins. A range of other proteins are co-purified with RISC which implies that there are different types of RISC.

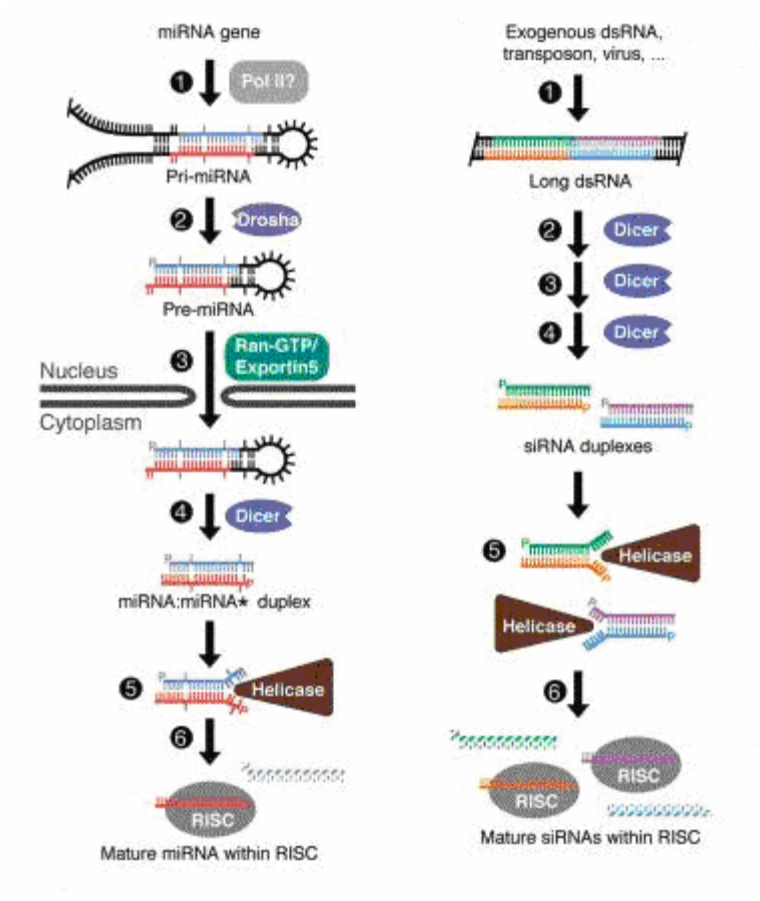

Figure 1 miRNA and siRNA Biogenesis and Maturation
(adapted from Bartel, 2004)

When the miRNA strand of the miRNA:miRNA* duplex (RNA duplex after the cut of the bulb) is loaded into the RISC, the miRNA* is peeled away and degraded. Which strand is chosen by the RISC and which one is degraded is determined by the relative stability of the two ends of the duplex: for both siRNA and miRNA duplexes, the strand that enters the RISC is nearly always the one whose 5′ end is less tightly paired [4].

Functional Mechanisms

MicroRNAs direct the RISC assembly to their target mRNAs to downregulate the posttranscriptional gene expression. If the target section is within the ORF then the mRNA is

cleaved by RISC and digested in the cytoplasm; or, if the target is in the 3'UTR region, the mRNA stays intact, but the functioning of the ribosome is blocked and translation is inhibited.

Plant miRNAs base pair with their targets near perfectly. They are complementary to the transcribed regions of the target gene, while animal miRNAs tend to function as translational repressors by finding their targets in 3' UTR regions of the mRNAs. Hence plant miRNAs generally function by mRNA cleavage and animal miRNAs act as translational suppressors.

Target Selection

Computationally predicted miRNA targets provide lots of insights and hypotheses but they need experimental verification. Majority of computational methods for target identification used evolutionary conservation to distinguish miRNA target sites from the multitude of 3′ UTR segments that score equally well with regard to the quality and stability of base pairing. The cell, on the other hand, cannot use the filter of evolutionary conservation to choose among the possibilities. Also, it cannot be said that miRNAs will bind to the all co-expressed cognate mRNAs. It is very probable that there are other major factors affecting the target specificity. Proteins or mRNA structure could restrict miRNP accessibility to the UTRs. For example, a recently developed algorithm incorporates mRNA structure before searching for the complementary base sequences to miRNA [20]. But there is a limit to generalization; gene knockdown experiments with siRNAs have very high success rates and they are merely based on sequence matching. How proteins or mRNA structure are effecting the recognition of the authentic mRNA targets are not known.

The following figure depicts miRNA and siRNA target relationships. miRNAs bind to the 3' UTR region in a complicated fashion, whereas siRNAs bind to coding region with almost exact sequence match.

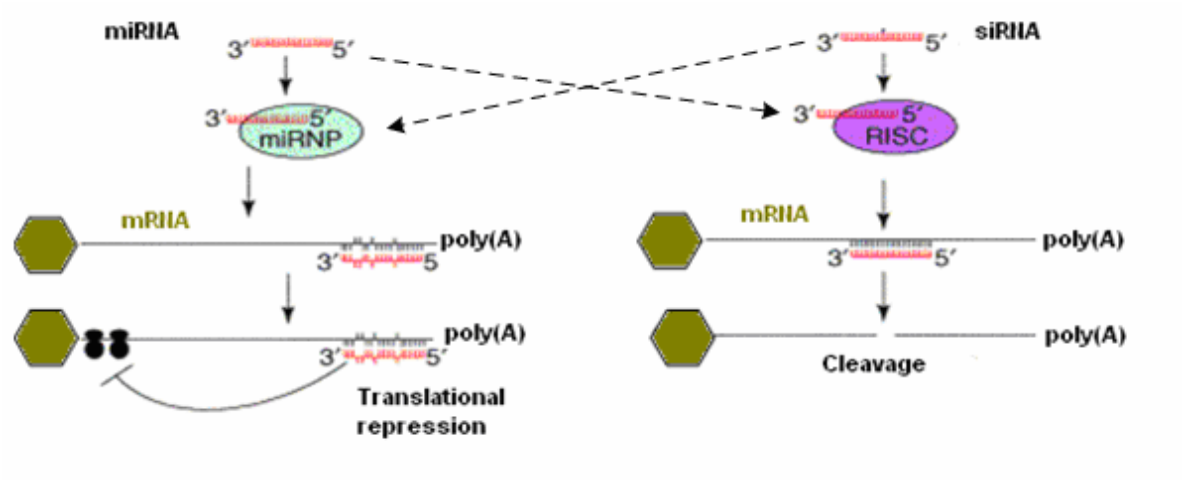

Figure 2 , miRNA and siRNA target selection pathways

A more detailed analysis of the subject is given in the next chapter, as it will help to establish the hypothesis of this thesis.

CHAPTER 3

PRINCIPLES OF MiRNA-TARGET RELATIONSHIP

Bioinformatics algorithms developed for miRNA target predictions are mainly based on:

- Sequence match characteristics derived from the analysis of known targets,
- Minimum Free Energy for the stability of the binding,
- Conservation analysis among related species [7-12].

mRNA folding, geometry and the effect of the interacting proteins are not much incorporated because there is not enough experimental evidence [20]. The methods employed so far were able to catch most of the known targets but also created a multitude of false positives. Two comprehensive experimental and computational attempts have been done to lay down the framework of specificity of target selection [13,14]. The report published in 2005 by EMBL researchers J. Brennecke *et al*. [14] tries to lay down the underlying principles in the miRNA-Target pairing phenomena in animals, based on a comprehensive set of experiments in *Drosophila*. Search strategy developed in this thesis is derived mainly from these findings.

It has been known from the experiments and bioinformatics analysis of the known targets that the 5' side of miRNA plays a more important role in the pairing and in the regulation [4,7,8,9]. No role has been given to the 3' end, although miRNAs are generally conserved over their full length14. J. Brennecke *et al*. did a series of experiments in *Drosophila* wing imaginal disc[14] to observe the repression of an EGFP expressing transgene which contains a single target site for miRNAs in its 3′ UTR. By introducing changes as small as a single nucleotide to the designated target site and measuring the degree of repression by comparing EGFP levels in miRNA-expressing and non-expressing cells, they have been able to understand the characteristics of sequence matching down to a single nucleotide level.

The following pictures adopted from J. Brennecke *et al*. show the effects of the mismatch introducing experiments. In the darker regions, the fluorescence is inhibited by suppression of the translation of the EGFP protein by miRNA action. Less dark or brighter regions are where the miRNA action is less effective or is not observed at all. Figure 3 shows the change in the level of suppression as single nucleotide changes are introduced to the UTR segment matching with the 5' side of miRNA. Figure 4 depicts the analysis to understand the minimum 5' seed size for a functioning target site. Figure 5 depicts the relation between the minimum seed size and multiple hits within the same UTR.

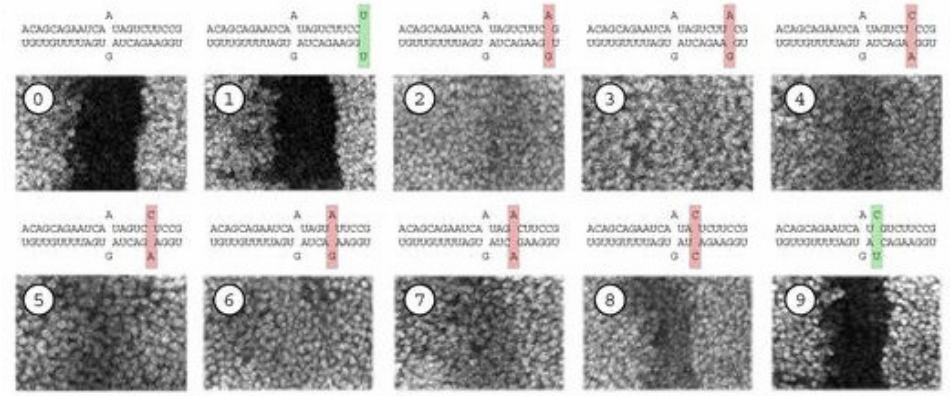

Figure 3, Mismatches in 5' and their effect (Pictures from J. Brennecke *et al*, 2005)

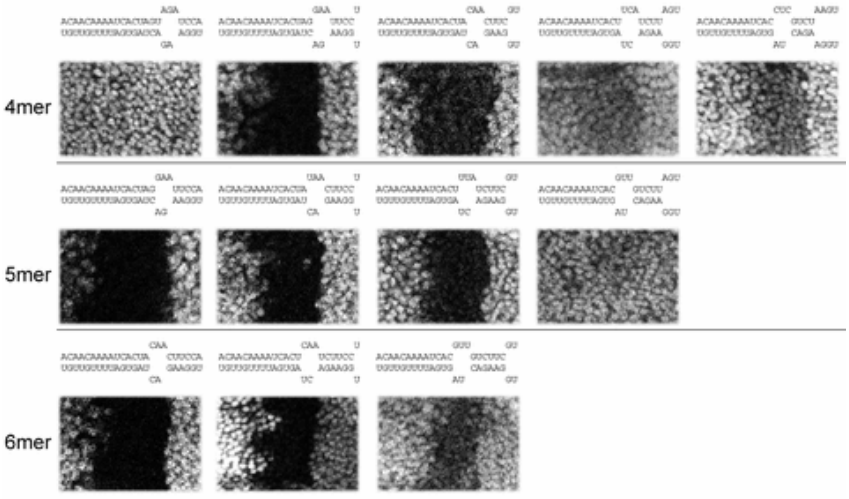

Figure 4, Test for minimum seed for a functional site (Pictures from J. Brennecke *et al*, 2005)

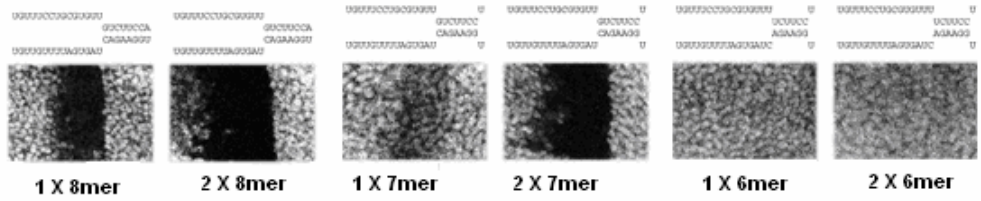

Figure 5 Test for seeds with single and multiple hits (Pictures from J. Brennecke *et al*, 2005)

The major findings of the experiments are summarized below:

- Full binding on the 5' side creates strong repression.
- Any mismatch between 2nd and 8th nucleotide reduces target regulation strongly.
- There has to be a minimum of 4 nucleotide perfect Watson-Crick pair (seed) on the 5' side in any functioning target site.
- Minimum 5' seed size is 7 base pairs if not accompanied by strong 3' pairing.

- Strong 3’ binding does not make a functioning target if not accompanied by the minimum 5’pairing.

- Functioning targets start at positions 1 and 2. Matches at positions 3 and after are less functional.

- 5’ Free Energy is not a determinant of function as some non-functioning targets have more favorable free energies than some functioning targets (conflicts with [12,13]).

- In conformance with the above funding G:U base-pairs in the seed region are detrimental to functioning target.

