ABSTRACT

WHEELER, BENJAMIN M. Automating the Annotation and Discovery of MicroRNA in Multi-species High-throughput 454 Sequencing. (Under the direction of Professor S. Heber).

With the the advent of high-throughput pyrosequencing techniques, biologists now have the ability to obtain large quantities of sequence data faster, and at a substantially reduced cost to traditional Sanger sequencing. Currently, 454 DNA sequencing technology is the leading implementation of pyrosequencing. 454 sequencing is particularly effective for sequencing shorter molecules, where its relatively short sequence reads, when compared to traditional methods, is not a hindrance. This has popularized 454 sequencing with researchers exploring the many classes of small non-coding RNA (ncRNA).

With the capacity to achieve a 100-fold increase in throughput over more traditional sequencing techniques, effective use of 454 sequencing for the investigation of small ncRNAs requires new software tools and analysis techniques to handle the amount and variability of data produced.

We present a software tool, MirMiner, for the investigation of MicroRNA (miRNA), a class of ncRNA, in 454 sequence experiments, and apply it to two ongoing miRNA studies, encompassing 13 different genomes.

The intent of these studies was to catalogue previously known miRNA and discover novel miRNA. Starting with 796,286 sequence reads, MirMiner assisted these studies by identifying over 80 percent of the reads as putative miRNA or non-miRNA sequences (representing approximately 60 percent of the non-redundant sequences), and ranking the remaining reads in order of their potential to represent novel miRNA. In this smaller, ranked set of sequence reads, researchers have already discovered 101 novel miRNA families in six of the 13 species.

Additionally, a novel miRNA and miRNA star sequence predictor is introduced.

Automating the Annotation and Discovery of MicroRNA in Multi-species High-throughput 454 Sequencing

by
Benjamin Matthew Wheeler

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fullfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2008

APPROVED BY:

Dr. Nagiza Samatova

Dr. Brian Wiegmann

Dr. Steffen Heber
Chair of Advisory Committee

DEDICATION

This work is dedicated to my wife, Katie Wheeler, without whose support this work could never have been completed.

BIOGRAPHY

Benjamin M. Wheeler received B.A. Degrees in Mathematics and Computer Science from Thiel College in Greenville, PA in May 2006.

ACKNOWLEDGMENTS

I would like to thank Professor Steffen Heber for his guidance, patience, and support from my very first semester at North Carolina State University. Additionally, I am eternally grateful to Professor Kevin Peterson and Professor Brian Wiegmann for initiating the studies on which this Thesis is based, and collaborating with me throughout the process.

I would also like to thank Professor Wiegmann and Professor Nagiza Samatova for serving on my committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

With the advent of high-throughput pyrosequencing techniques, biologists now have the ability to obtain large quantities of sequence data faster and at a substantially reduced cost to traditional Sanger sequencing. Currently, one of the most popular commercially available pyrosequencing technologies is 454 sequencing. This technology is particularly effective for sequencing shorter molecules, where it's relatively read length is not a hindrance. This has popularized 454 sequencing with researchers exploring the many classes of small non-coding RNA(ncRNA). Examples include MicroRNA(miRNA), Piwi-interacting RNA, and small nucleolar RNA [1, 2, 3].

To effectively handle the amount and variability of such 454 sequencing data, new software tools and analysis techniques are needed.

We introduce MirMiner, a software tool to aid in the annotation, analysis, and discovery of miRNA in 454 sequence experiments. Further, we introduce mirStar, a tool to identify miRNA and miRNA star sequence pairs.

1.1 Motivation

MicroRNA are a class of short ncRNA that are quickly becoming the focus of many interesting and important biological questions. These relatively short, approximately 22 nucleotide RNA molecules have been found to play important regulatory roles in a wide range of organisms [4]. The total number of miRNA, their evolutionary conservation across species, and the scope of their regulatory activities are active areas of research [5].

A necessary precursor to any comprehensive use or study of miRNA is the ability to annotate known miRNA sequences, and discover novel miRNA sequences within the organism or organisms of interest.

Direct cloning of miRNA in cells is a problematic approach to identification, as these experiments are often dominated by a small number of highly expressed miRNA [6]. The traditional laboratory method, direct cloning and sequencing, is limited by the need to obtain large numbers of sequence reads, so as to capture both highly and rarely expressed miRNA. However, the availability of high-throughput pyrosequencing technology has provided renewed interest in direct cloning as a feasible means of miRNA study. On current machinery, such high-throughput technology can sequence approximately 100 megabases of raw DNA per 7-hour run, and even combine multiple species in a single sequencing experiment [7]. This represents a 100-fold increase in throughput when compared to prior Sanger sequencing techniques [8], and lets researchers obtain sufficient sequence reads for useful miRNA studies in a fraction of the time previously required.

However, in the large datasets being produced, low-expressed miRNAs represent only a small fraction of the total sequence reads. These reads are muddled by highly expressed miRNA, degraded products of larger RNA, and other classes of small ncRNA. The situation becomes even more complicated if PCR (cloning/amplication) and sequencing errors are considered. A recent study [3] has shown as many as 20 percent of cloned sequences in 454 experiments differ by one or more base pair mismatches to the genome, presumably caused by post-transcriptional modification, or PCR/sequencing error. This variability, when combined with the vast number of sequence reads, makes the identification of known and novel miRNA a challenging task.

1.2 Contribution and Results

A software tool, MirMiner, is presented to automate the discovery and annotation of miRNAs within the context of multi-species, high-throughput 454 sequencing. MirMiner extracts and annotates known miRNAs, potentially novel miRNA, and ‘biologically interesting’ non-miRNAs from raw 454 sequence data (Figure 1.1). In addition, a classifier, mirStar, was developed to predict candidate miRNA and miRNA star sequence pairs from a set of potential pairings, without the need for additional genomic information. This classifier can be used in combination with MirMiner to further prioritize candidate miRNAs for subsequent validation experiments.

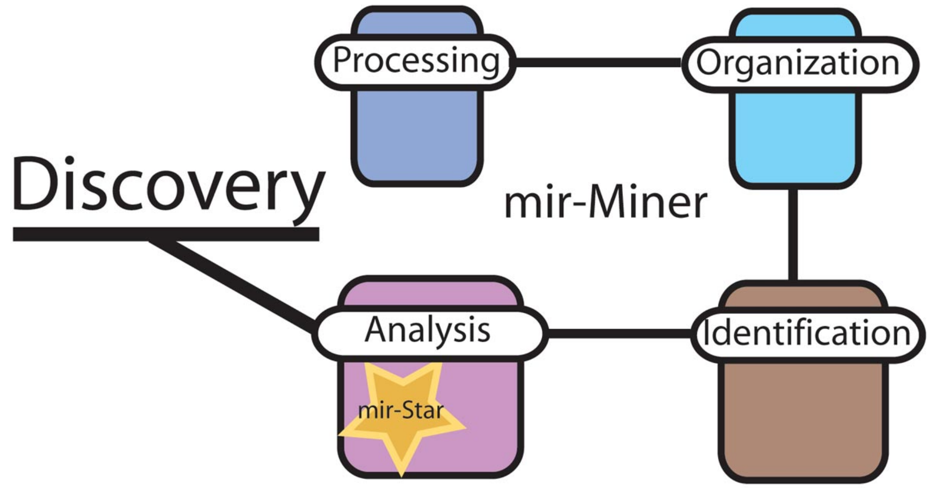

Figure 1.1: MirMiner divides the miRNA annotation and discovery process into four steps, Processing, Organization, Identification, and Analysis, changing the raw sequencing data from a set of unclassified elements into sets of biologically interesting sequences worthy of further investigation. At the final stage of this process the classifier mirStar may be used to identify potential miRNA:miRNA* sequence pairs.

MirMiner was applied in two ongoing miRNA studies spanning 13 different species, totaling 796,286 cloned sequences. It automatically annotated over 80 percent of the total sequence reads, identifying 108,822 reads matching 89 known miRNA families and 443,368 reads matching non-miRNA elements. The remaining reads were ranked according to their potential to represent a novel miRNA element. To date, within the set of ranked reads, 101 novel miRNA families, representing 5,576 total reads, have been verified among six of the species. From this same ranked set, mirStar predicted 399 miRNA:miRNA* pairs, encompassing 46 of the newly verified miRNA families.

1.3 Organization

Chapter 2 covers key biological concepts, introduces existing computational tools and algorithms utilized, and defines key terms. Chapters 3 and 4 introduce the software tools MirMiner and mirStar. Chapter 5 demonstrates the inclusion and reports the results of these tools within ongoing miRNA discovery projects. Chapter 6 concludes the work and offers insight to future directions for the project.

Chapter 2

Background

2.1 Cellular Biology

The cell is the building block of life, and serves as both the primary structural and functional element within all living beings. Cell function is facilitated through the production of proteins, used to carry out vital work within the organism. The blueprint by which to generate these necessary proteins and direct other cellular activity is stored in DNA (DeoxyriboNucleic Acid), which contains genetic instructions directing cell life and function. Inside the cell, DNA is organized as chromosomes, the complete set of which make up an organisms genome. RNAs (RiboNucleic Acid) are intermediaries between DNA and protein, serving both as templates for protein production, and governors of the molecular machinery responsible for protein production. The functional elements of a genome are divided into genes, where each gene is responsible for the coding of one or more functional products, such as proteins or regulatory RNA elements.

2.1.1 DNA

DNA contains the blueprint and the instruction manual of an organism. Usually, it is a double-stranded helix-shaped molecule composed of a combination of four different nucleotide bases: adenine, thymine, cytosine, guanine. These are represented as the letters A, T, C, and G. Each strand of the double helix is complementary to the other by means of complementary base pairings: $A \Leftrightarrow T$ and $C \Leftrightarrow G$.

Given this relationship, it is possible to represent a DNA sequence by just one or the other strand, realizing the missing piece can by reconstituted by means of the complementary base pairings.

2.1.2 RNA

RNA differs from DNA in both form and function. Rather than a two-stranded double helix, RNA molecules are single stranded, and the nucleotide base T, thymine, becomes U, uracil, when DNA is transcribed to RNA.

Protein coding genes are transcribed by the enzyme RNA polymerases into messenger RNA (mRNA). Through interaction with transfer RNA (tRNA), mRNA are translated by ribosomes (rRNA) into proteins. Thus, mRNA is called coding RNA, and all others termed non-coding RNA (ncRNA).

In addition to the mRNA, rRNA, and tRNA, several classes of other ncRNA exist which participate in regulation of other RNA or other cellular processes. Among these are several varieties of short ncRNA that have been a hotbed of research in the last 10 years [9]. Examples include MicroRNA (miRNA), small nuclear RNA(snRNA), small nucleolar RNA (snoRNA), and piwi-interacting RNA (piRNA). For many of these molecules, little is known about their specific roles or function. miRNA, for example, are known to regulate gene expression by RNA interference. However, other than their existence and general function, the mechanisms behind their interactions and many of their mRNA targets are unknown [6].

2.1.3 MicroRNA

Since the discovery of the first miRNA in 1993 by [10], miRNA have quickly become the focus of many interesting and important biological questions. While small, around 22 nucleotides, these RNA molecules have been found to play important regulatory roles in a wide range of organisms, having the ability to bind on their 5' end to UTRs(Un-Translated Regions) of mRNA through sequence complementarity [4]. They target and bind to mRNA by their 'seed' sequence, the nucleotides in position 2-7 or 8 as measured from the 5' end [4](Figure 2.1). The miRNA then act to either cleave or repress the bound mRNA sequence when combined with additional co-factors, thus providing post-transcriptional down-regulation of the target mRNA.