In other words, (1) complementarity of seven or more bases to the 5′ end miRNA is sufficient for regulation, (2) sites with weaker 5′ complementarity require compensatory pairing to the 3′ end; and (3) extensive pairing to the 3′ end of the miRNA is not sufficient without a minimum seed of matches on the 5′ side.

Functional Categories of Target Sites

J. Brennecke *et al* contributed to the miRNA target terminology by categorizing the functional targets as:

- 5’ dominant sites, (sites that depend critically on pairing to the miRNA 5′ end)

- 3’ compensatory sites (sites that cannot function without strong pairing to the miRNA 3′ end).

The 3′ compensatory group includes seed matches of four to six base-pairs and seeds of seven or eight bases that contain G:U base-pairs, single nucleotide bulges, or mismatches.

5’ dominant sites can be divided into two subgroups:

- Canonical sites (good pairing to both 5′ and 3′ ends of the miRNA)
- Seed sites (good 5′ pairing but with little or no 3′ pairing)

Canonical sites are likely to be more effective because of their higher pairing energy, and may function in one copy. Seed sites are expected to be more effective when present in more than one copy, due to their lower pairing energies. Figure 6 presents examples of the different site types in biologically relevant miRNA targets and illustrates their evolutionary conservation in multiple *drosophilid* genomes.

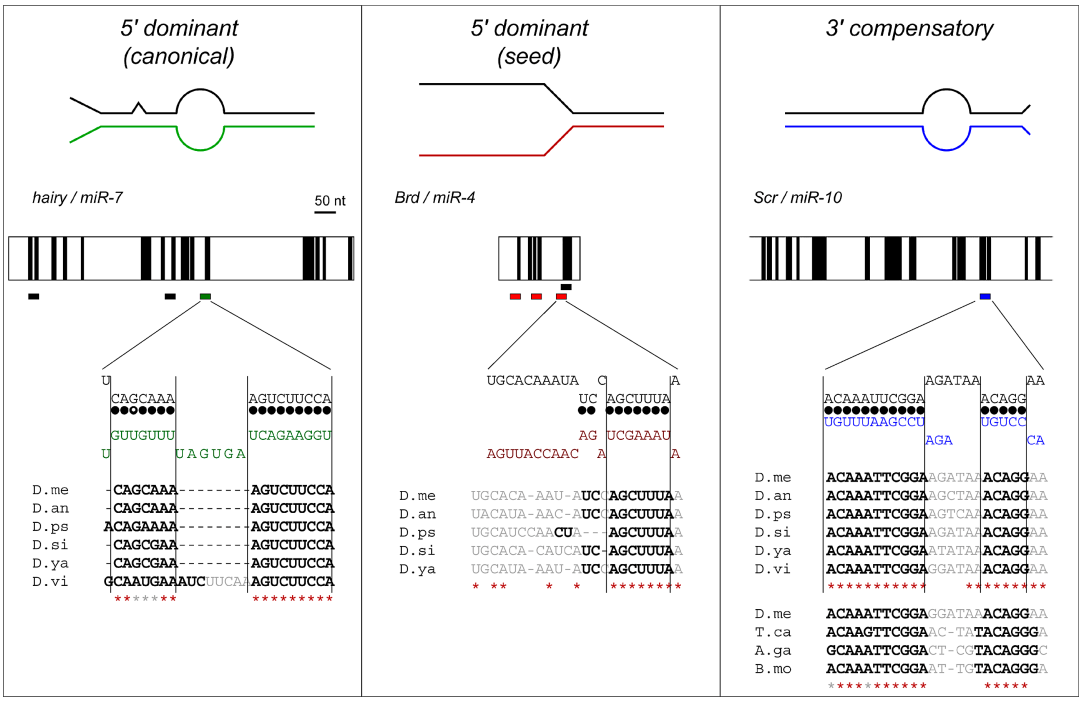

Figure 6, Three Classes of miRNA target site (From J Brennecke *et al* (2005))

Most currently identified miRNA target sites are canonical. The 3′ UTR of *hairy* gene which is active in biological processes like cell proliferation and nervous system development, contains a single site for *miR-7,* with a ninemer seed and a stretch of 3′

complementarity. This site was shown to be functional *in vivo* [10], and it is conserved both in the seed and in the complementarity to the 3′ end of *miR-7*.

The 3′ UTR of *Bearded (Brd),* a gene which is involved in biological processes like notch signaling pathway and sensory organ development is an example to the seed sites, with three sequence elements, known as Brd boxes, complementary to the 5′ region of *miR-4* and *miR-79* 14. All three *Brd* box target sites consist of 7mer seeds with little or no base-pairing to the 3′ end of either *miR-4* or *miR-79*. The alignment of *Brd* 3′ UTRs in Figure 6 shows that there is little conservation in the *miR-4* target sites outside the seed sequence.

The 3′ UTR of the HOX gene *Sex combs reduced (Scr)* which has functions like axis specification, sex comb development and sex differentiation is an example of a 3′ compensatory site. *Scr* contains a single site for *miR-10* with a 5mer seed and a continuous 11-base-pair complementarity to the miRNA 3′ end [10]. The *miR-10* is encoded within the same HOX cluster downstream of *Scr,* and the pairing between *miR-10* and *Scr* is perfectly conserved in all *drosophilid* genomes [14, 21].

CHAPTER 4

COMPUTATIONAL APPROACHES TO MiRNA-TARGET PREDICTIONS

It is not possible to identify all the targets of all miRNAs with long and cumbersome experimental techniques; computational approaches have to be employed. Computational approaches have been successful in plants, where known target sites are almost perfectly complementary to miRNAs; 4 in animals, however, the miRNA:mRNA base pairing is not perfect and this creates a challenging computational problem.

miRanda

Of the packages and algorithms developed up to now, the most widely used, referenced and frequently updated package is miRanda developed by John Enright *et al*. presented in their manuscript "MicroRNA Targets in Drosophila" [9]. miRanda is free and open source, with its newer versions, is still being used today to predict miRNA targets in nematodes, flies and mammalians. miRanda is available at:

http://www.microrna.org/

Recently, microRNA Registry [15], hosted and managed by Sanger Institute, started to present the candidate targets for miRNAs in several genomes, computed by miRanda:

http://microrna.sanger.ac.uk/targets/v3/

miRanda is no different than the previous apporaches to the problem:

- Sequence-matching to assess whether two sequences are complementary,
- Free energy calculation to estimate the energetics of this physical interaction,
- Evolutionary conservation as an informational filter.

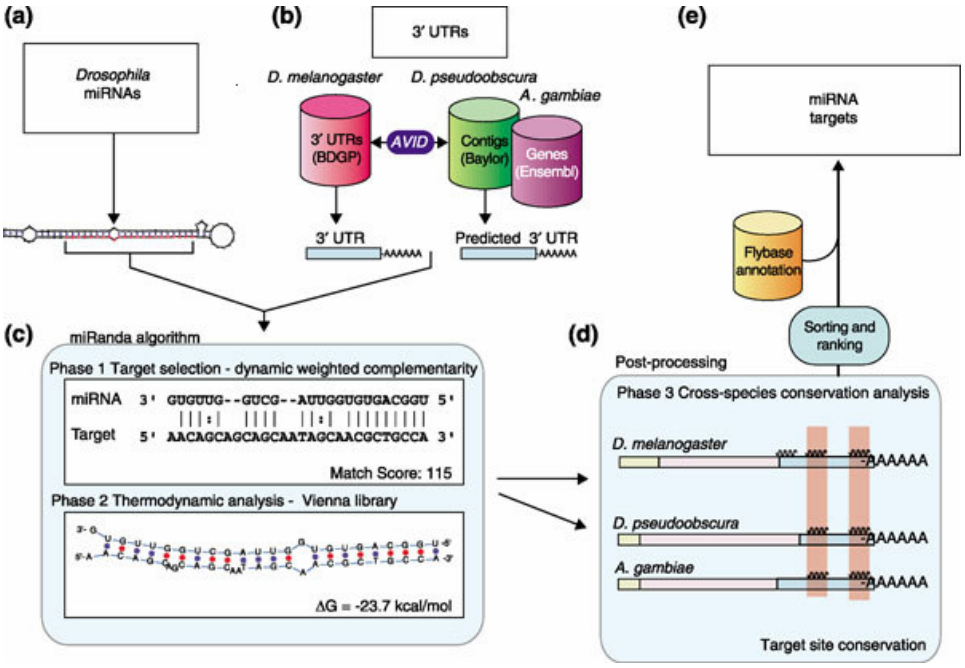

Figure 7, miRanda Target Prediction Algorithm (from J. Enright *et al*.,2003)

Sequence match

Using a dynamic programming algorithm, miRNA sequences are searched through the 3' UTRs of Drosophila melanogaster genes for possible complementarity. The algorithm takes into account G-U wobble pairs, allows some insertions and deletions and, uses a weighting scheme that rewards complementarity at the 5' end of the miRNA. The result is a score (S) for each detected complementarity match between a miRNA and a potential target gene.

Free energy calculation

For each match, the free energy (ΔG) of optimal strand-strand interaction between miRNA and 3' UTR is calculated using the Vienna package [26].

Evolutionary conservation

The conservation of predicted miRNA-target pairs in related organisms is an important additional criterion in miRanda. A miRNA target pair is considered to be conserved across species if a specific miRNA independently matches orthologous UTRs in two other species and show more than a specified threshold of nucleotide identity with each other.

PicTar

One of the latest package fror miRNA target prediction is PicTar developed by Grün *et al*. at Rajewsky Lab at NYU [21]. PicTar starts with pre-aligned RNA sequences (typically 3' UTRs) from several related or non-related species. It is obvious that the package takes conservation as the main indicator of a functioning target. One of the distinct features of the package is that it can locate combinatorial targets for co-expressed microRNA sequences.

The program nuclMap locates all perfect seed (length 7, starting at position 1 or 2 of the 5' end of the microRNA) and imperfect seed in 3' UTR sequences. At the seed matching positions, the free energy of binding is calculated along ~22nt UTR segments, and, those positions that survive the optimal free energy filter and fall into overlapping positions in the alignments for all species are categorized as "anchors". If a 3' UTR multiple alignment has a minimal (user-defined) number of anchors, each UTR in the alignment will be scored by the central PicTar maximum likelihood procedure.

Scores for individual UTRs in an alignment are combined to obtain the final PicTar score, which can be used to obtain a ranked list of all sets of orthologous transcripts. Scores of all segmentations of the RNA sequence (3' UTR) into binding sites and background

sequences are listed. PicTar computes a maximum likelihood score using Hidden Markov Model that the RNA sequence is targeted by combinations of microRNAs from the search set.

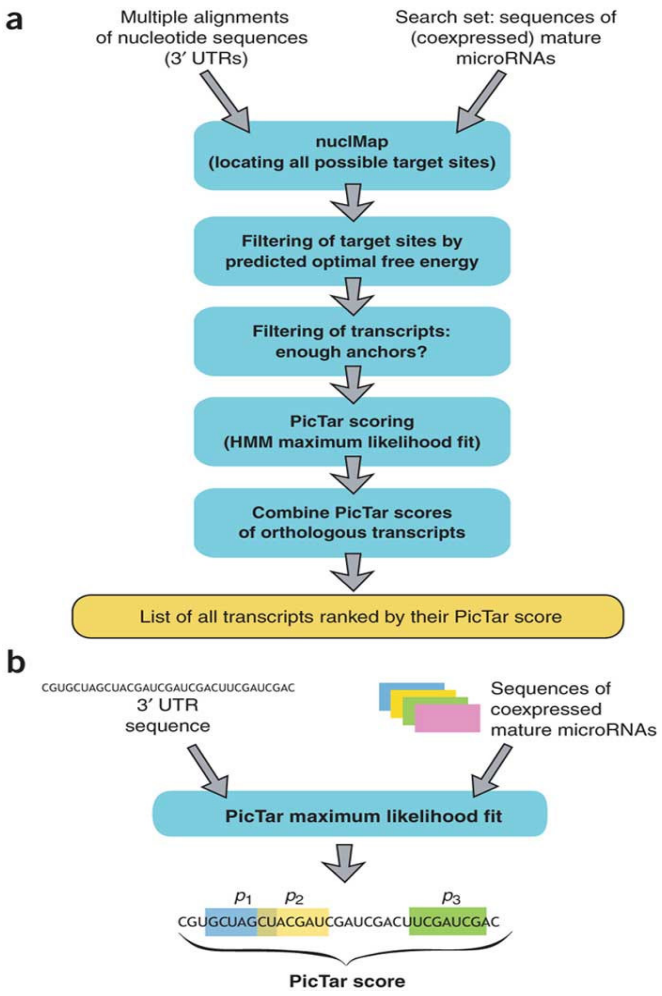

Figure 8, PicTar Algorithm , from Grün *et al*. (2005).

Incorporating Structure of mRNA

One radical deviation from the above sequence based approaches is by H. Robins *et al*. [20] where they searche binding sites in a folded mRNA secondary structure. The algorithm consists of four parts :

- Look for 7 nucleotide matches from miRNA 5' side,
- Calculate the overall matching score with the target 3' UTR sequence,
- Incorporate the 3' UTR secondary structure,
- Combine the scores for multiple sites in targets.