In animals, the expression of mature miRNA is predicated by transcription from

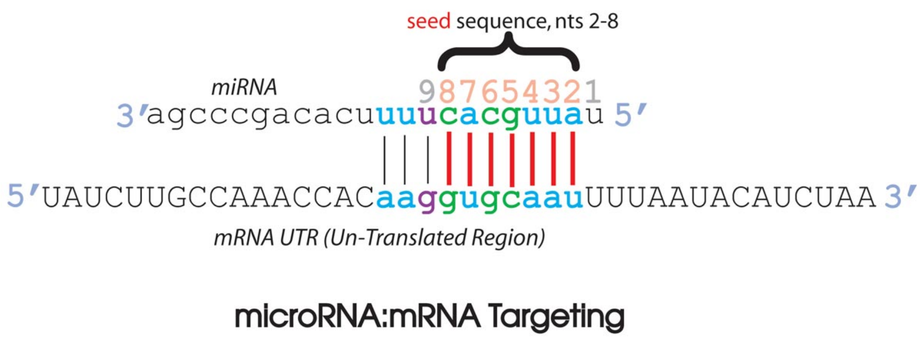

Figure 2.1: Targeting of mRNA by miRNA is accomplished by base-complementarity with the seed sequence, nts 2-7,8, on the 5' end of the miRNA, and a similar region on the 3' UTR (Un-Translated Region) of the target mRNA. The pairing shown is between partial sequence region of the mRNA shavenoid CG13209-PA and the microRNA miR-312 in *Drosophila melanogaster*, as shown in the miRBase Targets, Version 5.0.

the host gene to primary miRNA transcripts (pri-miRNA), which forms a RNA secondary structure. RNAase III endonuclease Drosha forms a precursor miRNA(pre-miRNA)[11]. This pre-miRNA is then cleaved by the RNase III enzyme Dicer to form an imperfect duplex, 2.2. The precise processing of miRNA differs in plants, using different compounds to perform cleavage, and different mechanisms to transport the RNA around the cell.

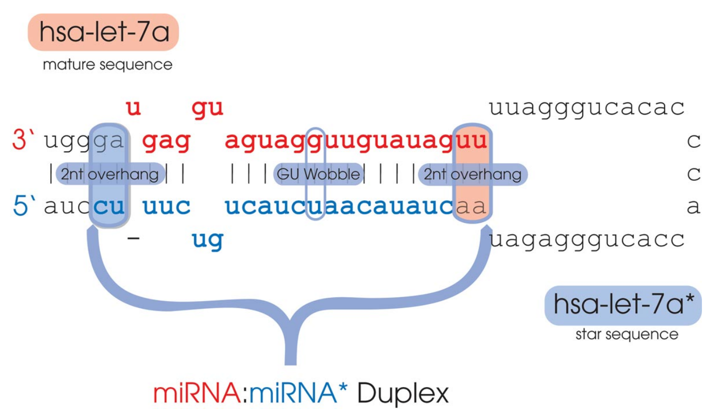

Figure 2.2: The miRNA:miRNA* pair forms a duplex within the pre-miRNA secondary structure following the GU Wobble model, and is conserved through to the inclusion is RISC, where one strand is subsequently degraded into a complex substrate, while the other incorporates into the RISC pathway.

In both cases, this duplex is incorporated in the RNA Induced Silencing Complex(RISC), working to down-regulate gene expression by the suppression and cleaving of targeted mRNA [12]. The strand participating in RISC is labeled the mature sequence (miRNA). The other is degraded as a complex substrate, and known as the star sequence (miRNA*). While normally degraded, miRNA* may still be found in libraries of cloned miRNA, with the mature counterpart often expressing at a rate of 10-100 times their star counterpart [4]. There is additional evidence to suggest the strand chosen for inclusion in RISC is not always unique, whereby certain miRNA* sequences may still act as functional miRNAs[4]. An outline of miRNA processing is shown in Figure 2.3.

Different miRNA are labeled based on form and presumed function. Similar sequences appearing across multiple species are grouped in miRNA families. Close variants of previously labeled miRNAs that are not similar enough to be considered identical may be classified as a variant, with an A, B, ect. appended to the family name.

Since the first miRNA was classified [10], functions of known miRNA have expanded to include a variety of roles in animals, and to developmental processes in plants, such as the regulation of cell proliferation and death in flies [14, 15], and the regulation of leaf and flower development in plants [16, 17, 18, 19]. However, the function of most small RNA molecules is unknown.

Two main avenues of miRNA investigation exist, gene finding and target prediction [6]. Gene finding involves identifying the genes which contain and translate to miRNA, whereas target prediction seeks to find the mRNA targeted by the gene for regulation.

2.1.4 MicroRNA Identification and Discovery

Several experimental and computational methods exist to identify and discovery miRNA. Experimental investigations normally begin by cloning small RNAs within the cells of the target organism, and sifting through this set of sequences to identify known and novel miRNA elements. However, such experimental sets are often composed primarily of a small number of highly expressed miRNA [6], making lesser expressed elements troublesome to clone in cells.

Computational methods were developed both to help verify cloned miRNA, and to identify candidate miRNA without the need for lab experimentation. Such methods (reviewed in [6, 20]), rely primarily on calculated secondary structure information to either predict novel miRNAs via searching the genome, as in mirScan [21], or by finding local high probability secondary structure formations via RNA folding algorithms, as in MFOLD

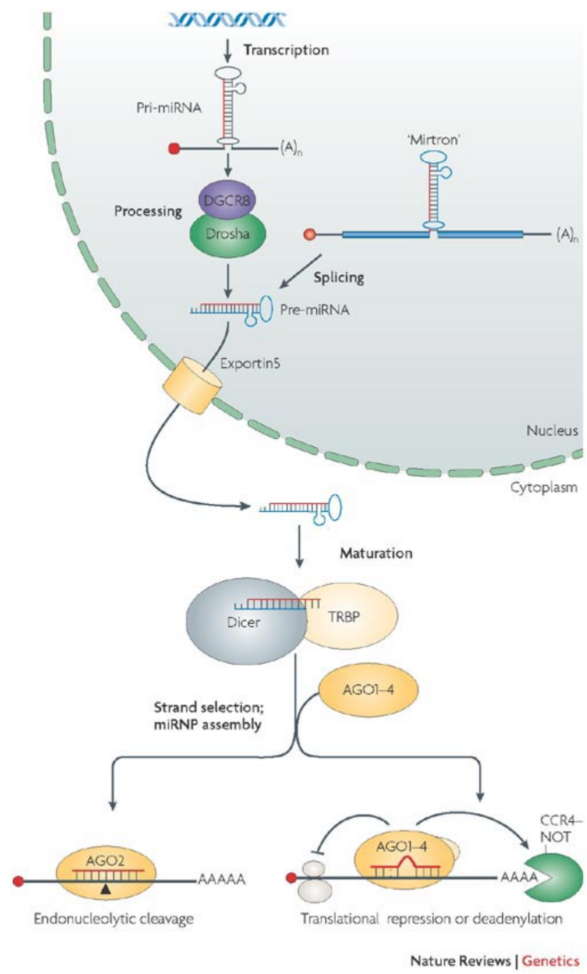

Figure 2.3: Image from [13]. miRNA processing in animals begins with the transcription of a pri-miRNA from the host gene. This pri-miRNA forms a hairpin structure, and is then processed by the enzyme Dicer [4]. This forms an imperfect miRNA:miRNA* duplex, which is then incorporated into RISC, whereby the conserved strand acts to cleave or repress the targeted mRNA by binding between the 5' and 3' ends of the respective molecules. In plants, the process is similar, except the pre-miRNA is cleaved to become the miRNA:miRNA* duplex while still in the nucleus, and undergoes additional editing to the 3' end [11]

Table 2.1: Five criteria, spanning two categories, were developed to facilitate a standardized system of miRNA identification and labeling [24]. Individually, none of these criteria are sufficient to positively identify a new miRNA. At least one criteria from each class, Expression and Biogenesis, must be met to classify a novel element. It is noted that demonstrated homologs of previously validated miRNA in different species are not subject to the above criteria [24], and either cloned expression in the target organism or computational prediction within its genome are enough to annotate such sequences.

Expression Criteria

A. Verification of approx. 22nt RNA via Northern Blot.

B. Cloning of approx. 22nt RNA in cDNA library from size-fractionated RNA (such as 454 Sequencing

Biogenesis Criteria

C. Prediction of Secondary structure fold-back, such as MFOLD.

D. Phylogenetic conservation of sequence and fold-back structure.

E. Increased expression of precursor strands in relation to a decrease in Dicer.

[22, 23]. However, these approaches necessarily rely on knowing the underlying genome sequence to facilitate identification, and are subject to high false positive rates with respect to their experimental cousins. These procedures also focus on the identification of pre-miRNA stemloops, which are extremely troublesome to clone, making their experimental verification difficult at best [3].

Experimental verification of a potential miRNA involves demonstrated expression of the target miRNA via Northern Blot [24], the gold standard, or similar analysis technique.

The criteria for both computational and experimental verification of candidate miRNA are listed in Table 2.1.

When studying organisms with incomplete genomes, the standard method of verifying cloned sequences against a reference genome is unavailable, forcing a renewed reliance on expensive laboratory experiments to verify potential miRNA candidates. Northern Blot allows researchers to test for very specific genomic sequences within a test subject. However, this process can take anywhere from one day to one week to complete the analysis of only a small number of potential miRNAs [25]. With this in mind, it is clear a naive attitude towards laboratory verification of potential miRNA is impractical, if not impossible, in cases where large sets of potential miRNA need to be examined. Separating known and potentially novel miRNAs from the vast amounts of non-miRNA sequence reads found in these data sets is thus a necessary and important step in the proper and timely annotation of miRNA within these un-sequenced species. Any computational method which can directly ease the uncertainty of these tests, eases the time and expense spent on these procedures.

When a reference genome is available, such separation is still desirable, so as to reduce and direct non-automated miRNA verification efforts to only those reads probable of representing novel miRNA. This limits the amount of time spent attempting to validate non-miRNA sequences, beneficially impacting research time and effort.

2.2 Enabling Technologies

2.2.1 DNA/RNA Sequencing

The goal of DNA/RNA Sequencing is to determine the correct base ordering of A,T,C, and Gs of a given DNA/RNA sequence or sequences. Its use has become a primary tool in the analysis of biological systems.

The defacto standard of gene sequencing is based on the dideoxy chain termination technique developed by Fred Sanger in the late 1970's [26]. Current machinery may generate sequence lengths to around 850bp per read, with the ability to sequence 96 templates on a machine, for a total of 96 sequence reads of 850bps [27].

Pyrosequencing dates back to the 1990's, but it was not until 2005 that 454 life sciences produced the first commercially viable machine based on pyrosequencing, achieving a 100-fold increase in throughput over Sanger sequencing [27]. When first introduced, 454 sequencing could effectively produce 25 million base reads at a maximum effective read length of 100 base pairs. Recent developments have seen the effective read length increase to more than 400 base pairs [28].

454 uses a modified pyrosequencing protocol, with nucleotide bases being determined by the release of inorganic pyrophosphate and the corresponding generation of photons [8].

A DNA library is created by randomly fragmenting an entire genome, separating DNA into individual strands, and adding specialized adapters to each fragment. Fragments are separated and amplified to produce 10 million copies of a unique DNA template. These fragments are then deposited on a fiberoptic slide containing over 1 million wells [8].

Sequencing reagents are added to the slide which react with the fragments, producing photons read by a CCD camera. Each reagent interacts with one of the 4 nucleotide bases, and produces a number of photons proportional to the number of bases the reagent incorporated during a given time step.

The process repeats with four different reagents interacting each with once of the four nucleic acid bases to generate sequence reads. Sequence tags and additional primers, added to the sequences in the initial stages, may be used to identify the origin of the sequence, and allow for multiple species to be sequenced at the same time.

A camera captures the intensity of the light, number of photons, coming from each well at each time-step. This technique produces a set of flowgrams, based off the of images captured by the camera, which are aligned and then analyzed to determine a set of consensus sequences. Problems with sequencing can occur where an individual base read is ambiguous due to the non-linear scaling of image intensity for long homopolymer, single base, reads. The signal strength as related to the number of bases incorporated per time step is linear for homopolymers up to length 8, but past this point accurate determination of the exact number of bases incorporated at a given step is reduced. The process is shown in Figure 2.4.

2.2.2 Northern Blot

Northern Blot analysis is a technique for the detection and expression analysis of RNA. Developed in 1977, it remains the gold standard for detection and analysis of many types of RNA, despite the development of more advanced techniques [25].

The goal of a Northern Blot analysis is to detect and measure the abundance of specific RNA molecules. This is accomplished by exposing a population of isolated RNA to specially made probes designed to bind through sequence complementarity with the RNA of interest. During hybridization the probe is exposed to the population of RNA, whereby complementary RNA will attach to the probe, and together bind to the experiment medium. At the conclusion of the experiment, the existence and expression level of a target RNA molecule can by evaluated by the quantity of RNA-probe, signal, attached to the experiment medium. Radioactive probes are often to used to aid in the detection of signals [25].