The hypothesis behind this algorithm is that single stranded miRNAs can only bind to the free bases of mRNA that are not base-paired in the folded structure. This approach dramatically reduces the number of candidate targets.

The downside of this approach, however, is that RNA folding is time consuming, and the target finding is done on a single UTR at a time. The other drawback is that contribution of miRNA 3' is not considered at all, and the only targets with 7 nucleotide seeds are searched as in PicTar. The writers report that they have almost no correlation with the results of miRanda algorithm. One interesting note about their results is the gene *reaper* which has been experimentally verified target for *mir-2a* by the work of Stark *et al*. [10], appears as the top scoring candidate when mRNA folding structure is considered, it drops to 25th when only sequence match is considered.

CHAPTER 5

PROPOSED METHOD AND ALGORITHM FOR micTAR

Constraint programming is a new high-level paradigm developed for solving complex combinatorial satisfaction and optimization problems. Such problems are solved searching through a very large search space to find a solution or the optimal solution. In the constraint programming paradigm, constraints are used to limit the search as much as possible. Hence, the two main components in a constraint programming system are the constraint solver and the search engine which implements some strategy, such as backtracking, for exploring the search space.

Constraint Logic Programming (CLP) is the simplest and most elegant approach to constraint programming. This is because the logic programming paradigm is well matched with the constraint paradigm, as both paradigms are based on the fundamental concept of a relation. The high-level nature of CLP programs is ideal for fast program development and experimentation, and the resulting programs are concise, easy to maintain and readily extendible. Another advantage of CLP languages is that they inherit the simple declarative semantics of logic programs. This means that they are suitable for powerful, high-level program transformations and optimizations which can dramatically improve performance. In this thesis we adopted Constraint Logic Programming to solve the miRNA-Target Problem using the tool Sicstus Prolog developed and supported by Swedish Institute of Computer Science. The version utilized is 3.12.5. Sicstus Prolog is a ISO Prolog Compliant Prolog language, but it is also a host to a multitude of constraint solvers [33].

As many other genomics problems, miRNA-Target problem is a discrete problem over finite domains, i.e., there is a limit to the size of all the different discrete values a variable can take. These types of combinatorial, finite domain problems are handled by a finite domain constraint programming approach. Finite Domain Constraint Problems are also called Constraint Satisfaction Problems.

A constraint satisfaction problem is solved by:

- Declaration of variables:

$$X_1,.....,X_i,...., X_n,$$

- Domain declarations for these variables:

$$D_1,....,D_i,.....,D_n,$$

- Posting of Constraints:

$$C_1(X_i..X_j),\ C_2(X_k..X_l),..,\ C_k(X_m.X_n),\qquad i,j,k,l,m\ \text{in}\ \{1,..n\}$$

The solver attacks the problem by several search methods available in Sicstus Prolog, like starting from the variable with the smallest domain, or starting from the most constrained variable, or going from small values to large values etc. Upon propagation, which eliminates the impossible values for each variable, the Constraint Solver enumerates different values for each variable to find the solutions which satisfies all the constraints. If the optimal solution is required, Sicstus Constraint Solver employs the Branch and Bound algorithm which maximizes or minimizes an objective function. A brief introduction to constraint programming is provided in Appendix A.

Modeling of The Target Recognition Problem for MicTar

The position dependencies in the binding characteristics of miRNA to its target had been discussed in Chapter 3. In this thesis, these positions are counted from miRNA 5' end, and the following naming convention is used:

- 5' Side : nucleotides from 1 to10.
- 3' Side : int(miRNASize/2), int : integer part.
- Seed : At least 4 nt perfect match at 5' side with start positions 1 or 2.

The functionality contributing and noncontributing parts of the binding are depicted in Figure 9.

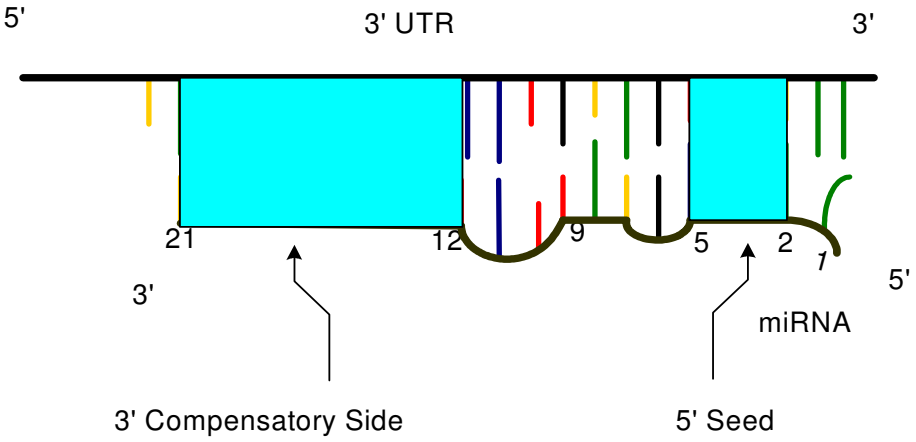

Figure 9, Typical miRNA-3'UTR target binding

Table 1 summarizes the experimentally verified rules for a functioning target presented in Chapter 3. The constraints of micTar are derived from this table. Conservation analysis is not used as a search criterion; instead it is used as a post processing filter.

MicroRNA Target Prediction Constraints						
StartPos	SeedLength	3' pairing	F.Energy	Copies	Conservation	Regulation
2	4	High	>Feth	1	Yes/No	Good
1	5	High	>Feth	1	Yes/No	Good
2	6	High	>Feth	1	Yes/No	Good
3	4..6	High	>Feth	2	Yes/No	Good
1	7	High	>Feth	1	Yes/No	Good
1	7	N/A	-	2	Yes/No	Good
2	7	High	>Feth	1	Yes/No	Good
2	7	N/A	-	2	Yes/No	Good
1	8	N/A	-	1	Yes/No	Good
1	8	High/Low	-	1	Yes/No	Good
2	8	High/Low	-	1	Yes/No	Good
1	9	High/Low	-	1	Yes/No	Good
2	9	High/Low	-	1	Yes/No	Good
1	10	High/Low	-	1	Yes/No	Good
2	10	High/Low	-	1	Yes/No	Good
3	10	High/Low	-	1	Yes/No	Good

Table 1 Constraints of Target Prediction in MicTar

Analysis of the Table 1 starts at the top with the conditions of strong 3' binding. The seeds of at least length of 4 with starting position 2 can function if supported by a strong 3' pairing. On the other hand, seeds of length 4, starting position 1 are not functional and they do not appear in the table.

Since this "at least" condition helps us to contain the 5 nucleotide match between positions 1 and 5, a constraint which states "perfect matches of at least 4 nucleotides with start position 2, and with a strong 3' binding" will give almost all 3' compensatory sites.

Perfect exact matches of 7 nucleotides starting position 1 with no 3' binding are considered to be a target, if they work in tandem with at least as two copies within the same UTR. Seed sites starting position 2 with at least 7 perfect matches are always considered to

be functional. 10 nucleotide long exact Watson-Crick pairs starting at position 3 are also functional targets.

Some experimentally verified targets are known to contain G:U pairs and some bulges. This was one of the major reasons that early bioinformatics approaches mainly searched for strongest bindings based on free energy calculations. The contribution of G:U pairs to the free energy, lost its importance, as mentioned in Chapter 3, and, in micTar they are only tolerated rather than searched for. The allowable conditions for G:U pairs, and bulges/mismatches are shown in Table 2.

StartPos	SeedLength	Bulge/G:U	3' pairing	F.Energy
2	8	1	H/L	-
1	9	1	H/L	-
2	9	1	H/L	-
1	10	1	H/L	-
2	10	1	H/L	-
3	10	1	H/L	-

Table 2, Allowable G:U pairs or bulges

Implementation of the model in Constraint Logic Programming

Since the conditions for a functioning target are stated in terms of match positions of miRNA, in Table 1, it can spontaneously be inferred that the variables will be the positions of the miRNA, and the database to be searched will be the 3'UTR sequence. The size of the search space of such a problem is the Cartesian product of the size of the domains of the individual variables. As the average size of miRNA is 22nt, the size of the search space will be ≈ (Genome Size)^{22} . Fortunately, though, the positions of the miRNA follow an order (i.e. no twists are allowed in the binding):

$$P_{n,} ..< P_i <,.P_2<P_1 ,$$

a constraint which breaks the symmetry of the search, and the size of the computation reduces to N x m. Furthermore, since all position variables can be expressed with their constant distance to P_1, the problem becomes a linear search problem with one variable: P_1.

Again the search space is still not small, for a relatively small organism like Drosophila; the total size of the 3'UTR sequences is $\approx 7x10^6$. A further enhancement can be made by noticing the importance of the fourmers, as the functioning targets must have at least one perfect fourmer to the 5' side of the miRNA. So the first 2 fourmers starting position from 1, and the first 2 fourmers starting from position 2 are taken into account. The 3 'UTR positions not matching 4mer1 or S4mer1 or T4mer1 (Third 4mer from 5' side) can be excluded from the search space.

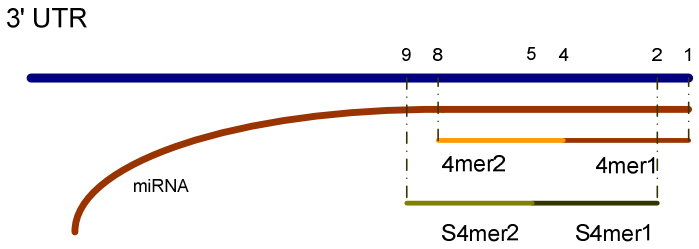

Figure 10, Important fourmers of the miRNA

Since we have to search this space for all the miRNAs of an organism, and, since we are only interested in fourmers and the single nucleotides changes around those fourmers, it is a very good investment to create the fourmers map of the genome.

Pre-Processing into 4mer Arrays:

Since the Genome is written in 4 letter alphabet, there are $4^4 = 256$ different types of fourmers, a number which can be expressed in one byte. Thus a genome can be expressed as a list of fourmers without increasing the data size. Position lists of all 256 fourmers can be

processed out from the genome to speed up access. In this thesis, these lists of fourmers are named "4mer Arrays" with inspiration from the suffix arrays.

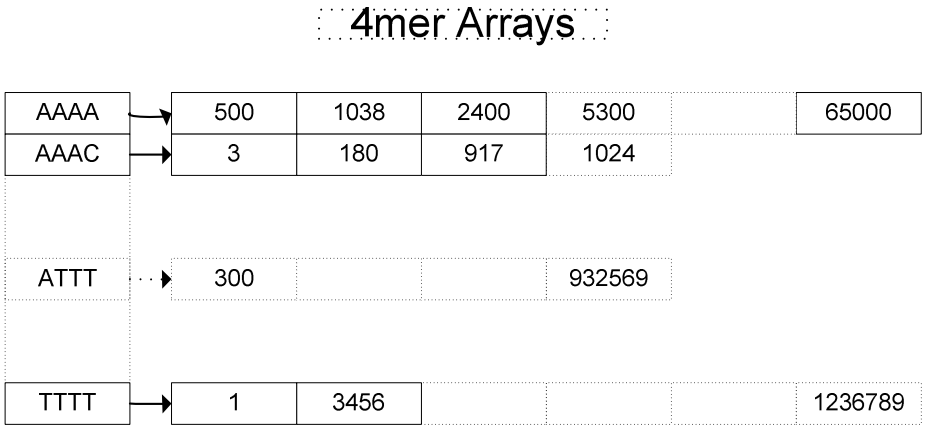

Figure 11, 4mer Arrays Data Structure

The project is run on Drosophila Genome to be compatible with most of the referenced work that had been done on this species. The 3'UTR sequences are downloaded from:

http://flybase.bio.indiana.edu/annot/download_sequences.html

The microRNA sequences for Drosophila are downloaded from microRNA Registry [15]:

http://microrna.sanger.ac.uk/sequences/

The flow of the algorithm is depicted in Figure 8.