The time it takes to perform a Northern Blot Analysis is variable. A recently

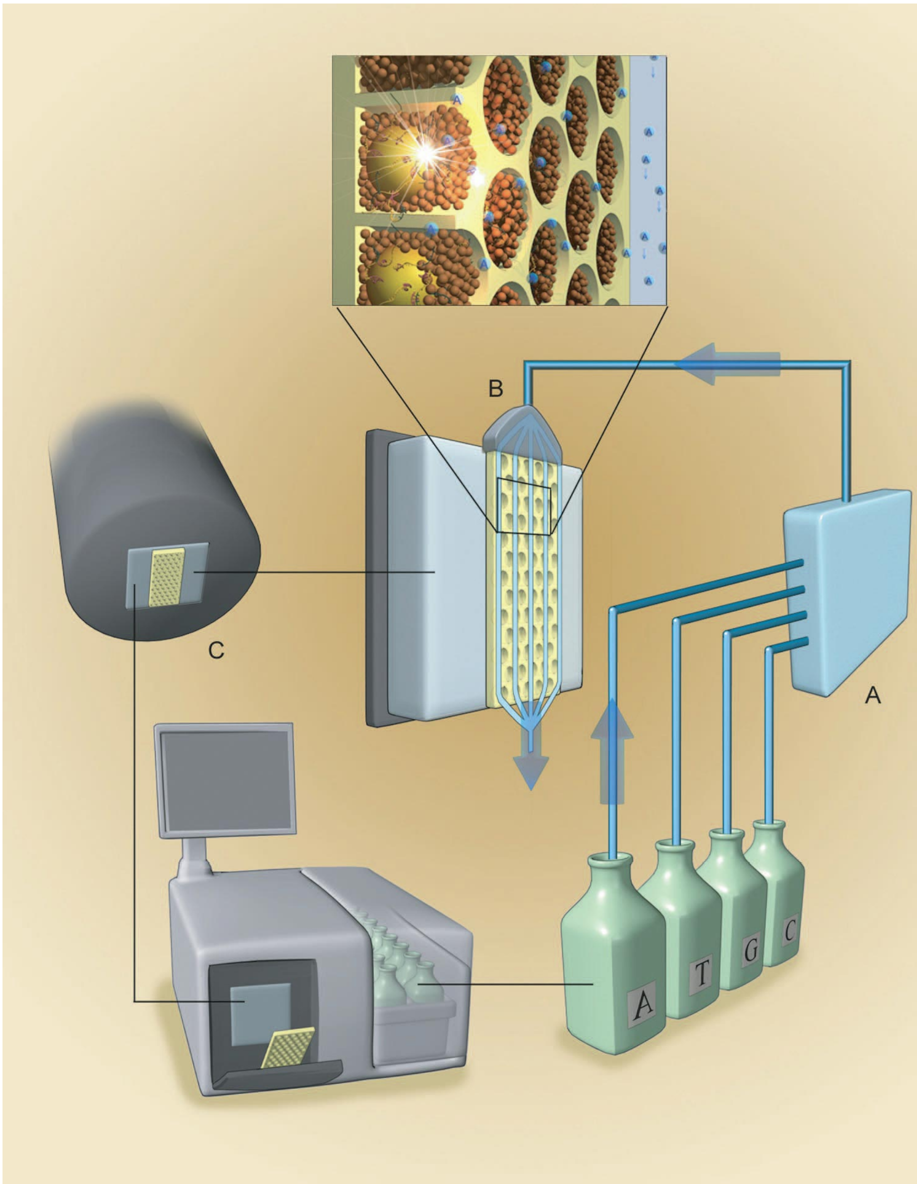

Figure 2.4: (Image taken from [8]) 454 Sequencing uses a modified pyrosequencing methodology to all massively parallel high throughput sequencing a fraction of the cost of traditional Sanger based techniques. The 454 Sequencing machine is made up of three components, (a) flow chamber, (b) CCD camera, (c) computer interface. The flow chamber holds the wells, which hold the genetic material. The CCD camera captures the light from the reactions with the 4 nucleotide bases.

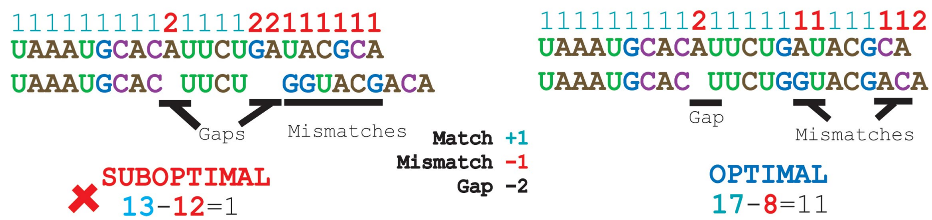

Figure 2.5: An optimal sequence alignment seeks to maximize a scoring function as applied across a set of base parings. Using the scoring criteria of 1 for a match, -1 for a mismatch, and -2 for a gap, the above shows the optimal sequence alignment of sequences A and B. Such an alignment may be calculated of an alignment matrix, shown on the left, utilizing dynamic programming to efficiently calculate and recover the optimal alignment.

released protocol [25], specifically for miRNA detection, reports the time will vary between 24 and 48 hours plus exposure time to a radioactive probe, a variable that will add between 2 hours and several days dependant on the activity of the isotope.

2.3 Computational Tools and Concepts

2.3.1 Sequence Alignment

A sequence alignment compares two sequences so as to infer regions of similarity that might be a consequence of a structural, functional, or evolutionary relationship. Sequence similarity may be measured by the number of proper match-ups present in the best alignment of the two strands (Figure 2.5).

A global alignment seeks to maximize the score of the sequence pair while utilizing every nucleotide from both sequences. A local alignment relaxes this condition, seeking the highest scoring substring pairing. Both alignment types can be computed via dynamic programming, see [29] for global alignment, and [30] for local alignment.

2.3.2 BLAST

The Basic Local Alignment Search Tool(BLAST) implements a heuristic search for local sequence alignment, matching a query sequence to a database of sequences. The effectiveness of the tool relies in it's ability to reduce the search space via filtration. The statistical significance of a match is reported as a p-value, or e-value.

Multiple BLAST versions for both protein and nucleotide sequence alignment exist.

Table 2.2: Regular expressions are made up of a collection of symbols. A set of such symbols defines a regular language. An example regular expression would be a^*, describing a language of 0 or more a's.

Σ	The set of symbols making up the text on which the regular expression will be applied. When multiple symbols are concatenated together they are called words.
(,)	Used to group words in an expression.
e	The empty word. A word that contains no symbols.
ø	The empty set. A regular expression with no words or other symbols.
$\cup$	Union. $A \cup B$ reads 'A or B'.
*****	The Kleene star. Used to specify 0 or more something, or the 'infinite union'. For example, a^* represents $\{\}, \{a\}, \{aa\}, \ldots$.

Blastn is designed to identify similarity in nucleotide sequences. Megablast improves blastn when used used on highly similar sets of sequences. It speeds computation by comparing the entire input set of queries to the database in parallel, and using a greedy matching algorithm. With correct parameters, megablast can run more over 10 times faster than blastn.

The programs blastn and megablast are used in this Thesis. Blastn is used to help classify homologous miRNA across taxons, through comparison with a database of known miRNA. Megablast is used to filter out and classify known non-miRNA elements in the set.

2.3.3 Regular Expressions

A regular expression is a form of formal language definition that is equivalent to finite state automata and regular grammars. Regular expression matching is a standard way of processing text-based data, providing a flexible means by which to identify regions of interest in strings of text.

Regular Expressions are made up of a collection of symbols, Table 2.2. These expressions specify regular languages, defined as anything that can be described by a regular expression. For example, the regular expression,

$$(ab)^*(c \cup d)(c \cup d)(aaa)$$

reads 'any number of ab's, followed by a c or d, followed by a c or d, ending in aaa. All the following strings would be included in this 'language',

$$ccaaa$$
$$dcaaa$$
$$abddaaa$$
$$ababcdaaa$$

Regular expression matching is a procedure that takes a regular expression and attempts to 'match' it to a provided string of symbols. If the string is a member of the language defined by the regular expression, it returns true, along with a parsed version of the string. Using the previous examples, regular expression matching would return true, along with,

$$(c)(c)(aaa)$$
$$(d)(c)(aaa)$$
$$(ab)(d)(d)(aaa)$$
$$(ab)(ab)(c)(d)(aaa)$$

If given *abccaa*, regular expression matching would return false, because it does not end in three a's.

2.3.4 Randomized Variable-Depth Local Search

Randomized Variable-Depth Local Search (RVDLS) is a generic randomized global optimization algorithm. Given a function $F(x)$, RVDLS tries to find the value for x which maximizes the value $F(x)$ across all possible inputs. RVDLS is an modification of Local Search. Local Search may be broken into the following steps,

1. Using the starting variable x, evaluate $\alpha = F(x)$.
2. In the neighborhood of x, find the value x' that maximizes $F(x')$, setting $\beta = F(x')$.
3. If $\beta > \alpha$, repeat the process with x' as the new starting value (x' is a better solution than x). Else, the 'best' solution has been found, stop.

Local Search suffers from two competing drawbacks, computational complexity and local maximum. If the area searched in step 2 is too small, then the algorithm will end too early, erroneously choosing the local maxima as the optimal solution. Increasing the search area will alleviate this problem, but also increases the computational complexity.

RVDLS modifies Local Search by changing the way x' is calculated. Instead of evaluating all candidate x''s within a neighborhood of x, a single x' is selected by 'hopping' from x based on a randomized decision. The new candidate x' uniformly at random in the neighborhood of x. If $\beta > \alpha$ (the new value for $F(x)$ better than the old value) then x' is accepted, otherwise the hop is discarded and the original x kept. The algorithm is then repeated. The neighborhood x from x' is selected is also varied. The neighborhood starts small, and is subsequently increased if the algorithm fails to find a better solution. By increasing this search area, local maxima can be avoided. Figure 2.6 illustrates the problem of global optimization within a two-dimensional input space, and how the inclusion of variable-depth and randomization effects the search for an optimal solution.

RVDLS achieves a speedup over Local Search at the expense of not guaranteeing an optimal solution. The effectiveness of this algorithm relies in iterating the algorithm enough times to stumble across a better β. The key is the number of iterations it takes to find an improved β should take less time than evaluating all of the search space. This algorithm and those like it are useful in situations where Exhaustive Search and other exact solutions are too computationally expensive perform.

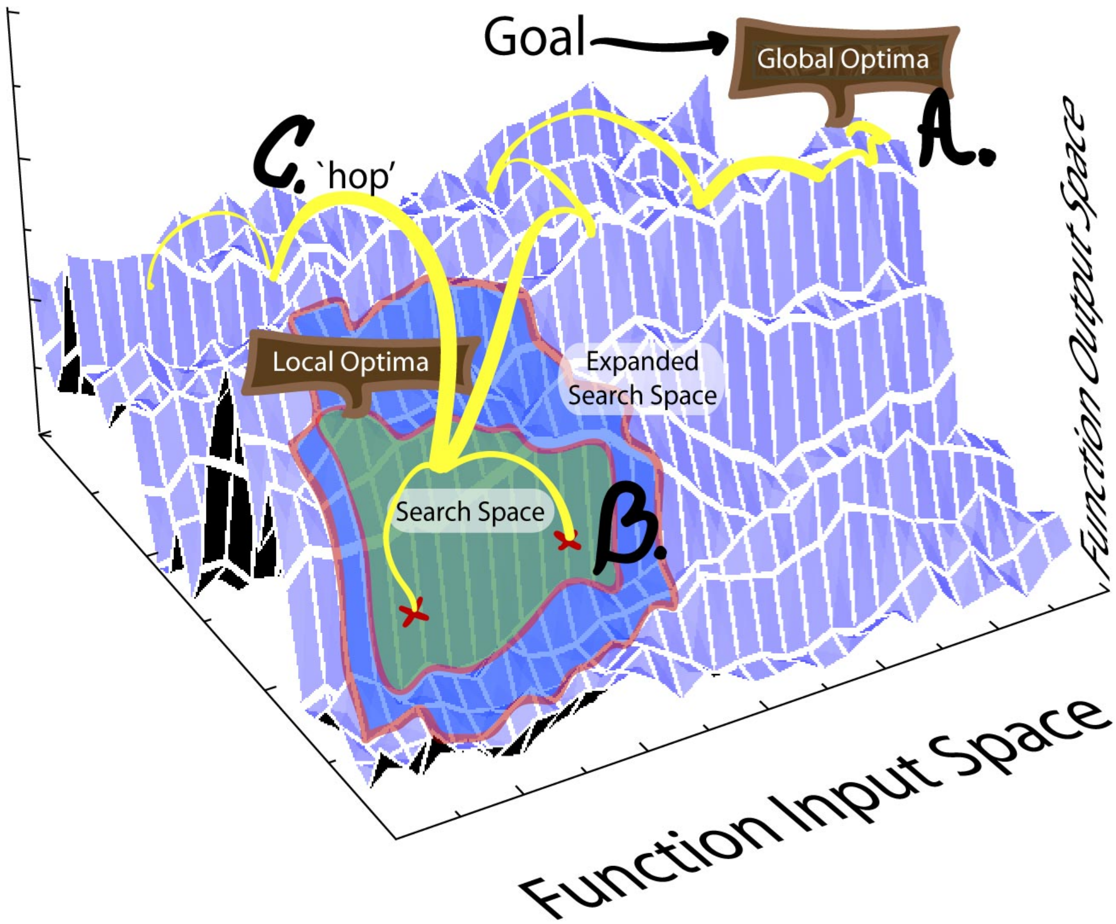

Figure 2.6: The problem of global optimization may be described by a plain with hills and valleys relating to some function value you are trying to optimize. The goal is to reach point A, the highest point in the area, called the global optimum. Local Search algorithms, those only looking for a solution in a limited amount of the function input space (the x-y region in the graph), have the danger of getting stuck in a local optima, if they do not search far enough abroad for a better solution, point B. Variable-Depth search will expand the neighborhood, when needed, in an attempt to find a better value. The Randomized variation of these algorithms introduces a random hop, point C, from one choice to the other. This hop precludes the computationally intensive task of evaluating all possible solutions in the neighborhood. In its simplest form, when you 'hop',you accept the value if it is better, and 'hop' again, or you reject the value, the red x's, and jump from your previous location.

Chapter 3

MirMiner

MirMiner is a program to annotate and discover novel and known miRNAs across a variety of species. It provides researchers with a workflow solution to automate the processing and analysis of sequence data resulting from high throughput, multiple species sequencing experiments. MirMiner assists the researcher with customizable tools to parse, sort, identify, and filter sequence data. Starting from a set of unclassified sequence reads, it presents the researcher with known miRNA sequences, and prioritized lists of potential miRNA novel sequences. These lists provide a concise set of sequences on which to perform various additional computational or experimental verification procedures.

MirMiner maintains intermediate information. This information provides a clear record of data manipulation, and allows users to reproduce all results.

MirMiner was developed through collaboration with researchers working with multi-species 454 experiments. Through this interaction a miRNA annotation and discovery workflow was determined, and then automated.

We will first explain the miRNA annotation and discovery workflow around which the tool was designed. Then we will describe the architecture and implementation. Finally, the functionality of each module is detailed.

3.1 Definitions

DNA sequencing experiments generate a quantity of *sequence reads.* Each of these *reads* represents a single strand of DNA present in a cloned *library* of genomic sequences. Each read will contain an *insert* corresponding to the cloned stand, and other nucleotides

Figure 3.1: Unprocessed 454 sequences are divided into a start tag, start primer, insert, end primer, and end tag. The tags are used to identify the species from which the target sequence was cloned. The primers are byproducts of the sequencing technology used to specify the beginning and end of the insert sequences.

that were added in the cloning and processing steps of the sequence experiment. One or more inserts will correspond to a *sequence* assumed present in the cloned organisms genome. Without matching against a genome, inserts may be determined as corresponding to the same sequence if they are sufficiently similar. The number of inserts which correspond to the same sequence is called the *frequency count* of that sequence.

3.2 MicroRNA annotation and discovery workflow

The search for miRNA begins with a set of raw sequence reads. Each unprocessed read contains a species identifier and sequencing byproducts in addition to the insert sequence (see Figure 3.1).

For each read, the non-insert elements are removed, and the inserts are grouped by species using the tag identifiers. Inserts are filtered for structural properties (such as length) to eliminate obvious non-miRNA candidates. From this point, all work is performed on a per species basis.

Duplicate inserts within species are identified and separated, both to eliminate redundancy within the data set, and to evaluate the frequency count of potential miRNA sequences. The frequency count serves as the primary discriminating statistic for unclassified sequences. To account for sequencing errors, and imprecise processing on the 3' end of miRNAs, a customizable match criteria is used to label and remove redundant sequences.

The resulting non-redundant sequences are searched against a database of known sequences, using customizable match criteria to infer homologous relationships. Identified sequences are removed from the data set and cataloged according to classification, known miRNA or known non-miRNA sequences. Examples of known non-miRNA sequences include degraded components of larger RNA molecules such as tRNA or mRNA, and other varieties of small RNA, such as piRNA or snoRNA.

Remaining sequences are potential novel miRNAs. These sequences are ranked by frequency count, and the existence of similar sequences among multiple species. Inter-species conservation is a strong indicator of a real miRNA. Within each species, potential miRNA:miRNA* sequence pairs are also identified, providing an additional ranking criteria. These rankings allow researchers to focus investigation efforts on those candidate sequences most likely to yield positive results. The entire process is displayed in Figure 3.2.

3.3 Architecture and Implementation

MirMiner has a modular design, and can be used as a complete tool, or subdivided into reusable modules suitable for inclusion in a customized workflow. MirMiner is subdivided into four modules,

1. Processing - Convert the data set into a useable form.

2. Organization - Sort candidate sequences into useful, manageable subsets.

3. Classification - Separate known and unknown elements within the data set.

4. Analysis - Characterize known and unknown elements, providing useful statistics of the data set, and rank orderings of candidate novel sequences.

MirMiner is implemented in Python, totalling approximately 2300 lines of code, and designed to run in a desktop computing environment. Programmed as Python package, work flow development with MirMiner is feasible across all platforms supported by Python, which include Linux, Macintosh, Solaris, Unix, and Windows. MirMiner uses BioPython [31, 32], an open source conglomeration of tools for computational molecular biology, to provide specific data structures and input/output functionality.

The MirMiner package is organized as a set of sub-packages, one for each of the four modules, Processing, Organization, Identification, and Analysis. A script within each sub-package is callable as a command line utility. A fifth sub-package contains general functions that do not fit within one of the other four. A main script in the root package contains a command line utility encompassing all steps in the MirMiner process, controlled by a configuration file. The structure and organization of the MirMiner package is shown in Figure 3.3.

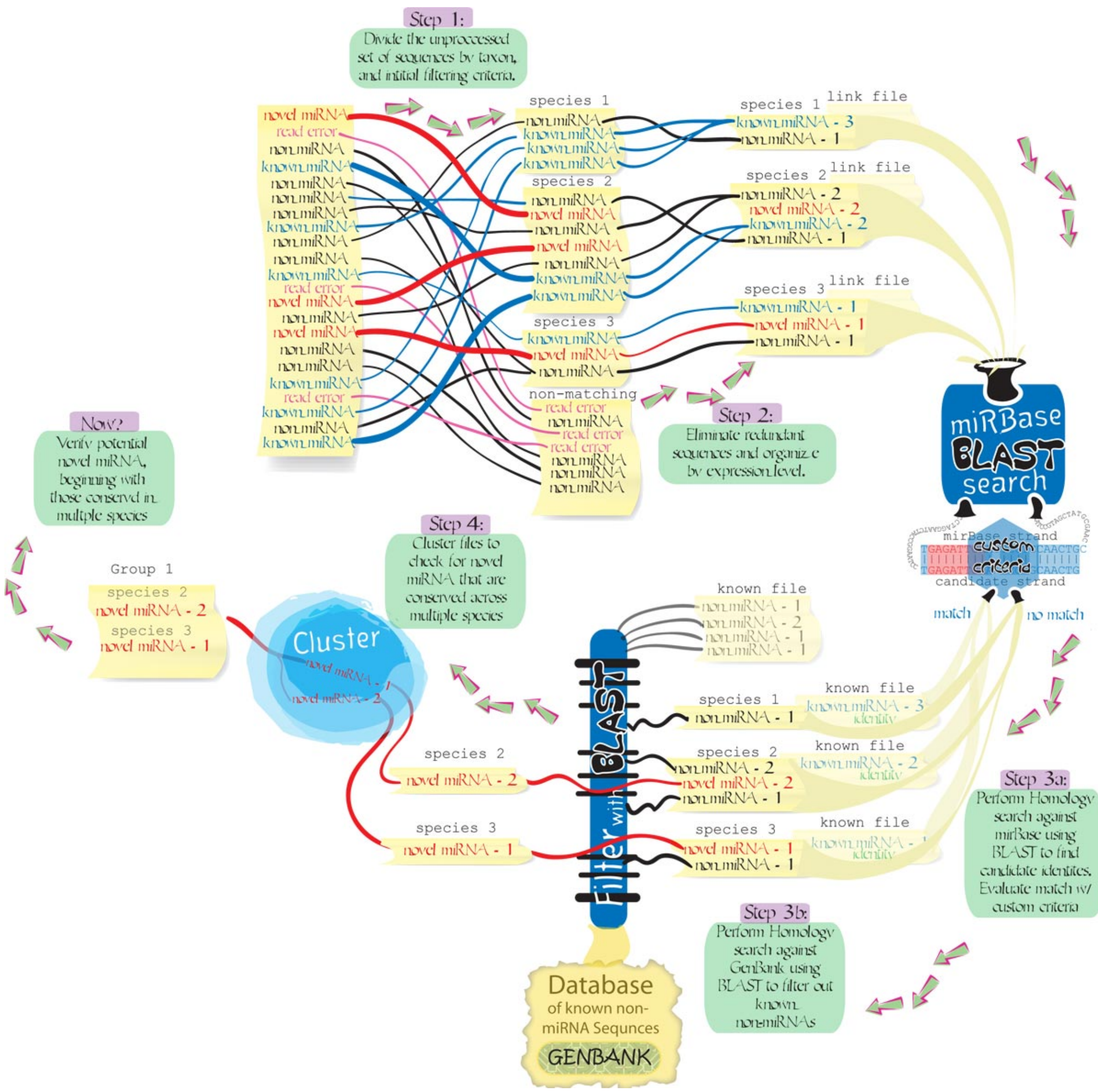

Figure 3.2: The miRNA discovery process is divided into four steps, Processing, Organization, Identification, and Analysis, changing the raw sequencing data from a set of unclassified elements into a sets of biologically interesting sequences worthy of further investigation.