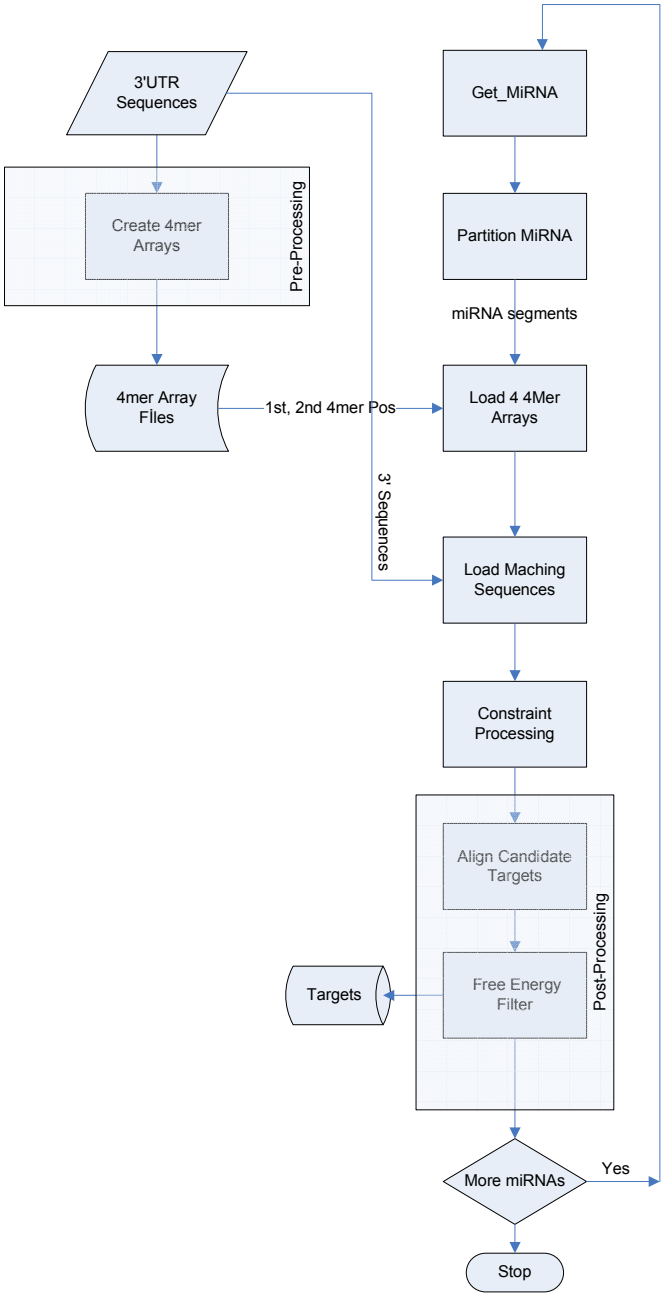

Figure 12, MicTar Algorithm

Get MicroRNA and Partition MicroRNA:

MicroRNA is read and partitioned for the important search positions. There are 4 important fourmers: 4mer1, 4mer2, S4mer1, S4mer2, as shown in Figure 10. Half of the miRNA from the 3' side is partitioned as 3Prime.

Load 4mer Arrays:

The four fourmers which exist in the miRNA are loaded into the memory.

Load Sequences:

A miRNA size sequence is loaded from the positions of first fourmers at position 1 or 2. The position is corrected for the first fourmer starting at position 2 (S4mer1 in Fig. 6) and the target sequence is loaded from Pos+1.

Constraint Processing:

The constraint processor locates the candidate targets by constraint propagation. There are several sets of constraints in the program to locate the different type of targets some of which are given below:

The constraints for eightmer full seed targets:

If we call our variables as P, then

$P4_1$ #=P, $P4_2$ #= P-4,

$PS4_1$ #=P-1, $PS4_2$ #=P-5, *(Eq. 4.1)*

Then a constraint for an, eightmer seed target between positions 1 to 8 is:

$P4_1$ in 4_1PSet #/\ $P4_2$ in 4_2PSet #<=> $Seed_{18}$, *(Eq. 4.2)*

Then a constraint for an eightmer seed target between positions 2 to 9 is:

$PS4_1$ in $S4_1$PSet #/\ $PS4_2$ in $S4_2$PSet #<=> $Seed_{29}$, *(Eq. 4.3)*

These two reified constraints will select a target side by :

$Seed_{18}$ + $Seed_{29}$ #>= 1. *(Eq. 4.4)*

The reification of two constraints helps to implement the logic operator OR between the two constraints in *4.2* and *4.3* .

The constraint for a typical 3'compensatory site is:

$PS4_1$ in $S4_1$PSet #/\ editdistance (3',{3'UTR}) #=< *d* . *(Eq. 4.5)*

The constraint in (*Eq. 4.5)* states that a matching fourmer UTR segment with the miRNA fourmer at position 2 is a target, if and only if the 3' UTR sequence matching the miRNA 3' Prime side has an edit distance less than or equal to a predefined distance *d*. The editdistance is a user defined global constraint.

Since this problem was reduced to a single variable problem, CLP might be seen overdoing. On the other hand, with the help of CLP the code has become simpler, mathematically elegant and easy to maintain. During the course of the project, it has become necessary to change the program, many times, as more in-depth biological knowledge was obtained from published new research, from communication with the researchers or just by rereading the existing material. Thanks to the declarative nature of the program, it was

sufficient just to change, add or delete the constraints rather than to write down full procedures to describe the changing physical situation. For example a target which has one mismatch in the second fourmer is formulated by the following constraint:

($PS4_1$ in $S4_1$PSet #/\ hamming ($S4_1$, UTR_{S41}) #=<1)#/\ (editdistance (3',$UTR_{3'}$) #=< *D (Eq. 4.6)*

Where UTR_{S41} and $UTR_{3'}$ are UTR segments matching with the first fourmer at position 2 and the 3' part of miRNA, respectively.

Align Candidate Targets

After the candidate targets are selected miRNA 3' side is aligned with the corresponding 3'UTR sequences. For this alignment a special CLP algorithm was devised. The result of the alignment is the input to the special Free Energy Calculation algorithm which is based on the information content of the aligned sequences. The algorithm of sequence alignment with CLP is presented in Appendix B.

Free Energy Filter

Gary Stormo *et al*. [19] propose a method of calculating the free energy of binding site based on the information content of the alignment. It is assumed that the total binding energy is the sum of independent contributions at each position and the good targets must have lower free energies and higher information content. Relying on this information a filter was implemented relying on the information content of the alignment. Upon the discussion in Chapter 3, free energy considerations are limited to the 3' side of miRNA to look for strong bindings to make 5'side binding functional. The information content of a binding site is defined as:

$$I_{\text{seq=}} \sum_{j}^{n} \sum_{b}^{n} f(k,j) \log_2 \frac{f(b,j)}{p(b)} = \frac{\langle \Delta G_s \rangle}{-RT \ln 2} \qquad \textit{(Eq. 4.6)}$$

where $f(b,j)$ is the probability of base b being at position j,

$p(b)$ is the probability of base b in the whole genome.

CHAPTER 6

RESULTS AND DISCUSSION

micTar is very fast and it has been very successful in locating the known targets that all the competing algorithms are checked against [9,11,12,14,16]. Thanks to the unique data structure of the 4mer Arrays, all full seed targets (8 and more) for *dme-mir-bantam* is reached in less than 4 seconds on 1.8GHz notebook computer with 2GB RAM.

Check for known targets

The program is run for different miRNAs to check whether it can locate the experimentally verified targets shown in Chapter 3, Figure 3.

1) *Bantam* targets

dme-mir-Bantam : UGAGAUCAUUUUGAAAGCUGAUU

UTR Pos	Gene ID	Gene Start	Gene Stop	Target Site
6624732	CG5123-RA-u3	6623719	6625987	TGGAATGCACATTAATGATCTCT
6625442	CG5123-RA-u3	6623719	6625987	AATTAGTTTTCACAATGATCTCG

Table 3, *bantam* hits hid (*wrinkle*) gene 3'UTR in two points.

The *head involution defective hid* gene with the ID CG5123-RA had two hits with 9 nucleotide perfect seeds on the 5' side. This gene is a very well known target and strongly regulated by the microRNA *bantam.* This site is a good example of a canonical target site.

The algorithm is tested to find other canonical sites for *bantam.* There is no other canonical site with 3' edit distance 2, 3 and 4. For a relatively mild constraint for the 3' side, i.e. an edit distance of 5 we get 11 target sites with one another site for CG5123 *hid* gene. Since the 5' site is perfectly bound with a 9mer, according to the principles set in Chapter 3, all these sites must presumably be functioning targets. These results are shown in Table 3, with comparison to two other target prediction algorithms.

Gene ID	Target UTR Site predicted by MicTAR	MiRanda	PicTar
CG31647-RB-u3	CCATCTCCTTGGCCATGATCTCG	NO	NO
CG31647-RA-u3	CCATCTCCTTGGCCATGATCTCG	NO	NO
CG6618-RB-u3	AAATGTGTTATTTAATGATCTCT	YES	NO
CG6618-RA-u3	AAATGTGTTATTTAATGATCTCT	YES	NO
CG6575-RA-u3	ATTTACTTTGTGTCATGATCTCA	YES	YES
CG15316-RA-u3	GTCATATCTTTGTCATGATCTCC	NO	NO
CG15316-RB-u3	GTCATATCTTTGTCATGATCTCC	NO	NO
CG12372-RA-u3	GAGCATTGTTCTTGATGATCTCC	YES	NO
CG11714-RA-u3	AATAAATAATACAAATGATCTCG	YES	NO
CG5123-RA-u3	AATTAGTTTTCACAATGATCTCG	YES	YES

Table 4, some canonical sites for *dme-bantam* in Drosophila

As can be seen on the Table 3, the results are highly divergent among the compared packages. PicTar and miRanda are less in agreement with each other than they are with micTar. This is indeed noted by N. Rajewsky [22], -author of PicTar -, in a very recent paper that compared the results of different algorithms and approaches to the target recognition problem. In the above example of Table 3, MicTar seems to be more in agreement with miRanda than PicTar. The reason for this is that all those targets are canonical, and, the algorithm of miRanda looks for overall sequence similarity 9, and micTar,

in this case, is also weighing the overall similarity of the sequences. It can be observed that small differences in algorithms create very divergent results [22].

The other target that the three approaches agree on is the transcript of CG6575, the *gliolectin* (*glec)* gene. *Gliolectin* has functions like cell adhesion and nervous system development. CG 6618 is the *Patsas* gene which has functions in cell proliferation and sensory perception. CG12732 is the *spt4* which has functions like RNA elongation, chromatin assembly or disassembly, non-covalent chromatin modification and positive regulation of transcription [23]. CG6618, CG12372 and CG11714 are all in agreement with the results of miRanda but not with PicTar. CG31647 and CG15316 are only reported by micTar, and the molecular function and the biological process that they are involved are not known [23]. The folding of some of the *bantam* targets located by micTar are shown in Figure 12.

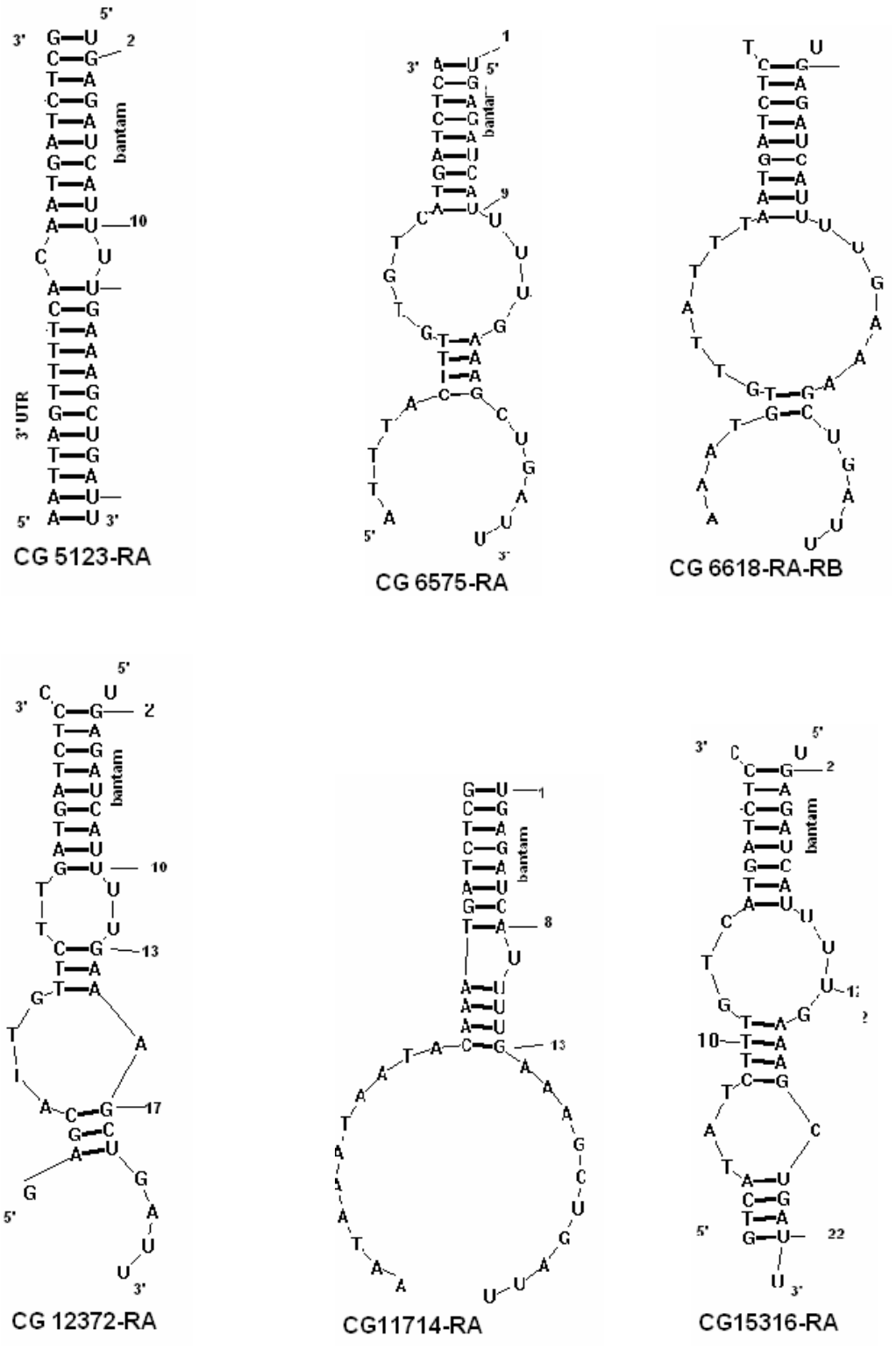

Figure 13, RNA folding of the sequences of Table 1 by RNAStructure program v4.2 [25]

2) Canonical targets for *mir-7*

The canonical targets of mir-7 are searched. The *hairy* gene, CG6494 is hit with the search parameters: start point =1, seed length= 8, 3'edit distance =6. All other transcripts in Table 4 are hit earlier with smaller edit distances. Here micTar results are 78% in agreement with both miRanda and PicTar. This is no surprise that canonical targets are easier to identify in all algorithms.

dme-miR-7: UGGAAGACUAGUGAUUUUGUUGU

Gene ID	Target UTR Site	miRanda	PicTar
CG6555-RA	ATGGCAACATTTCAAGTCTTCCA	YES	NO
CG10379-RA	CGAACCCAAATGCTTGTCTTCCA	YES	YES
CG8346-RA	GCAACAAGATCCGTTGTCTTCCA	YES	YES
CG15797-RA	AAAACAATCGTTGGGGTCTTCCA	YES	YES
CG16700-RA	CAGAAAATAGCCGAAGTCTTCCA	NO	NO
CG10444-RA	AGCGACCAAAACAGAGTCTTCCA	YES	YES
CG6494-RB	AGCAAATCAGCAAAAGTCTTCCA	YES	YES
CG6494-RA	AGCAAATCAGCAAAAGTCTTCCA	NO	YES
CG12487-RA	TTTAAGAAAATCATTGTCTTCCA	YES	YES

Table 5, Some Canonical Targets for *dme-mir-7*

3) Seed target for *mir-4*

The seed target example for *dme-mir-4* is searched with start point =1, seed length =8. The *Brd* gene CG3096 was located with the parameters start point =2, full first 4mer, 1 hamming distance at the fourth position of the second fourmer. *Brd* has three 7mer seed target by *mir-7* in its 3' UTR.

Gene ID	Target UTR Site
CG3096-RA-u3	CCACTTTCCAATCAGCTTTAA
CG3096-RA-u3	CATCATCCGCAACAGCTTTAA
CG3096-RA-u3	TGCACAAATATCCAGCTTTAA

Table 6, *Brd*, Seed target of *mir-7*

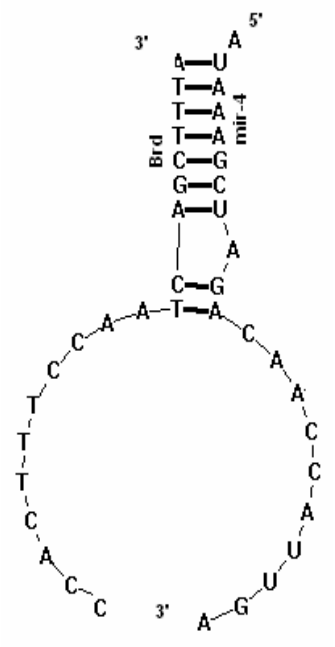

Figure 14, *Brd* and mir-4 (folded by RNAStructure [25])

4) *mir-10* and 3'compensatory targets

As the example of a known 3' compensatory site, the *Sex combs reduced gene* (*Scr)* as a target for *mir-10* is searched. micTar could not locate the *Scr* gene, CG1030 with all the possible parameters of the program. Although it was mentioned that sites starting at position

3 are not effective, it was decided to include the third fourmer from the 5' side, in the searches. The program was modified to consider a longer portion of 3'UTR when comparing with the mirRNA 3' end. The target UTR segment to be analyzed is taken 20% longer than the length of miRNA. The new version found the gene *Scr*, i.e. CG1030 as a target of *mir-10* with the parameters start point=3, and 3' edit distance =2.

Position	Gene ID	Target UTR Site
1317261	CG1030-RA-u3	AACAAATTCGGAAGATAAACAGGAA
1319523	CG1030-RB-u3	AACAAATTCGGAAGATAAACAGGAA
1321785	CG1030-RC-u3	AACAAATTCGGAAGATAAACAGGAA
4067045	CG12237-RA-u3	ACAACTTCGGAGGTGTGCCCAGGAC
6019196	CG33556-RA-u3	ACAATTTCGAATTTCTAAGCAGGAT

Table 7, 3' Compensatory targets for *mir-10*

As can be seen from these examples, micTar algorithm has been very successful to locate the experimentally known miRNA targets in Drosophila genome.

False positives

It must have been noted that some of the experimentally verified targets did not appear in the results without loosening the constraints. Loose constraints will increase the number of false positives which means the set of constraints used to locate the above targets are incomplete. One remedy to this problem is to find more constraints by examining physical situation or to add some more post processing like evolutionary conservation. From the discussion of Chapter 3, it is known that strong 3' binding is necessary to hold seed targets in its place. Up to know, only 3' edit distance was used to impose this constraint. The free energy of 3' binding and its effectiveness as a filter in removing some these possible false positives will be analyzed below. The experimentally verified functions for miRNAs like cell growth, development and apoptosis are the pathways that are under strong evolutionary

selective pressure. An evolutionary conservation filter can be very useful in removing some of non functional target candidates.

3'Free Energy Filter

A free energy filtering program was implemented in CLP paradigm. 5' parts of target UTR sites are aligned with miRNA 3' side. The algorithm of the CLP alignment program was provided in Appendix B. The free energies of the aligned sequences are calculated according to information content of the alignment as expressed in (Eq 4.6). The constraint for the filter is given as a percentage of the free energy of the perfect matching sequence with the mirRNA 3' sequence. For example, when a free energy filter is applied to eliminate the 3' alignments with less than 60% of the free energy of the perfect alignment; all the candidate targets of Table 6, other than *Scr*, CG1030 are perfectly eliminated. The results of this filtering and the folding of miRNA with the target UTR are shown in Table 7, and Figure 11, respectively.

Position	Gene ID	Target UTR Site
1317261	CG1030-RA-u3	AACAAATTCGGAAGATAAACAGGAA
1319523	CG1030-RB-u3	AACAAATTCGGAAGATAAACAGGAA
1321785	CG1030-RC-u3	AACAAATTCGGAAGATAAACAGGAA

Table 8, Results of Table 6 after 60% 3' Free Energy filter is applied.

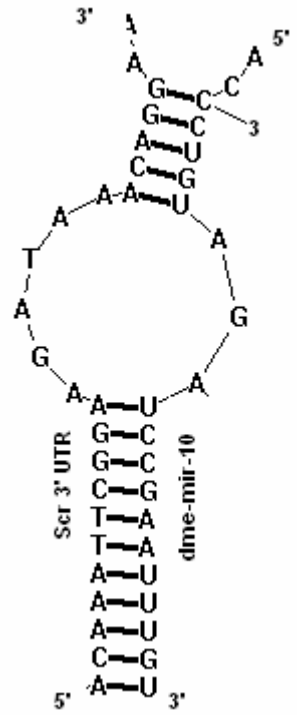

Figure 15, Folding of mir-*10* with *Scr* 3' UTR

Free energy filter has shown its effectiveness in removing many of would-be false positives. It also showed that 3' alignments with the same edit distance may have very different binding free energies. Although the method developed in this project does not look at 3' energies in full seed targets, it could be a good idea to check the 3' free energies just to rank them according to their effectiveness. It was shown that the seed targets accompanied by

higher 3' free energies are more effective in functioning as single copy [14]. This result caused a rethinking of some previous results and it was decided to use the free energy criterion is applied whenever the seed length is less than 7.

The following table is the results of bantam searched with a fourmer seed starting at position 1 and one mismatch in the second fourmer.

Position	Gene ID	UTR Target Site
33973	CG11490-RA-u3	AAATTAGTTCTCGTGCCTGTGAACTCA
726829	CG12292-RA-u3	ATATCACCTGCAATCACTTTCATCTCA
1060109	CG9339-RB-u3	AATTGTTTTCTATCTGAATTGTTCTCA
1062529	CG9339-RE-u3	AATTGTTTTCTATCTGAATTGTTCTCA
1064949	CG9339-RA-u3	AATTGTTTTCTATCTGAATTGTTCTCA
1067369	CG9339-RD-u3	AATTGTTTTCTATCTGAATTGTTCTCA
1069789	CG9339-RG-u3	AATTGTTTTCTATCTGAATTGTTCTCA
1072209	CG9339-RH-u3	AATTGTTTTCTATCTGAATTGTTCTCA
1208110	CG12163-RA-u3	TAATCATTTCAGACATCTGTAATCTCA
1208500	CG12163-RB-u3	TAATCATTTCAGACATCTGTAATCTCA
1654805	CG10097-RA-u3	TAATGAGTTTGTTCTTGATGGATCTCA
2225774	CG5740-RA-u3	AATCAAATCGCTCAAAGCTTGAACTCA
2226055	CG5740-RB-u3	AATCAAATCGCTCAAAGCTTGAACTCA
2837738	CG2041-RA-u3	AAACGCTATTGATATATATTGCTCTCA
3162888	CG12179-RA-u3	ATTGATATTTTATTGATTATCATCTCA
3163638	CG12179-RB-u3	ATTGATATTTTATTGATTATCATCTCA
3343011	CG1435-RA-u3	AATCGGCCGCCGAGGGCGATGACCTCA
3343939	CG1435-RB-u3	AATCGGCCGCCGAGGGCGATGACCTCA
5436902	CG13521-RB-u3	AATCAGTCTAGGAACTGAGTGAACTCA
5438931	CG13521-RA-u3	AATCAGTCTAGGAACTGAGTGAACTCA
6235260	CG8107-RA-u3	TATTTAGTTTTCAGATCAGTAATCTCA
6439669	CG9384-RA-u3	ATGATGCTTTTACCCTCGATTATCTCA
6444052	CG5185-RA-u3	GTTCAGCTTCGCATGTTCGTAATCTCA

Table 9, Bantam candidate targets eliminated with 40% FE filter.

None of these targets could survive a 3' free energy constraint as low as 40%. On the other hand, in micTar free energy is not used as a criterion for full seeds. The most widely used post processing for the elimination of false positives is evolutionary conservation.

Evolutionary Conservation Filter

Although there are criticisms to use the evolutionary conservation [20] as a filter, it is widely used [7,9,10-13], and recommended [23] to select the functioning targets. It is obvious that a miRNA cannot be aware of the evolution and, it cannot use evolution to select its targets. As the latest results show [20-22] that the miRNA regulation is much more complex a process than it was initially expected. Although indirectly, evolutionary filter can be a tool to incorporate some of the unknown interactions into the model.

The biggest problem with the evolutionary conservation filter is that not all the sequenced genomes are not fully annotated [24]. The general approach taken is to find the orthologs of the target genes in *Drosophila melanogaster* in relatives like *Anopheles gambiae* or *Drosophila pseudoobscura* and to take some 1000-2000 nucleotides downstream of that gene which is expected to contain the 3' UTR regions [9,21]. Those sequences are aligned with the annotated *D. melanogaster* 3'UTR and the candidate targets not falling inside the conserved regions are filtered out.

Evolution Analysis by BLAST Search

Initially the conservation analysis was not the objective of this study. For this reason MicTar does not have conservation filtering. Since miRAnda and PicTar are reliant on evolutionary conservation to locate the functioning targets, it was decided to check the results of micTar by applying conservation analysis.

The whole 3'UTR sequence of a *Drosophila melanogaster* target gene found by micTar are is searched against *Drosophila pseudoobscura* genome by BLAST search. If the search gives good local alignments (longer than a miRNA length) with *D. pseudoobscura* then the seed sequence is searched within the alignment with text search tools. The known targets are very well conserved almost over their entire length as shown in Figure 15.

```
CG5123- hid  bantam target
  Query: 1699   attgctaattagttttcacaatgatctcggtaaagttttgtggcct 1744
                ||||| ||||||||||||||||||||||||||||||||||||||
  Sbjct: 578684 attgccaattagttttcacaatgatctcggtaaagttttgtggcct 578729
```

```
CG6575-glec  bantam target
          Query: 519    caatttactttgtgtcatgatctcaattattaaaa 553
                         ||| |||||||||||||||||||| |||||||
          Sbjct: 740652 aaatgtactttgtgtcatgatctcataattaaaa 740686
```

```
CG1030-Scr  mir-10 target
      Query: 1772   ttgccactgaagaacaaattcggaagataaacaggaagtaaaa 1814
                    |||||||||||||||||||||||||  ||||||||| |||||
    Sbjct: 671917   ttgccactgaagaacaaattcggaagtcaaacaggaactaaaa 671959
```

Figure 16, BLAST alignments of some known *D. melanogaster* targets in *D. pseudoobscura.*

CG16700 which is in disagreement in Table 4, with both with miRanda and Pictar is not conserved and could not be located within the alignment. It shows that the disagreements with Pictar and miRanda are eliminated with the use of conservation filter.