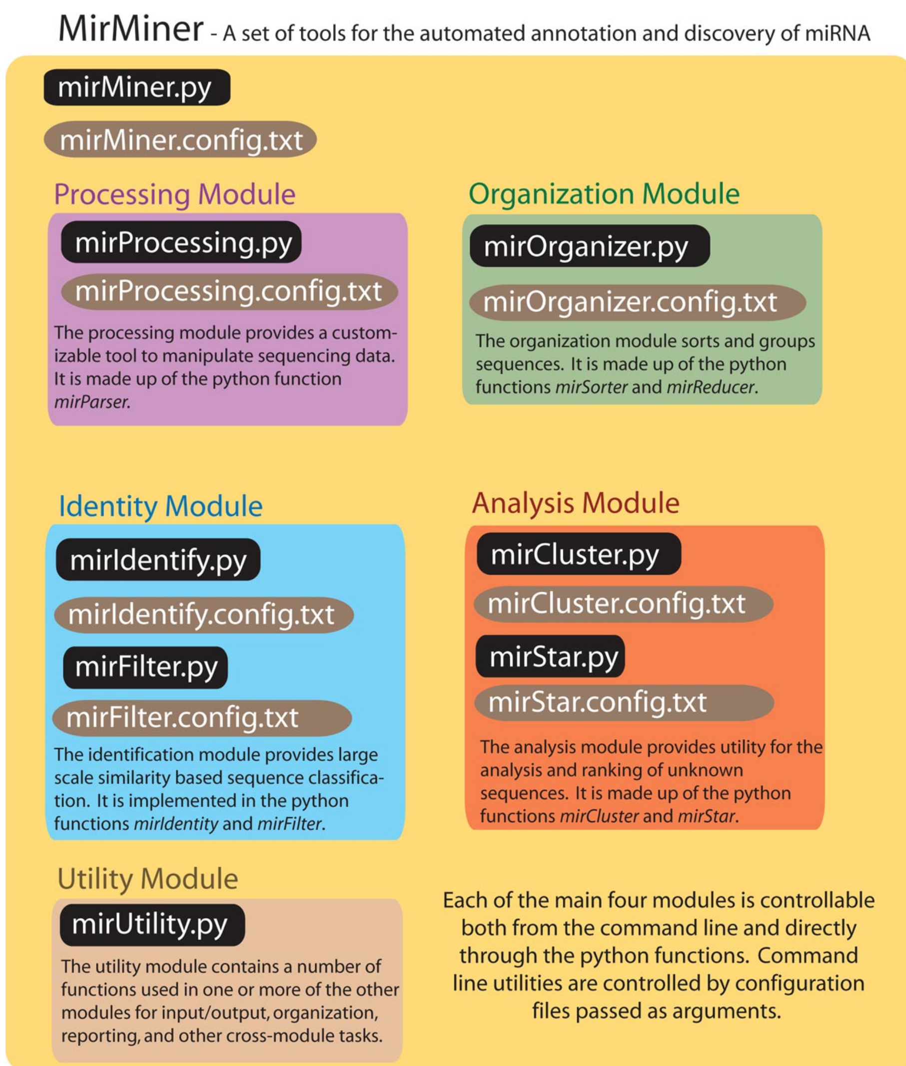

Figure 3.3: MirMiner is implemented as a Python package. It is organized as a set of sub-packages and command-line `scripts`. When used in Python, the functions which support the command-line utilities may be accessed through inclusion of the MirMiner package.

3.4 Processing Module

Processing is the first step in the miRNA discovery workflow. The processing module is designed to facilitate the conversion of unprocessed sequence reads into precise subsets of interest, allowing the researcher to set initial filtering criteria, and reformat the sequences as appropriate to the sequencing technology. Sequences are treated as sets of elements, each of which may be conserved or discarded as befits the problem instance.

3.4.1 Goals and Challenges: finding the sequence

The goal of the processing module is to accurately separate the insert from the raw sequence read, and process the information contained in the non-insert segments of the read. If the inserts are not properly extracted and the non-insert segments correctly evaluated, then further analysis can be meaningless.

The problem of correctly extracting the insert from the raw read is complicated when quality issues or other factors introduce variability into the length and nucleotides of the tags or primers. While in a perfect world the primers and tags will always correspond to reference elements, in our experience real data might look very different. In one of our experimental sets, almost 25 percent of the raw reads did not have primers corresponding to the reference when a naive cutoff was applied. Further investigation revealed the presence of extra nucleotides in the primers, dropped nucleotides from the end tag, and prematurely cutoff reads. Such errors in the tags or primers can be caused by several issues, including the construction of the DNA Library, cloning errors, and sequencing errors.

3.4.2 The anatomy of a sequence read

In multi-species 454 sequencing, reads from all species in the experiment are provided in a combined file. Every sequence consists of 5 distinct elements: start tag, start primer, insert sequence, end primer, end tag (See Fig. 3.1). In our example, a four nucleotide tag is used, along with a 17 nucleotide primer.

The start tag and end tag uniquely identify the species from which the read was cloned. The primers are byproducts of the sequencing technology and contain no information content. The insert is the actual nucleotide sequence cloned within the DNA library.

With 454 sequence reads, a quality measure is also provided with each read. Every nucleotide in the sequence read is assigned a number that represents the log-likelihood the base call is correct.

3.4.3 mirProcesssing

The command-line utility and function mirProcessing makes up the processing module, and provides users a way to split input sequences by species, and to filter species by discriminating features such as length and quality. Additionally, it annotates sequences for identification in latter parts of the workflow.

We use regular expressions and a set of definitions to process the sequence reads. Regular expression group matching is used in MirMiner to parse the sequence reads, and these parsed reads are then compared to the definitions for evaluation. Regular expression group matching will take the regular expression, and attempt to match it to a given query (sequence read). If successful, it will return the query as a list of groups as defined in the regular expression. An example of this concept is shown in Figure 3.4.

The pattern defines how a sequence read is broken up, and which piece corresponds to what part of the read (tag, primer, insert). Any valid regular expression that defines a set of groups may be used, allowing the inclusion of length restrictions, exact sequence restrictions, ect. An example pattern is shown in Table 3.1.

When regular expression group matching breaks up a sequence, the parsed read is compared to each of the definitions until a match is found. When the appropriate definition is found, the sequence read is processed by the definition and grouped with similar sequences. Definitions are processed in order, so a 'catch all' may be placed at the end to capture all 'Non-Matching' sequences.

An example definition would be $[`CTCG', `species', `Flatworm', 3]$, where the rule matches the sequence tag to 'CTCG', marks the 'species' annotation of the sequence as 'Flatworm', and conserves only the third group in the regular expression match.

The first stage in the miRNA discovery workflow may be accomplished by calling mirProcessing with an appropriate pattern and set of definitions, illustrated in Figure 3.5.

With data originating from 454 sequencing experiments, the user may also include the quality value of each read as an additional parsing criteria to remove low-quality sequences. This quality file may be used in conjunction with the sequence-reads to filter any sequences which fails a quality check. Users may specify either a straight cutoff or average cutoff value. A straight cutoff will remove a sequence if one nucleotide is below the cutoff

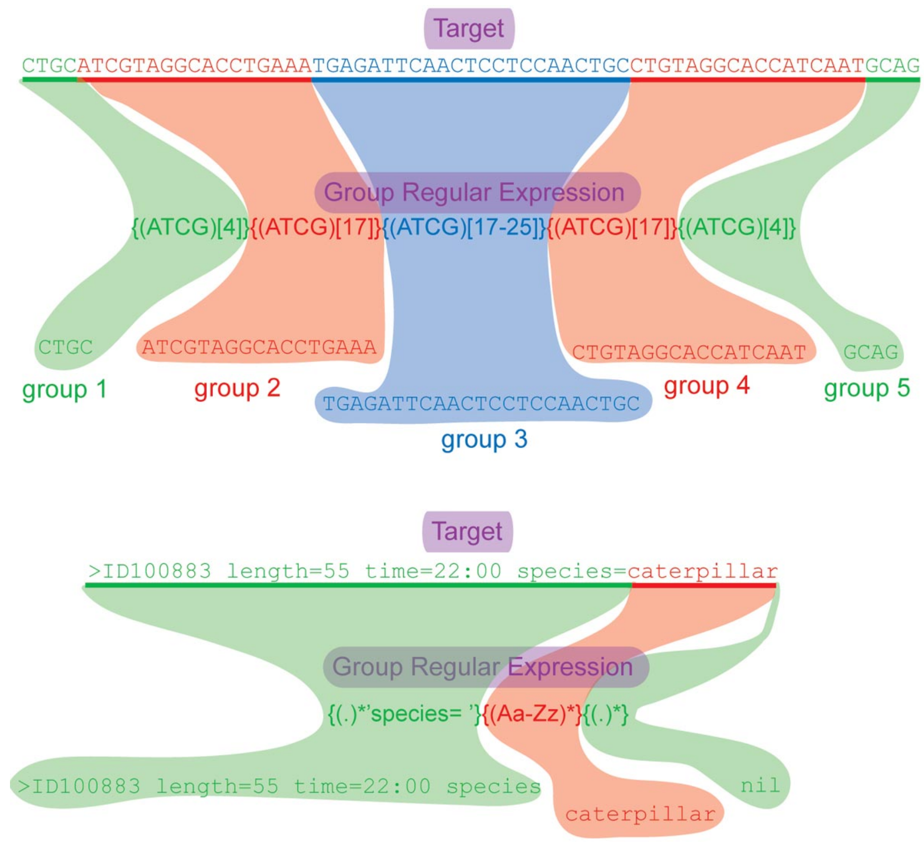

Figure 3.4: Regular Expression matching attempts to 'thread' a target through a regular expression. If successful, it returns a list of the matched groups. In the first example, the regular expression shown defines a sequence in five groups, denoted by {..}'s. Group 1 is 4 nucleotides long of any of the letters A,T,C,or G. Group 2 is 17 nucleotides long. Group 3 must be between 17-25 nucleotides long. Group 4 is the same as Group. Group 5 is the same as group 1. These groups are later compared to definitions to determine the appropriate sub grouping of the sequence. The second example illustrates it's use on the sample annotation of a sequence. Group 1 matches any number of characters or spaces followed by the text 'species='. Group 2 matches any number of letters. Group 3 matches anything. It is noted that NIL is a valid match for group 3, and that the regular expression will return a NIL group in this case.

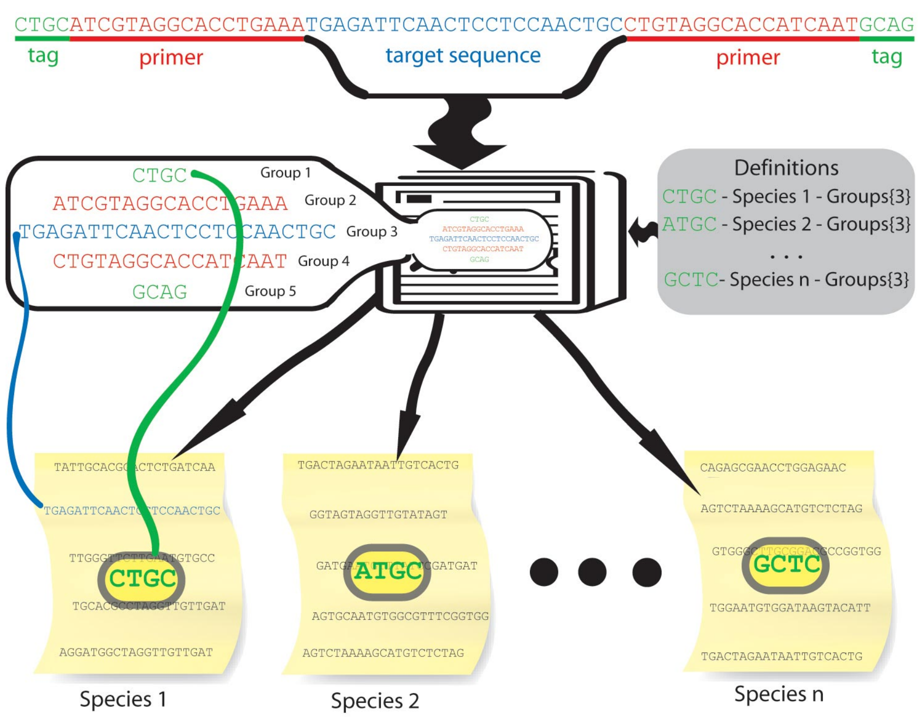

Figure 3.5: mirParser evaluates a sequence via regular expression matching, separating the tags and primers from the target sequence. The tag field is then compared to the species definitions in order to place the sequence in the appropriate file. Non matching sequences are thrown in a separate 'Non-Matching' file. Regular expression matching allows precise length, structure, and other filtering criteria to be imposed, allowing the tool to easily adapt to a specific data set. The pattern and set of definitions are shown in Table 3.1.

$$p \;=\; \{(\text{ATCG})[4]\}\{(\text{ATCG})[17]\}\{(\text{ATCG})[17\text{-}25]\}\{(\text{ATCG})[17]\}\{(\text{ATCG})[4]\}$$

$$D \;=\; \begin{bmatrix} tag_1 = \text{`}CTGC' & \mathit{field} = \text{`species'} & name = \text{`S 1'} & Groups = \{3\} \\ tag_2 = \text{`}ATGC' & \text{``} & name = \text{`S 2'} & \text{``} \\ \ldots & \text{``} & \ldots & \text{``} \\ tag_n = \text{`}GCTC' & \text{``} & name = \text{`S n'} & \text{``} \\ tag = ALL & \mathit{field} = \text{`event'} & name = \text{`no tag'} & Groups = \{All\} \end{bmatrix}$$

Table 3.1: Patterns, p and definitions, D, are the backbone of the processing module. The pattern describes how the sequence read should be evaluated, and the definitions map each read to a subset of interest to the user. With the above pattern, the first and last 17 nucleotides of the sequence read are assigned to groups 1, 2, 4, or 5. Then, if the remaining section of the read is between 17-25 nts, it is assigned to group 3. If the sequence read does not fit this pattern, it is discarded. These groups are then matched against the definitions. Reads are organized into files based on the definition they match, and annotated with the annotation and name specified. Only the group number(s) of the parsed read specified in the definition will be conserved. In the above example, only group 3 will be conserved in the final output for the first tag.

value, whereas the average cutoff will use the average of all nucleotide quality values.