It has also been interesting to observe that the long UTR sequences are better conserved than shorter UTR sequences as noted by Stark *et al*. [29]. This implies that some UTRs are evolutionarily conserved to be miRNA targets, while others are evolved to avoid becoming miRNA targets. This had been theoretically postulated before3 with the concept of "anti-targets", and, currently, it became a widely accepted fact of miRNA regulation phenomena [22, 29].

CG 31647 and CG 15316 which are very strong target candidates found for *bantam* by micTar and could not pass the evolutionary filter which again matches the results of both miRanda and PicTar. Target site on CG16700 3'UTR for *mir-7* is not conserved either, and it could not pass the evolutionary conservation filter. These three targets are not shown in the

lists of miRanda and PicTar which shows micTar is able to find the best candidates of both methods and if the non-conserved targets found by micTar are eliminated.

The alignments in Figure 15 are not known whether they are on the orthologous genomic loci. A further check is done using USCS Genome Browser [31,32] and the VISTA [30] tool which visualizes genomic alignments across several genomes. The genomic locations of found targets are entered into the browser and the built-in alignments across 7 species within ~ 22nt are observed.

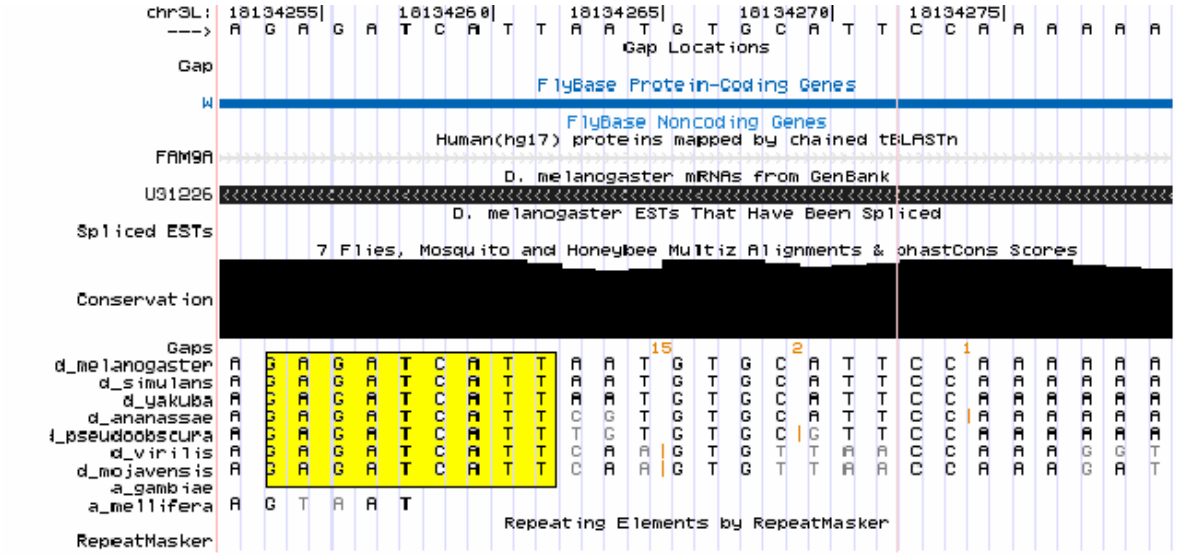

Figure 17, Conservation of the 8 nt seed (reverse complement) CG 5123 (*hid)* target across 7 *Drosophila* species.

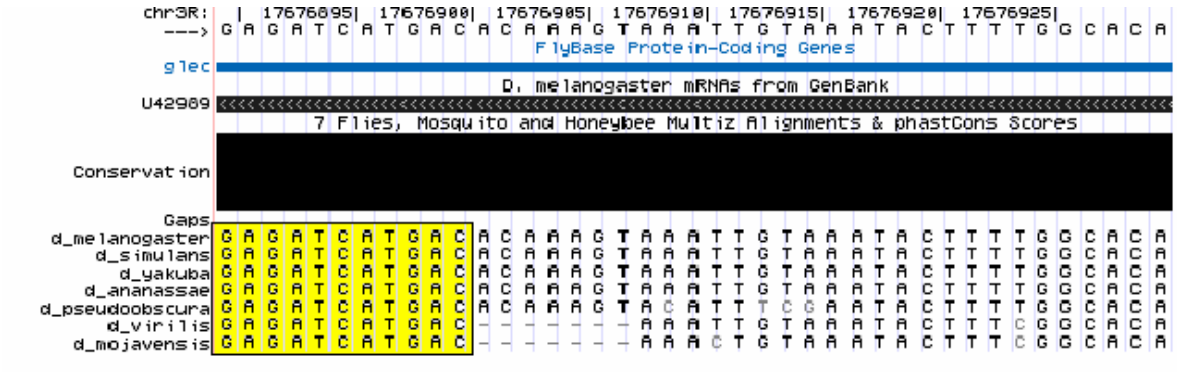

Figure 18, Conservation of the 11 nt seed (reverse complement) CG 6575 (*glec)* target across 7 *Drosophila* species.

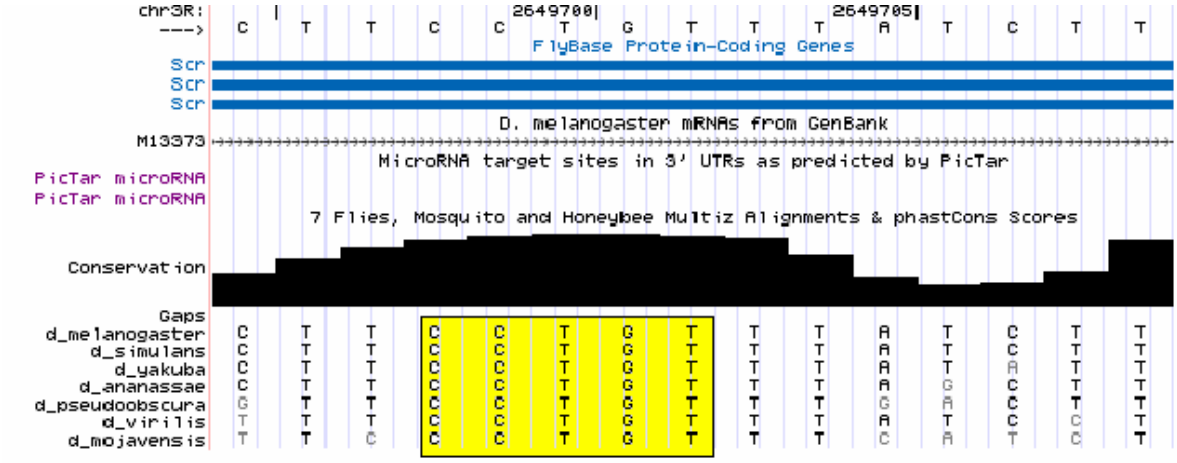

Figure 19, Conservation of 5nt 5' seed of CG1030 (*Scr*) across 7 *Drosophila* species.

Speed Considerations:

micTar is very fast to locate the targets for a given set of constraints. The results are instant, ranging from less than 2 minutes to 5 minutes. miRanda was downloaded from www.microrna.org and run on the same conditions. For small UTRs, it is also instant, but it takes as long as 30 minutes to align the whole genome. Also the version of miRanda at hand does not write the results to a file which eliminates some of the overhead. It was not possible to run PicTar on a local machine, and it is not possible to give information about its time performance.

CHAPTER 7

CONCLUSIONS

In this thesis, a novel approach is taken for a current bioinformatics problem: prediction microRNA targets. The identification of targets of microRNAs (miRNAs) is very important to understand their functions and the biological processes that they are involved. The problem was modeled as a constraint system and implemented in a constraint logic programming tool Sicstus Prolog. The work resulted in a software package **micTar**.

The constraints are developed on the latest findings of Cohen Laboratory at EMBL[14]. The interpretation of this work that "minimum exact match of 4 nucleotides is required" on the 5' side of miRNA for any functioning target led to a fast but a very comprehensive algorithm. With this unique approach of **micTar**, all UTR sequences of the genome are preprocessed into 256 4mer arrays. The search is done only on the 3 fourmers of miRNA at the 5' side with starting positions at 1, 2 and 3. This radically reduces the search space and eliminates the overall sequence alignment of miRanda, both of which improve the speed performance of the algorithm. The matching fourmer positions are processed further with the additional constraints for 5' side and the 3' side.

One of the latest of the existing packages, **Pictar** only looks at 7mer perfect matches on the 5' and incorporates the 3' side by looking at free energy over the full length of miRNA.

Widely used **miRanda** package looks to overall complementarity, giving more weight to the 5'side. **micTar** sits in just in the middle of the two approaches, ignoring 3' when it is not necessary, and, incorporating it when the 5' side is not perfectly bound. The 3' matching had been incorporated into the program by edit distance constraint to be fast but soon it proved to be wrong. Instead, the Free Energy Filter works as a postprocessor removed most of the target UTR segments which are in disagreement with the compared packages.

Further strong eliminator of false positives was the conservation filter. Initially, it was not built into the model due to the criticism about its use [20]. When a small conservation analysis is done as shown in Chapter 6, the targets which do not appear in neither of the compared packages were removed. It was also recommended by Prof. Stephen Cohen to use evolutionary conservation to locate functional targets [23]. Since there are some protein groups involved as mentioned in Chapter 2, and there may be intermediate stages in the miRNA-mRNA binding, evolutionary conservation could be a way to incorporate them into the sequence based search models. The folding structure of the target mRNA is another factor which may limit the number of positions available to the matching miRNAs. On the other hand, it is still an open question why a perfectly matching target should not function at all, because sequence based RNAi is very successful and becoming a major exogenous means of control of gene regulation [22].

All those approaches including micTar are incomplete because all the results of computational approaches still need experimental verification. There is no standard data set against which to compare the specifity and sensitivity of the algorithms except to check for the known targets. Fortunately, the number of known targets is increasing, and, as more experiments being done, we learn more about the mechanics of the miRNA-mRNA relation. The future models might include the protein interactions and the structure of the involved proteins in miRNA regulation pathway. micTar, miRanda and PicTar are sequence based and the folding of the mRNA is not considered. There are approaches that incorporate the

mRNA structure into model [20], by folding the mRNA first, and look for available seed sites afterwards and ignoring conservation.

Also in all these experiments from which the rules of binding are derived, the expression levels of both miRNAs and the target mRNAs are so high, the expression of miRNAs are tissue specific, the found miRNA-mRNA relationships might not be occurring in time and space in any organism. The concentration level of target mRNA should also be incorporated, as the miRNA will regulate the ones high in cellular concentration among the cognate mRNAs [22].

With regard to the use of constraint logic programming (CLP) and Prolog in this problem, no unsuitability of the tool for bioinformatics has been observed in terms of the speed of execution and memory management. Moreover, the declarative nature of the Prolog language enables to express the physical models more easily, closer to human logic. The down side of using Prolog is the unsuitability of the tool to create user-friendly user interfaces. Nevertheless with the provided C, C++, Visual Basic and Java interfaces in Sicstus Prolog, a user interface even a web interface can be built.

Constraint Programming is very useful in combinatorial problems where the search space is the Cartesian product of the domains of individual variables. With constraint propagation many of these possibilities are pruned away before the search starts. In micTar, the problem was reduced to a single variable problem, and because of this, the benefit of constraint programming was not taken to the full extent, except simplification of programming by leaving the search to constraint processor. Different models could be built with more variables, and different performances could be obtained. In the alignment algorithm where CLP is used in its full extent, the performance was not satisfactory. This may be due to the model employed; to have results that are biologically meaningful a scoring function is used to optimize the solution. Care must be taken in modeling problems as

optimization problems and there should be enough number of constraints to prune the non-viable solutions before optimality search starts.

The next version of micTar will start with conservation as a constraint rather than a post processing filter, and will incorporate the folded secondary structure of the mRNA. The conservation analysis will be done on a multitude of closely related and one not so closely related genomes (e.g. species from insects and one rodent) at the same time. Since micTar preprocesses the genomes into 4mer arrays, if one of important fourmers of miRNA is found at position *P*, e.g. Drosophila, the same fourmer will be constrained to exist in the ortholog UTRs in a predefined interval around *P*. If the structural analysis option is selected, the searched UTR first will be folded and the fourmer will be searched on the folded structure to look for available places for miRNA binding. The free energy filter should be improved in speed performance, for this reason the sequence alignment will be done with dynamic programming instead of constraint programming.

Finally, Constraint Logic Programming is a new paradigm in programming and its use in bioinformatics opens up new possibilities and we should explore more use of it in coming problems. miRNAs are very important post transcriptional regulators of gene expression and a lot of work other work should to be done in putting their role in gene regulation in a much more systemic way.