3.5 Organization Module

Organization is the second step in the miRNA annotation and discovery process. It includes tools to sort and group sequences through a variety of similarity measures, providing a means to organize, rank, and purify the data set. While pyrosequencing will produce large volumes of sequences, the number of truly unique sequences is normally but a fraction of this entire amount. This module serves to collapse this set of potentially redundant data into a smaller set of sequences representing the unique elements.

The entire process is divided into two steps, a sorting phase and a reduction phase.

3.5.1 Goals and Challenges: making sense of it all

Sequencing via direct cloning results in a large number of reads originating from the same genomic sequence. When searching for miRNA, it is pointless to evaluate the same sequence more than once, making it necessary to identity and quantify this redundancy.

It is important to note that identifying redundancy is a challenging task. Sequencing errors, cloning anomalies, and biological processes may introduce small variations into the data set. It is advantageous to identify and remove these changes in order to obtain accurate numbers of unique sequences, as well as reducing the amount of work required by the user to identify known and novel elements.

3.5.2 Defining Redundancy

Redundancy is defined both as exact or near-exact duplication of the sequence-read nucleotides. The amount of redundancy of a particular sequence within a data set is its frequency count. This count can be used as a piece of evidence in the search for novel and the characterization of known sequences. For example, in the search for novel miRNAs, the greater the frequency count of a candidate unclassified sequence, the greater evidence that this sequence is not an artifact, but a novel miRNA element.

The exact frequency count of a read is the number of exact copies in the data set. Non-exact redundancy will exist in the set as well, a product of cloning or sequencing errors, and/or biological variation. Sequences should be grouped together if they represent the same biological sequences within the data set. We call such a grouping a sequence family. The read with the highest exact frequency count in the family is called the consensus sequence. The entire family is represented by the consensus sequence.

3.5.3 Sorting phase: mirSorter

The sorting phase, implemented by the python function mirSorter, eliminates exact duplicates from the set and computes the exact frequency count for each resultant sequence. It works by first sorting the reads by their nucleotide sequences, and then collapsing this sorted list into a set of non-redundant sequences. In our experiments, between 40 and 90 percent of the sequence reads are redundant. The resulting sequences are then sorted by their frequency count. The entire step can be performed on the order of $n \log n$ running time (n being the number of the sequences), when using a standard comparison based sorting algorithm. mirSorter uses a python implementation of quicksort to sort the reads. The performance of this simple solution is sufficient for standard sizes of input (350-500K reads). In the case of much larger data sets, this step could be improved by using radix or counting sort.

3.5.4 Reduction phase: mirReducer

The reduction phase, implemented by mirReducer, groups sequences by a more relaxed matching criteria than the previous phase. Sequences are grouped into sequence families if they differ by no more than a fixed number of nucleotides, ignoring length differences. This accounts for minor sequencing errors and imprecise processing in the 3' end of potential miRNAs. The number of nucleotides is a user defined value, defaulting to zero. When set to zero, two sequences are grouped if one is a precise subset or superset of the other. In the event of ambiguity, a sequence is grouped with the matching sequence of highest frequency count.

Groups are formed by taking the top sequence and comparing it to all other sequences in the list. Comparisons are made by aligning the two sequences with a modified global alignment. The standard global alignment is modified by ignoring end gaps, using a scoring scheme of 1 for a match, and 0 for either a mismatch or gap. Using this approach, the number of different nucleotides will equal the length of the shorter sequence subtracted from the score of the sequence alignment. This score is compared to the cutoff value, and if the difference in nucleotides is less than or equal to the cutoff, the sequences are grouped. When the top sequence has been compared to every other in the list, it is removed along with the matching sequences to form a sequence family. The top sequence becomes the consensus sequence of this new sequence family. This process is repeated until no sequences remain.

When all the groups have been formed, the consensus sequences (top sequences) from all groups are taken to form a new, ‘reduced’, list. The groups are kept as a record of the reduction step, allowing the complete reconstruction of the reduced file.

In worst case, if every sequence is a group of one and must be compared to every other sequence lower in the list, this step will run in $\frac{1}{2}n^2$ time. The algorithm improves as redundancy in the sequences increases, reducing the number of sequences that need to be compared in subsequent iterations. In our experiments, between 10 and 30 percent of the sequences from the previous phase are removed from the set with a 0 mismatch cutoff (ignoring length differences).

3.6 Identification Module

Identifying known elements is the third step in the miRNA discovery work flow. The identification module combines standard homology search techniques, customizable similarity criteria, and database search tools to provide a framework for large scale homology based sequence classification. First, similar sequences are found via database search, and then homologies identified by sequence comparison using customizable similarity criteria. This methodology is effective at identifying sequences within the same species, across closely related species, and in the case of highly conserved sequence elements, such as miRNA, among even distantly related species.

The identification module in mirMiner provides users tools to annotate known sequences. mirIdentity identifies known miRNA sequences or novel members of known miRNA families by comparison with a database of known miRNA. mirFilter extracts known non-miRNA elements by interfacing with a more general database. This two-step process provides the ability to quickly identify known miRNA similarities (inferring homology) by comparing sequences to the relatively small number of known miRNAs. The time consuming step of searching a general database is saved when it is necessary to find known non-miRNA elements by comparison to a much larger set of sequences.

Both tools use the similarity search tool BLAST to identify similar sequences in databases of known elements (such as the miRNA Registry [33, 34, 35] and NCBI's 'nt' database), and evaluate hits by user defined criteria to determine classification.

3.6.1 Goals and Challenges: automated annotation

The identification module seeks to appropriately annotate each sequence in the data set as a member of a known miRNA family, member of a class of known non-miRNAs, or a potentially novel miRNA. The challenge is to eliminate as many sequences as possible, without erroneously misclassifying a novel miRNA.

3.6.2 mirIdentity

In mirIdentity, the default criteria for validating a BLAST hit are based on the biology of miRNA. It sets three conditions to validate a 'hit', based on an un-gapped alignment of the potential-sequence and the hit sequence starting at the 5' end,

```
>EVSY06X02GMYKY length=22 species=Neries expression_level=9
TATCACAGCCAATTTGACAGCC
>aga-miR-13b MIMAT0001502 Anopheles gambiae miR-13b
UAUCACAGCCAUUUUGACGAGU

# TATCACAGCCAATTTGACAGCC
# ||||||||||| ||||||
# UAUCACAGCCAUUUUGACGAGU
# ----------------------
```

Figure 3.6: A potential miRNA sequence is deemed homologous to a known sequence if matches displays three criteria; 1. Seed Sequence Similarity, 2. within 2 nts of length, 3. no more than 3 mismatches when ignoring the last 2 nts on the 3' end.

1. Length Similarity: Sequences length must match within two nucleotides

2. Seed Sequence Identity: The seed sequence, position 2-7, must be identical

3. Non-Seed Sequence Similarity: The remainder of the sequence may contain no more than 3 mismatches

These criteria are depicted in Figure 3.6.

A blastn search is performed against a local database of known miRNAs. The parameters of blastn may be modified by the user, defaulting to the recommended values for short nucleotide sequences [36].

Each hit is evaluated by the three conditions previously described. If these are met, the hit is considered a match to the potential sequence, and the sequence is removed from further steps in MirMiner. A list of matches, containing the potential sequence, the hit miRNA, and the alignment, are saved together in a file for verification by the researcher. Researcher verification is important as data sets may contain contamination, the presence of sequences from other organisms, or other problems.

3.6.3 mirFilter

mirFilter acts as a parser on output from a large set of BLAST queries against a general database, such as the NCBI nucleotide database, nt. When given the XML file from a BLAST search, and the set of potential sequences, mirFilter will annotate the set of sequences based on the analysis of hits. Since the calculation of e-values is unreliable for

```
>EVSY06X02G7L4U length=23 species=Acoel expression_level=31(piRNA)
TGGTTCAGTGGTAGAATTGTCGC
#(piRNA) 0.956...% id : >...piRNA piR-64143 (14 total hits in class at 6.73547 avg e-value)
#(tRNA) 0.956...% id : >Zebrafish DNA sequence ... (10 total hits in class at 6.73547 avg e-value)
#(mRNA) 0.956...% id : >...  mRNA (3 total hits in class at 6.73547 avg e- alue)
#(oocyte_piRNA) 0.956...% id : >...oocyte_piRNA3732...(2 total hits in class at 6.73547 avg e-value)
```

Figure 3.7: mir-Filter will read the description of the BLAST hits and decide on a RNA class for each sequence in the hit list. Then, after counting summary statistics for each groups, makes a final determination as to the sequence identity. Summary statistics are highlighted in orange, and the annotation in blue.

short sequences [36], it is an insufficient measure of sequence identity. To compensate for this, a five stage procedure is used, mitigated by a threshold (percent identity).

1. Pull out all hits with the phrase 'RNA' in the annotation, classify them by the complete word containing the key phrase (tRNA, mRNA, rRNA, ect), and divide the hits into groups by this classification.

2. Within each group, sum the number of base pair matches and the e-value scores, and choose a consensus sequence as the sequence with greatest percent identity to the sequence-read. If the percent identity is greater than the cutoff, the classification of the lead candidate is the proper annotation for the unknown sequence.

3. Return the annotation of the winning group, and the consensus sequences for the remaining groups, ranked by their sum base pair score.

An example classification is shown in Figure 3.7.

By default, mirFilter uses megablast against a local database to generate hits, using a wordsize of 14. Megablast was chosen for performance benefits over blastn when searching highly for similar sequences. These optimizations are practical when using larger general databases such as NCBI's nt database (approximately eight gigabytes, compared to approximately 600 kilobytes for all known miRNAs). Alternatively, mirFilter may work directly from a user provided xml file generated from any nucleotide BLAST algorithm. If desired, mirFilter can also interface with the online BLAST NR nucleotide database hosted by NCBI [36].

3.7 Analysis Module

The analysis module provides a set of tools for the analysis of sequences left unclassified by the identification module. Two tools are provided to examine sets of unknown sequences. mirCluster identifies inter and intra species conservation, and mirStar identifies probable miRNA:miRNA* pairs.

3.7.1 Goals and Challenges: predicting novelty

Identifying miRNA-like properties in the unknown sequences allow researchers to prioritize their effort. Unknown sequences conserved across multiple species, or those identified as part of a probable miRNA:miRNA* duplex structure, are more likely to represent novel miRNAs.

Both of these tools are geared towards to the investigation of legitimate miRNA, and their effectiveness is challenged by the presence of non-genomic, and non-miRNA elements in the set of unclassified sequences.

Non-genomic elements are those sequence reads which did not originate from the genome of the host organism. Whether introduced by contamination in the library, or by sequencing or cloning errors, these sequences intermingle with real sequences and can mistakenly influence clustering, and create false miRNA predictions.

Non-miRNA elements which escaped the filtering stage also complicate analysis. While the end goal is to separate these non-miRNAs from novel miRNAs, their presence can also negatively influence clustering and create false miRNA predictions.

3.7.2 mirCluster

mirCluster groups sequences into clusters by a similarity function. The goal is to cluster matching sequences across multiple species as well as within the same species. Cross-species clustering provides a discriminator to focus novel identification efforts. Intra-species clustering serves as a final step to reduce redundancy within the data set, flagging groups which may represent the same sequence, saving redundant effort. A list of matching sequences across all given species is reported, as well as a list of unique sequences divided by species. Sequence clusters including only one species are placed in the unique list for that species.

mirCluster works in the same fashion as mirReducer, grouping sequences around those with higher frequency counts. Sequences from all species are handled together. The same identity function used in mirIdentity (seed sequence identity, length similarity, non-seed sequence similarity) is used to determine matches. Starting from the sequence with highest frequency count, groups are formed by matching the current sequence to all other sequences in the combined list. Subsequently, the species annotations for each group are evaluated. Groups containing sequences from more than one species are placed in a single file. Single species groups and sequences not contained in a group are organized by species. Like mirReducer, running time in worst case is proportional to $\frac{1}{2}n^2$ (where n is the total number of sequences) reduced by the number of groups that are found.

3.7.3 mirStar

mirStar identifies probable miRNA:miRNA* duplex pairs within sequences from a single species. It is based on a classifier which will be described in the next chapter. This classifier is used like the identity function in mirCluster to form groups around the highest expressed sequence, iterating through the list. Unlike mirCluster, the classifier assigns each member of the group a score, and grouped sequences are not removed from the list for remaining iterations (so the same sequence could appear in multiple groups). The sequence in each group with the highest score is evaluated against a reporting cutoff, and if the score is greater than this threshold, the pair is reported. Since no sequences are removed from the list at each iteration, the running time is directly proportional to $\frac{1}{2}n^2$, without the assumed speedup gained from groupings as seen in mirReducer and mirCluster.

3.8 Usage

MirMiner may be used via a set of command-line utilities, or accessed directly through a python interpreter or script. Command-line utilities are controlled by means of a configuration file. These configuration files allow module behavior to be modified in a range of predefined ways, and input/output files to be set. When the python functions are accessed directly, greater control may be achieved by passing different function arguments. In cases such as mirCluster and mirReducer, custom similarity functions may be passed as arguments to achieve different clustering effects. Similar functionality is included in the identity module, where the function used to validate hits as annotations is a user defined option. Table 3.2 shows the different command-line utilities and python functions, with a short synopsis of each tool.

Table 3.2: MirMiner may be used via a set of command-line utilities, or accessed directly through a python interpreter or script. While he command-line tools are simpler, more control may be obtained through direct interaction with the underlying python implementation.

MirMiner	With the set of instructions provided in the configuration file, MirMiner will apply the user selected set of modules (processing, organization, identification, analysis) to FASTA input file and optional 454 quality file.
mirProcessing	This tool is implemented via the python functions tagFinder and mirParser. tagFinder will report the top X tags present in the data set, and mirParser will extract regions of interest from a set of sequence reads. From the command line, users can run one or both functions, organizing sequence reads by their tag sequence, either explicitly, or by finding them automatically with tagFinder.
mirOrganizer	The functions mirSorter and mirReducer will eliminate exact and non-exact redundancy in the data set. Similarity criteria are modifiable in the configuration file and by passing custom matching functions to the python functions. Outputs are the sorted and reduced set of sequences, provided with a 'linker' file that describes the modifications.
mirIdentify	This tool and the python function by the same name identifies miRNA homology against a database of known miRNA. Homology criteria are modifiable in the configuration file and by passing a custom match function to the python function. Databases may be reformatted, or provided as a FASTA file and automatically formatted in-progress by the tool. Outputs are the remaining unknown sequences and a match file that reports identified homologies.
mirFilter	This tool and the python function by the same name remove sequences matching known non-miRNA elements from the data set. A percent identity threshold is modifiable via the configuration file, and function parameters. This tool may use a local database, or interface online with a database hosted by NCBI. Outputs are the remaining unknown sequences, and a 'filtered' file containing the identified non-miRNA sequences.
mirCluster	This tool and the python function by the same name cluster sequences by a user specified similarity function. This function. Similarity criteria may be modified via the configuration file, or a custom similarity function may be passed passed as a parameter. A group file containing sequence clusters spanning multiple species is outputted, as are individual files for each species.
mirStar	This tool and the python function by the same name report a set of sequence pairs representing predicted miRNA:miRNA* pairs. Threshold values and processing options (such as frequency count cutoffs) are modifiable via the configuration file or as function parameters.

Chapter 4

mirStar

Conventional computational methods of miRNA finding and discovery require knowledge of the entire genome sequence, using this for cross genome comparison or to identify miRNA secondary structures. In the absence such information, we must rely on other approaches. We describe mirStar, a tool which identifies potential miRNA:miRNA* duplex pairs for a given set of un-annotated candidate miRNAs. Our program provides researchers a means to separate valid from erroneous potential miRNAs, without the need for extra genomic data.

A miRNA:miRNA* duplex pair is an important part of miRNA secondary structure, see Figure 4.1. It is here that the pri-miRNA transcript forms a bond that will not be broken until one strand is incorporated into the cellular machinery of RISC, and the other degraded. The strand which becomes a functional component is termed the mature sequence, and will target and bind a complementary mRNA, preventing it from coding for a protein. The other strand is called a star sequence (see Figure 2.3 for an overview of miRNA biogenesis). Finding a candidate miRNA:miRNA* pair is an additional strong piece of evidence for a true miRNA.

mirStar helps streamline the novel miRNA discovery process by identifying candidate pairs worthy of further investigation. Such findings can be used to justify and prioritize experimental verification, such as Northern Blot Analysis. The previous chapter described how mirStar was implemented and is used in MirMiner to identity miRNA:miRNA* pairs. Here, we describe the performance of the classifier, and how it was developed.

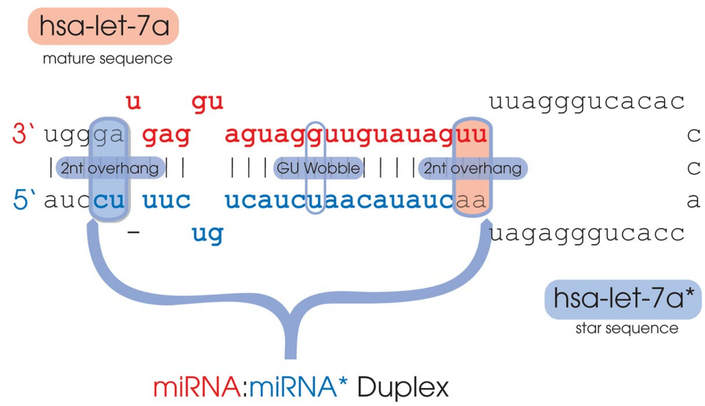

Figure 4.1: The miRNA:miRNA* pair forms a duplex within the pre-miRNA secondary structure following the GU Wobble model. It is conserved through to the inclusion in RISC, where one strand is subsequently degraded into a complex substrate, while the other incorporate into the RISC pathway. Many miRNA libraries sequence both strands of this duplex, with the mature normally expressed at a rate of 100 times that of its complement.

4.1 Approach

Inspired by sequence alignment, a classifier was built that utilizes dynamic programming to compute a score for candidate miRNA:miRNA* pairings, as an approximation of their potential MFEV (Modified Free Energy Value). The resulting score is then evaluated by a cutoff value based on known miRNA duplexes. Inputs to the scoring algorithm and classification cutoffs were inferred by a randomized Variable-Depth Local Search algorithm, using a training set of known miRNA:miRNA* duplexes and illegitimate pairs.

4.1.1 Background

Current computational approaches to novel miRNA prediction have centered either around whole genome analysis, like mirScan [21], or in the Modified Free Energy Value (MFEV) of the miRNA's predicted stem-loop structure, like mFold [23, 22]. Both these methods rely on extra genomic information, precluding their effective use on previously un-sequenced species.

Programs such as mFold work by determining biologically viable RNA folds. Folding is governed by stability of the RNA secondary structure, and can be predicted using dynamic programming. The stability of an RNA strand can be estimated using experimentally obtained binding energies for the RNA bases A, U, C, and G. By minimizing the 'free energy' in a RNA strand, the most probable RNA fold may be recovered.

RNA folding is used to find viable miRNA stem-loop structures, the existence of which is sufficient for the classification of novel miRNA. It is this scored stem-loop structure which yields the MFEV score used in the differentiation between potential miRNA and non-miRNA structures.

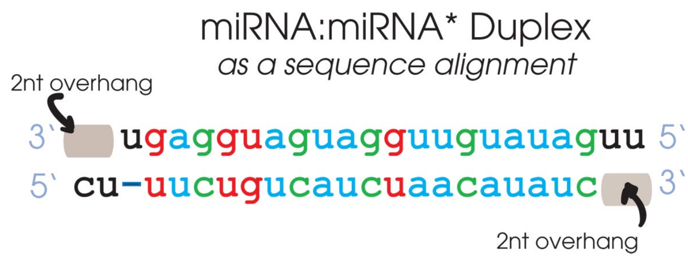

Figure 4.2: One may view the same sequences from Figure 4.1 as a sequence alignment, when instead of looking for exact matches, you look for matches as they pertain to the GU Wobble base pair model. These are $A \Leftrightarrow U$, $G \Leftrightarrow U$, and $G \Leftrightarrow C$

This is a well accepted method of verifying miRNA, but unusable if the complete hairpin is not available (such as when working solely with sequence data). In mirStar, we narrow the view of the predicted secondary structure to include only the miRNA:miRNA* duplex (Figure 4.2).

The miRNA:miRNA* duplex is unique when compared to the processing of other RNA molecules, and is the principle distinguisher between miRNA and other small classes of RNA. While normally degraded, star sequences do appear at lesser ratios than their mature counterparts in many sequenced libraries. When examining a set of unclassified sequences, the existence of a properly formed mature - star sequence pair yields high confidence such sequence is a legitimate miRNA, and not a degradation product or non-miRNA element.

When working with sequence data, we assume two sequences may form a duplex, and use dynamic programming to generate a score expressing the potential of such a pairing. In this way, an approximation to the MFEV is obtained without the need for the otherwise 'missing pieces' (the remainder of the stem-loop). We trained a classifier by seeking to maximize the differentiation between the scores of real miRNA:miRNA* pairs and false pairs using a randomized variable-depth local search algorithm.

4.1.2 Defining the Classifier

Our classifier computes an alignment score of two RNA strands. It uses this score as a metric to quantify the compatibility of these two strands to form a proper miRNA:miRNA* duplex, and infer classification as a valid miRNA.

For a given scoring matrix, a standard alignment is a set of base pairings that maximize the pairwise score of two sequences. To infer miRNA duplexes, we find an alignment scheme that differentiates valid from erroneous miRNA:miRNA* pairs, by producing higher total pairwise scores for legitimate pairs than false.

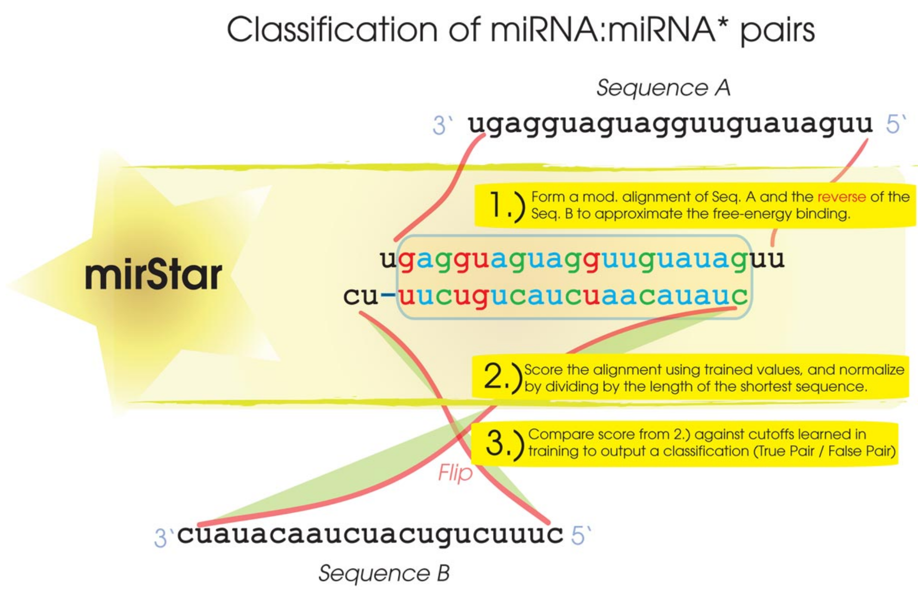

Figure 4.3: The classifier for mirStar uses a modified global sequence alignment, and a scoring matrix learned through training, to obtain an approximation to the modified free energy binding value. This approximation is compared to cutoffs learned in the training phase to produce a prediction.