BIBLIOGRAPHY

1. **Lee R.C., Feinbaum R.L and Ambros V.(1993)** ***The C. Elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14*****. Cell 75, 843-854**
2. **Ambros V. (2001).** ***microRNAs: tiny regulators with great potential*****. Cell 107,823-826.**
3. **Bartel D.P. and Chen C.Z. (2004)** ***Micromanagers of gene expression: the potentially widespread influence of metazoan microRNAs*****. Nat. Rev. Genet. 5, 396-400**
4. **Bartel D.P. (2004)** ***MicroRNAs: genomics, biogenesis, mechanism, and function*****. Cell 116, 281-297.**
5. **He L. and Hannon G.J.** ***MicroRNAs:small RNAs with a big role in gene regulation.*****. Nat. Rev. Genet. 5, 522-531.**
6. **Craig C. Mello,Darryl Conte Jr. (2004).** ***Revealing the world of RNA intereference*****. Nature, Vol 431, 338-342.**
7. **Kiriakidou M., Nelson P.T., Kouranov A., Fitziev P., Bouyioukos C., Mourelatos Z., and Hatzigeorgiou A. (2004).** ***A combined computational-experimental approach predicts human microRNA targets*****. Genes Dev. 18, 1165-1178.**
8. **Lewis B.P., Shih I.H., Jones-Rhoades M.W., Bartel D.P., and Burge C.B. (2003).** ***Prediction of mammalian microRNA targets*****. Cell 115, 787-798.**
9. **Enright AJ, John B, Gaul U, Tuschl T, Sander C, Marks DS (2003).** ***MicroRNA targets in Drosophila*****. Cell 115, 787-798.**
10. **Stark A, Brennecke J, Russell RB, Cohen SM. (2003).** ***Identification of Drosophila MicroRNA Targets.*** **PLoS Biol 1: E60, 2003**
11. **Rajewsky N. and Socci N.D. (2004).** ***Computational identification of microRNA targets*****. Dev. Biol. 267, 529-535.**
12. **Rehmsmeier M Steffen P Höchsmann M Giegerich R. (2004).** ***Fast and Effective prediction of microRNA/target duplexes*****. RNA 10, 1507-1517.**
13. **Doench J G, Sharp P, (2004).** ***Specifity of microRNA target selection in translational repression.*****Genes & Development 504-511.**
14. **J. Brennecke, A. Stark, R. B. Russell, Stephen M. Cohen of EMBL. (2005).** ***Principles of MicroRNA-Target Recognition*****. PLOS Biology, Vol. 3.**
15. **Griffiths-Jones S, Grocock RJ, van Dongen S, Bateman A, Enright AJ.** ***miRBase: microRNA sequences,targets and gene nomenclature*****. NAR, 2006, 34, Database Issue, D140-D144.**

16. **Krek, D. Grün, M. N Poy, R. Wolf, L.Rosenberg, E J Epstein, P. MacMenamin, I. da Piedade, K. C Gunsalus, M. Stoffel & Nikolaus Rajewsky, (2005).** ***Combinatorial microRNA target predictions.*** **Nature Genetics, Vol 37, pp 495 – 500.**
17. **Po Yu Chen, Heiko Manninga, Krasimir Slanchev, Minchen Chien, James J. Russo, Jingyue Ju, Robert Sheridan, Bino John, Debora S. Marks, Dimos Gaidatzis, Chris Sander, Mihaela Zavolan and Thomas Tuschl. (2005).*The developmental miRNA profiles of zebrafish as determined by small RNA cloning.* Genes & Development, 19, 1288-1293.**
18. **Lewis B P, Burge C.B. (2005).** ***Conserved Seed Pairing, Often Flanked by Adenosines, Indicates that Thousands of Human Genes are microRNA Targets*****. Cell, Vol. 120, 15-20.**
19. **Stormo G.D. Fields D S. (1998).** ***Specifity,free energy and information content in protein-DNA interactions*****. Trends in Biochem Sci 23,110 -113.**
20. **Robins H, Li Y, Padgett R W. (2005).** ***Incorporating structure to predict microRNA targets*****. PNAS, Vol 102, no 11 4006-4009.**
21. **Grün D., Wang Y L, Langenberger D, Gunsalus K, Rajewsky N. (2005).** ***microRNA Target Predictions across Seven Drosophila Species and Comparison to Mammalian Targets.*****. PLOS Computational Biology. Vol 1, Issue 1, 51-66.**
22. **Rajewsky N (2006).** ***microRNA target prediction in animals*****. Nature Genetics, vol 38, 8-13.**
23. **Cohen S. (2006),** ***Personal communication.***
24. **www.flybase.org**
25. **Mathews, D.H.; Disney, M.D.; Childs, J.L.; Schroeder, S.J.; Zuker, M.; and Turner, D.H.,** ***"Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure,"*** **2004. Proceedings of the National Academy of Sciences, USA.** ***101*****. 7287-7292.**
26. **Hofacker I., Fontana W., Stadler P., Bonhoeffer S. Tacker M., Schuster P. (1994),** ***Fast folding and comparison of RNA Structures,*** **Monatsh.Chem. 125: 167-188.**
27. **Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),** ***Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,*** **Nucleic Acids Res. 25:3389-3402.**
28. **Marriot K., Stuckey P. (1998).** ***Programming with Constraints*****, The MIT Press, 1998.**
29. **Stark, A. Brennecke, J., Bushati, N., Russell R.B.&Cohen,S.M. (2005).** ***Animal MicroRNAs confer robustness to gene expression and have a significant impact on 3'UTR evolution,*** **Cell 123, 1133–1146.**
30. **Frazer KA, Pachter L, Poliakov A, Rubin EM, Dubchak I.** ***VISTA: computational tools for comparative genomics. Nucleic Acids Res*****. 2004 Jul 1;32(Web Server issue):W273-9**
31. **USCS Genome Browser http://genome.ucsc.edu**
32. **Kent, W.J., Sugnet, C. W., Furey, T. S., Roskin, K.M., Pringle, T. H., Zahler, A. M., and Haussler, D. (2002).The Human Genome Browser at UCSC.** ***Genome Res.*** **12(6), 996-1006.**

33. Carlsson M., Ottosson G., Carlson B. (1997). ***An Open-Ended Finite Domain Constraint Solver*****, Proc. Programming Languages: Implementations, Logics, and Programs.**

APPENDIX A

CONSTRAINT PROGRAMMING OVERVIEW

Adapted from the paper 'Constraint Programming -What is behind?

And 'Guide to Constraint Programming' at

http://kti.ms.mff.cuni.cz/~bartak/constraints/index.html

by Roman Bartak, Charles University, Czech Republic

Constraint programming is an emergent software technology for declarative description and effective solving of large, combinatorial problems. Constraint networks and constraint satisfaction problems have been studied in Artificial Intelligence starting from the seventies.

Constraint programming has been successfully applied to fields like computer graphics (to express geometric coherence in the case of scene analysis), natural language processing (construction of efficient parsers), database systems (to ensure and/or restore consistency of the data), operations

research problems (like optimization problems), molecular biology (DNA sequencing), business applications (option trading), electrical engineering (to locate faults), circuit design (to compute layouts), etc.

A constraint is simply a logical relation among several unknowns (or variables), each taking a value in a given domain. A constraint thus restricts the possible values that variables can take; it represents some partial information about the variables of interest.

Constraints have several interesting properties:

- constraints may specify *partial* information, i.e., constraint need not uniquely specify the values of its variables,

- constraints are *non-directional*, typically a constraint on (say) two variables X, Y can be used to infer a constraint on X given a constraint on Y and vice versa,

- constraints are *declarative*, i.e., they specify what relationship must hold without specifying a computational procedure to enforce that relationship,

- constraints are *additive*, i.e., the order of imposition of constraints does not matter, all that matters at the end is that the conjunction of constraints is in effect,

- constraints are *rarely independent*, typically constraints in the constraint store share variables.

There are two branches of Constraint Programming research which arise from distinct bases and, thus, use different approaches to solve constraints: Constraint Satisfaction and Constraint Solving.

Constraint Satisfaction

The Constraint Satisfaction Problem (CSP) is a problem where one is given:

a finite set of variables,

a function which maps every variable to a finite domain,

a finite set of constraints.

Each constraint restricts the combination of values that a set of variables may take simultaneously. A solution of a CSP is an assignment to each variable of a value from its domain satisfying all the constraints. The task is to find one solution or all solutions. Thus, the CSP is a combinatorial problem which can be solved by search.

Constraint Solving

Constraint Solving differs from Constraint Satisfaction by using variables with infinite domains. Also, the individual constraints are more complicated, e.g., nonlinear equalities. Consequently, the constraint solving algorithms uses the algebraic and numeric methods instead of combinations and search. However, there exists an approach which discretizes the infinite

domain into finite number of components and, then, applies the techniques of constraint satisfaction.

Solutions to Constraint Satisfaction Problems

Solutions to CSPs can be found by searching systematically through the possible assignments of values to variables. Search methods divide into two broad classes, those that traverse the space of partial solutions, and those that explore the space of complete value assignments stochastically.

The advantages of CSP over mathematical programming (e.g. LP) are twofold:

CSP representation of a problem is much closer to the original definition: the variables of the CSP directly correspond to problem entities, and the constraints need not be expressed in linear inequalities. This makes the formulation simpler, the solution easier to understand, and the choice of good heuristics to guide the solution strategy more straightforward.

CSP algorithms are essentially very simple; they can sometimes find solution more quickly than integer programming methods.

In general, the tasks posed in the constraint satisfaction problem paradigm are computationally **NP-hard**.

Systematic Search

From the theoretical point of view, solving CSP is trivial using systematic exploration of the solution space. The basic constraint satisfaction

algorithm, that searches the space of complete labeling, is called **generate-and-test (GT)**. The idea of GT is simple: first, a complete labeling of variables is generated (randomly); if this labeling satisfies all the constraints then the solution is found, otherwise, another labeling is generated. The efficiency of GT algorithm is poor because of non-informed generator and late discovery of inconsistencies. There are two ways to improve the efficiency of GT:

The generator of valuations is smart (informed), i.e., it generates the complete valuation in such a way that the conflict found by the test phase is minimized.

Generator is merged with the tester, i.e., the validity of the constraint is tested as soon as its respective variables are instantiated. This method is used by the backtracking approach.

Backtracking (BT) is a method of solving CSP by incrementally extending a partial solution that specifies consistent values for some of the variables, towards a complete solution, by repeatedly choosing a value for another variable consistent with the values in the current partial solution. BT can be considered as a merge of the generating and testing phases of GT algorithm. The variables are labeled sequentially and as soon as all the variables relevant to a constraint are instantiated, the validity of the constraint is checked. If a partial solution violates any of the constraints, backtracking is performed to the most recently instantiated variable that still has alternatives available. Whenever a partial instantiation violates a constraint, backtracking is able to eliminate a subspace from the Cartesian product of all variable domains. Backtracking is strictly better than generate-and test, however, its

running complexity for most nontrivial problems is still exponential. There are three major drawbacks of the standard (chronological) backtracking:

thrashing, i.e., repeated failure due to the same reason,

redundant work, i.e., conflicting values of variables are not remembered, and

Late detection of the conflict, i.e., conflict is not detected before it really occurs.

Consistency Techniques

Another approach to solving CSP is based on removing inconsistent values from their variable domains until the solution is obtained. These methods are called consistency techniques. The names of basic consistency techniques are derived from the graph notions. The CSP is usually represented as a constraint graph (network) where nodes correspond to variables and edges are labelled by constraints. This requires the CSP to be in a special form that is usually referred as a binary CSP (contains unary and binary constraints only). An arbitrary CSP can be transformed to an equivalent binary CSP. The simplest consistency technique is referred to as a node consistency (NC). It removes values from variables' domains that are inconsistent with unary constraints on respective variable. The most widely used consistency technique is called arc consistency (AC). This technique removes values from variables' domains that are inconsistent with binary constraints.

In particular, the arc (Vi,Vj) is arc consistent if and only for every value x in the current domain of Vi which satisfies the constraints on Vi there is some value y in the domain of Vj such that Vi=x and Vj=y is permitted by the binary constraint between Vi and Vj.

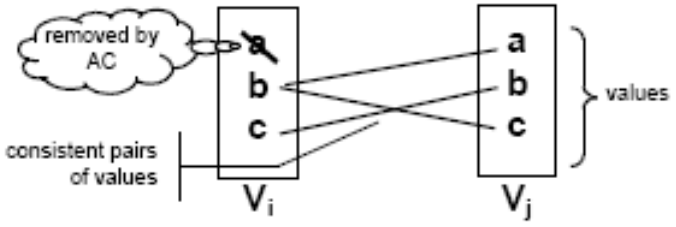

Figure A.1 Arc-consistency removes local inconsistencies (from R. Bartak)

There exist several arc consistency algorithms starting from AC-1 and concluding somewhere at AC-7. These algorithms are based on repeated revisions of arcs till a consistent state is reached or some domain becomes empty. The most popular among them are AC-3 and AC-4.

More inconsistent values can be removed by path consistency (PC) techniques. Path consistency requires for every pair of values of two variables X, Y satisfying the respective binary constraint that there exists a value for each variable along some path between X and Y such that all binary constraints in the path are satisfied. There exist path consistency algorithms like PC-1 and PC-2 but they need an extensive representation ({0,1}-matrix) of constraints that is memory consuming.

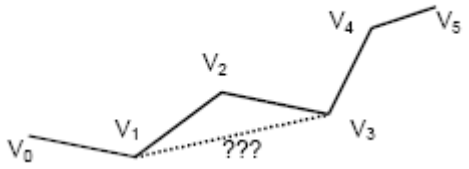

Figure A.2 Path consistency checks
constraints along the path (From R. Bartak)

All above mentioned consistency techniques are covered by a general notion of K-consistency and strong K-consistency. A constraint graph is K consistent if for every system of values for K-1 variables satisfying all the constraints among these variables, there exists a value for arbitrary K-th variable such that the constraints among all K variables are satisfied. A constraint graph is strongly K-consistent if it is J-consistent for all $J \leq K$. Visibly:

NC is equivalent to strong 1-consistency,

AC is equivalent to strong 2-consistency,

PC is equivalent to strong 3-consistency.

Algorithms exist for making a constraint graph strongly K-consistent for K>2 but in practice they are rarely used because of efficiency issues. Although these algorithms remove more inconsistent values than any arc consistency algorithm they do not eliminate the need for search in general. Clearly, if a constraint graph containing N nodes is strongly N-consistent, then a solution to the CSP can be found without any search. But the worstcase complexity of the

algorithm for obtaining N consistency in an N-node constraint graph is exponential. Unfortunately, if a graph is (strongly) K-consistent for K<N, then, in general, backtracking (search) cannot be avoided, i.e., there still exist inconsistent values.

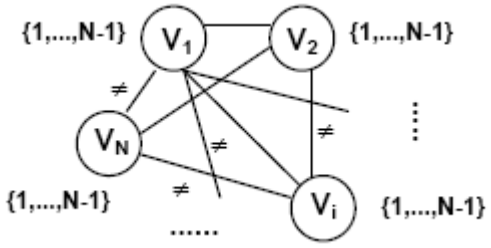

Figure A.3 Strongly N-1 consistent constraint graph still requires search (from R.Bartak)

Constraint Propagation

Both systematic search and (some) consistency techniques can be used alone to solve the CSP completely but this is rarely done. A combination of both approaches is a more common way of solving CSP. The **Look Back** schema uses consistency checks among already instantiated variables. BT is a simple example of this schema. To avoid some problems of BT, like thrashing and redundant work, other look back schemas were proposed. Backjumping (BJ) is a method to avoid thrashing in BT. The control of backjumping is exactly the same as backtracking, except when backtracking takes place. Both algorithms pick one variable at a time and look for a value for this variable making sure that the new assignment is compatible with values committed to so far. However, if BJ finds an inconsistency, it analyses the situation in order to identify the source of inconsistency. It uses the violated constraints as a

guidance to find out the conflicting variable. If all the values in the domain are explored then the BJ algorithm backtracks to the most recent conflicting variable. This is a main difference from the BT algorithm that backtracks to the immediate past variable.

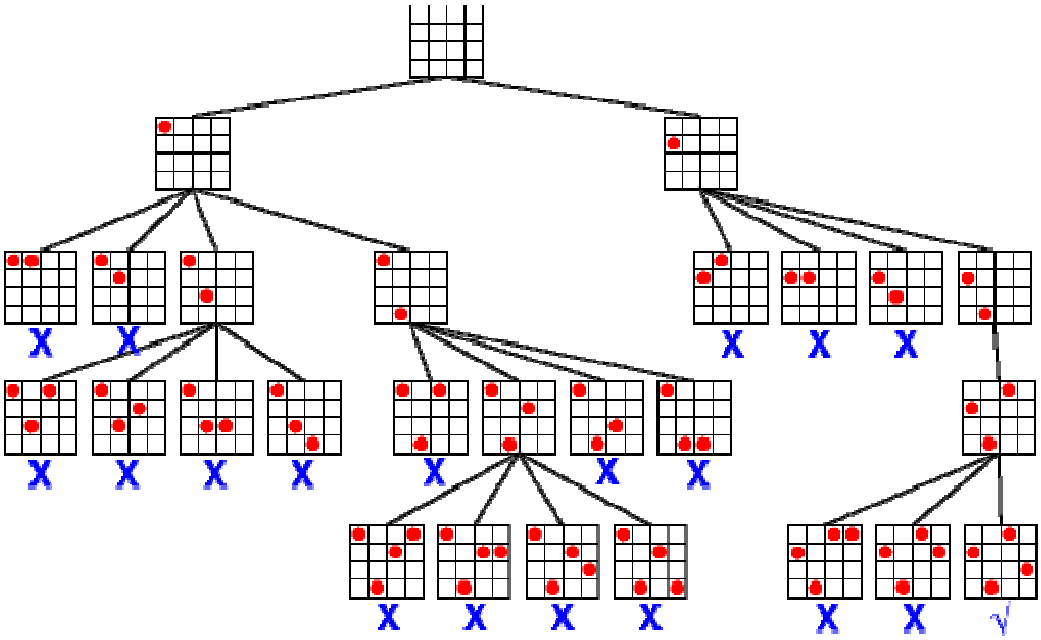

Figure A.4 Application of BT to 4-queens problem (from R. Bartak)

Other look back schemas, called backchecking (BC) and backmarking (BM), avoid redundant work of BT. Both backchecking and its descendent backmarking are useful algorithms for reducing the number of compatibility checks. If the algorithm finds that some label Y/b is incompatible with any recent label X/a then it remembers this incompatibility. As long as X/a is still committed to, the Y/b will not be considered again. Backmarking is an improvement over backchecking that avoids some redundant constraint checking as well as some redundant discoveries of inconsistencies. It reduces

the number of compatibility checks by remembering for every label the incompatible recent labels. Furthermore, it avoids repeating compatibility checks which have already been performed and which have succeeded. All look back schemas share the disadvantage of late detection of the conflict. In fact, they solve the inconsistency when it occurs but do not prevent the inconsistency to occur. Therefore Look Ahead schemas were proposed to prevent future conflicts.

Forward checking (FC) is the easiest example of look ahead strategy. It performs arc-consistency between pairs of not yet instantiated variable and instantiated variable, i.e., when a value is assigned to the current variable, any value in the domain of a "future" variable which conflicts with this assignment is (temporarily) removed from the domain. Therefore, FC maintains the invariance that for every unlabelled variable there exists at least one value in its domain that is compatible with the values of instantiated/labelled variables. FC does more work than BT when each assignment is added to the current partial solution; nevertheless, it is almost always a better choice than chronological backtracking.

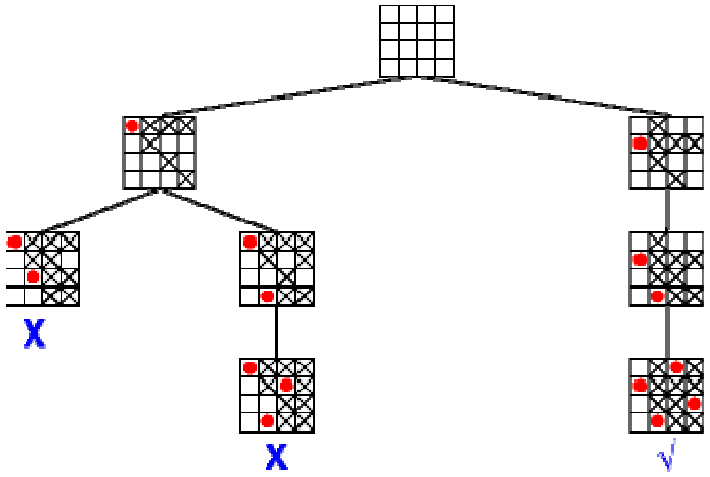

Figure A.5 Application of FC to 4-queens problem (from R. Bartak)

Even more future inconsistencies are removed by the Partial Look Ahead (PLA) method. While FC performs only the checks of constraints between the current variable and the future variables, the partial look ahead extends this consistency checking even to variables that have not direct connection with labeled variables, using directional arc-consistency. The approach that uses full arc-consistency after each labeling step is called (Full) Look Ahead (LA) or Maintaining Arc Consistency (MAC). It can use arbitrary AC algorithm to achieve arc-consistency, however, it should be noted that LA does even more work than FC and partial LA when each assignment is added to the current partial solution. Actually, in some cases LA may be more expensive than BT and, therefore FC and BT are still used in applications.

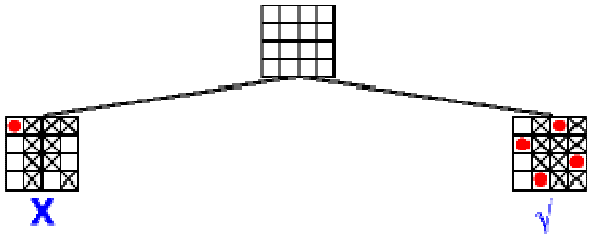

Figure A.6 Application of LA to 4-queens problem (from R. Bartak)

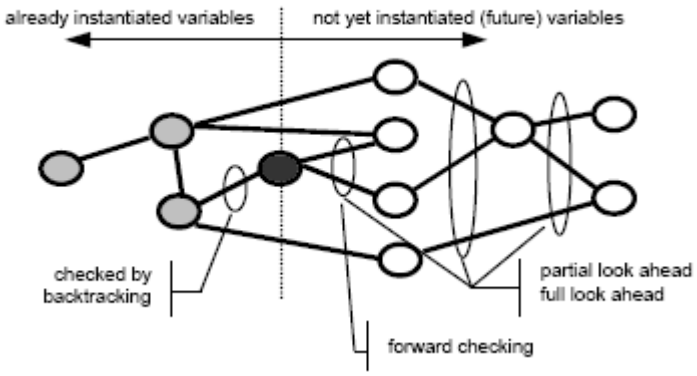

Figure A.7 Comparison of propagation techniques (from R. Bartak)

Limitations of Constraint Programming

Extensive application usage of constraint programming in solving real-life problems uncovers a number of limitations and shortcomings of the current tools. As many problems solved by CP belong to the area of NP-hard problems, the identification of restrictions that make the problem tractable is

very important both from the theoretical and the practical points of view. However, as with most approaches to NP-hard problems, efficiency of constraint programs is still unpredictable and the intuition is usually the most important part of decision when and how to use constraints. The most common problem stated by the users of the constraint systems is stability of the constraint model. Even small changes in a program or in the data can lead to a dramatic change in performance. Unfortunately, the process of performance debugging for a stable execution over a variety of input data, is currently not well understood. Another problem is choosing the right constraint satisfaction technique for particular problem. Sometimes fast blind search like chronological backtracking is more efficient than more expensive constraint propagation and vice versa. Sometimes, it is very difficult to improve an initial solution, and a small improvement takes much more time than finding the initial solution. There is a trade off between “anytime” solution and “best” solution.

APPENDIX B

SEQUENCE ALIGMENT WITH CLP

Let two sequences to be aligned A_i $i \in$ 1..n, B_j $j \in$ 1..m

Two sequences will be aligned in two equal aligned strings of size

k > max(m,n)

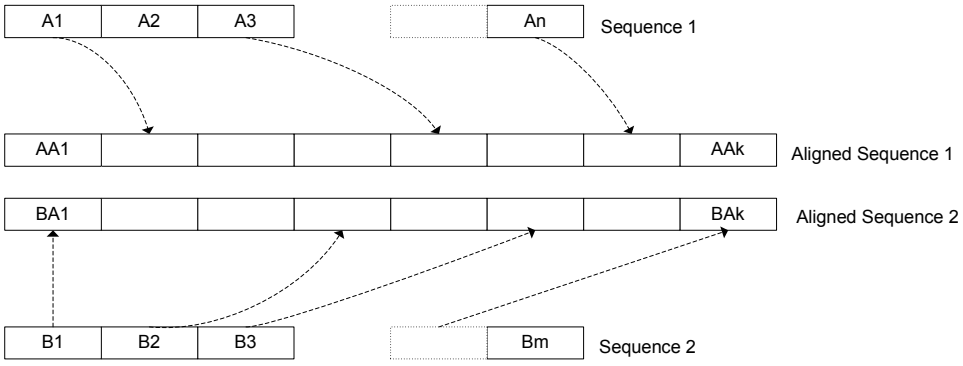

Figure 12

Two aligned sequences form a 2 x k matrix and the members of the sequences A and B will be distributed along this matrix. Insertions or deletions are represented as zeros.

The variables are the entries of the Aligned Sequences:

$AA_1 \ldots\ldots AA_k$, $BA_1 \ldots\ldots BA_k$,

The domain declarations:

$AAi \in \{0, A_1, .. A_i\} \qquad \forall i \leq n,$

$AAi \in \{0, A, C, G, T\} \qquad \forall i,\ n < i \leq k\text{-}n,$

$AAi \in \{0, A_n, .. A_{n-(k-i)}\} \qquad \forall i,\ k - n < i \leq k,$

The major constraints will be:

1) Ordering Constraint (symmetry breaker):

Let P(Ai) be the position of Ai in the aligned string AA

$P(A_{i-1}) < P(A_i) < P(A_{i+1}) \qquad 1 \leq i \leq k$

2) No Mutual Shifts Constraint:

$$AA_{i+} BA_i \quad > 0 \qquad\qquad 1 \leq i \leq k$$

Objective function: The optimal solution which looks for the best score of alignment:

$$AA_i = BA_i \Rightarrow Score = 4$$
$$(AA_i \neq BA_i) \wedge (AA_i > 0 \wedge BA_i > 0) \Rightarrow Score = -1$$
$$AA_i = 0 \wedge BA_i \neq 0 \Rightarrow Score = -3$$
$$AA_i \neq 0 \wedge BA_i = 0 \Rightarrow Score = -3$$