To accomplish this, we implement the classifier as a modified version of global sequence alignment (see Background Chapter), using a dynamic programming approach to calculate an alignment score. We normalize the score by the length of the longest sequence, and aligning one sequence with the reverse of the other (See Figure 4.3).

DNA sequence alignment is generally used to identify sequence similarity by matching up identical bases, and a simple match/mismatch scoring matrix used. Instead, we chose a substitution matrix (See Figure 4.1) such that specific base pairings may have higher or lower scores than their peers. Values for the substitution matrix were learned during training.

To classify a sequence pair, the score from the alignment is compared with a cutoff value learned during during training. Pairs whose score is above the cutoff are reported, and those below discarded. Reported pairs are ranked by their score. The cutoff value is defined by a percentage, representing the fraction of all 'true' pairs in the training set whose score is above the cutoff value. For example, a value of 60% corresponds to a raw score that when used as a cutoff will return 60% of the 'true' pairs in the training set. Smaller values correspond to higher cutoff scores, increasing the confidence returned pairs are valid, but potentially excluding other valid pairs with lower scores.

4.2 Training Schema

The success of using an alignment score as a classifier is dependant on the scoring criteria used, as it is the summed values of these parameters that are used to distinguish the true pairs from false. Using sequences downloaded from the MicroRNA Registry, an initial investigation was made by modifying a simple match/mismatch scoring scheme to reflect the GU-Wobble model. Matches (defined as $A \Leftrightarrow T$, $C \Leftrightarrow G$, and $G \Leftrightarrow U$) were given a score of 1, and mismatches and gaps -1. This simple scheme yielded a positive result, showing a discernable score differential between randomly selected sequences, and correct miRNA:miRNA* pairs (Figure 4.4).

Working from this idea, a more complex scoring scheme was developed using a nucleotide substitution matrix combined with gap open/extend penalties. Parameters were trained to differentiate legitimate miRNA:miRNA* duplexes from a background model using randomized variable-depth local search. Table 4.1 shows the learned scoring matrix, and Figure 4.5 shows the distribution of score densities using the same data set as the GU-Wobble test.

4.2.1 miRNA reference and test sets

Mature miRNA and miRNA* sequences were obtained from miRBase[33, 34, 35] release 10.1(http://microrna.sanger.ac.uk/sequences/ftp.shtml). A set of miRNA:miRNA* pairs were then prepared by matching annotation names. In the event of ambiguity, the sequences were not included, leaving 846 sequences, comprising 423 miRNA:miRNA* pairs, incorporating sequences from across all species. This formed the raw training set.

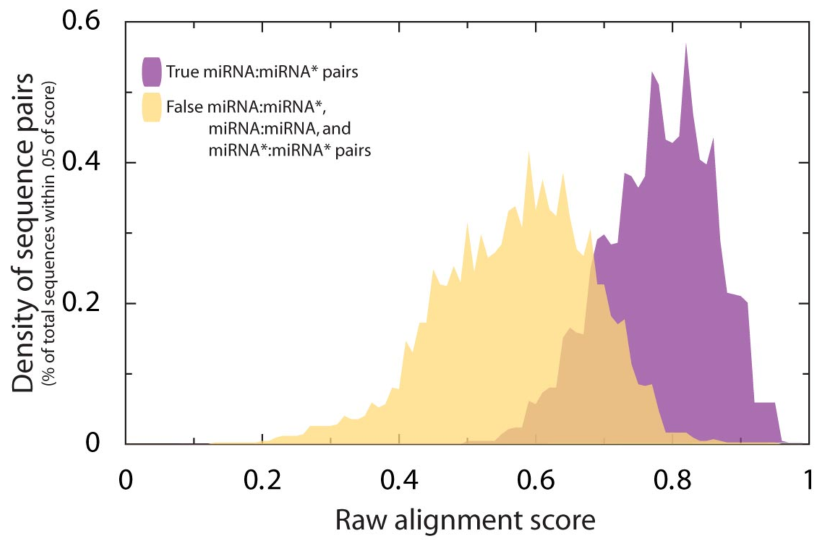

Figure 4.4: The idea of using an alignment score as a classifier was tested by applying a simple GU-Wobble based scoring scheme of 1 for matches and -1 for mismatches and gaps, to differentiate true miRNA:miRNA* pairs from false miRNA:miRNA* pairs, miRNA:miRNA pairs, and miRNA*:miRNA* pairs. Equal sized sets of true pairs and false pairs were scored, and the density of the scores plotted. The density of a score was calculated as the percentage of sequence pairs whose score fell with .05 of the target score. The data set was sampled in the interval 0 to 1 in steps of .01 to generate a continuous curve.

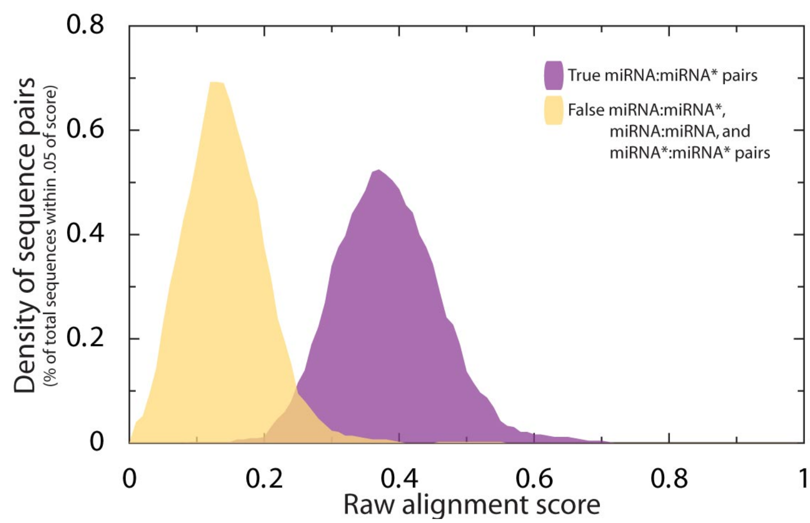

Figure 4.5: After training a nucleotide substitution matrix combined with gap open/extend penalties, the score densities of the training set using the learned parameters, Table 4.1, was plotted to compare it to the GU-Wobble test. The graph was generated as described in Figure 4.4.

The Match Set used in training consists of these 846 sequences in correctly paired form, and the Background Set was formed by taking a similarly sized set of random incorrect pairing among the raw training set. The background model consisted of a mixture of incorrect miRNA:miRNA* pairs, miRNA:miRNA pairs, and miRNA*:miRNA* pairs, covering all possible variations to the data among legitimate miRNA or novel miRNA sequences.

4.2.2 Implementation

The training schema and randomized variable-depth local search algorithm were implemented as Python functions, classes, and scripts totalling 364 lines. The `pairwise2` module from Biopython was used for sequence alignment, and the `random` module used as a random number generator.

4.2.3 Training

We use randomized variable-depth local search (RVDLS) to optimize our scoring function, made up of a nucleotide substitution matrix and gap open/extend penalties.

RVDLS works by randomly selecting candidate solutions, keeping those which improve the classifier, discarding those which do not. New candidate solutions are selected from a variable neighborhood of the last accepted candidate solution. Initially, the scoring matrix was set to zero (no discrimination). In subsequent steps, one value in the scoring matrix was shifted. The performance of the new scoring matrix is then evaluated. If performance increases, the new scoring matrix is accepted, otherwise, the old matrix is kept. In this way, incremental improvements to the scores were kept, and negative choices abandoned, resulting in a gradual improvement of the classifier. The range of the random shift is increased each time the candidate solution is not accepted. The range is re-tightened when an improvement is found. The ability of the classifier is measured by a two-stage evaluator.

Stage one of the evaluator measures the ability of a classifier to use a cutoff value to correctly differentiate miRNA:miRNA* pairs. Stage two is used as a tiebreaker for stage one. In the case where cutoffs are equally effective at discriminating the training data, stage two determines which scheme is ‘closer to’ increasing its stage one score.

To begin, the score of each miRNA:miRNA* pair in the training set is determined by application of the new scoring scheme. Using these scores, a cutoff value is used as a

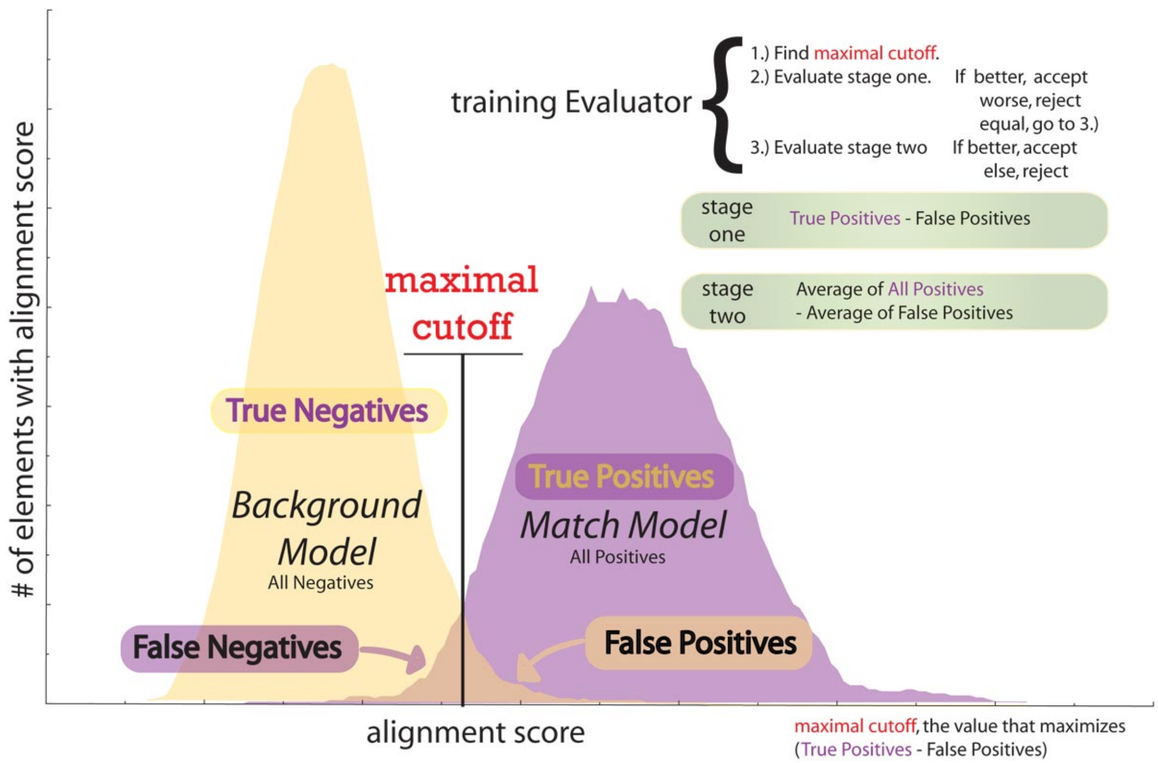

Figure 4.6: A two-stage evaluator was used to calculate the ability of the classifier to identify real pairings. The best cutoff value is calculated, and then the number of True Positives(TP)-False Positives(FP) is used as a metric to compare scoring schemes. If this number is equal, it examines the average of the match set to the average of False Positives. If the number of TP-FP was equal, stage two was 'won' by the classifier whose values were 'closer' to the next increase in stage one score.

binary classifier to predict true miRNA:miRNA* pairs. Scores above the cutoff value are predicted miRNA:miRNA* pairs, those below are predicted random pairs. In the training set, true miRNA:miRNA* pairs are present in the Match Set, and false pairs in the Background Set. The 'maximal cutoff' is the cutoff value that captures the most members of the Match Set and the fewest members of the Background Set. This is expressed as the number of True Positives (TP, members of the Match Set) - False Positives (FP, members of the Background Set). The resulting number (TP-FP) is the score of the 'maximal cutoff' and is used in stage one of the evaluator. If the maximal cutoff score of the new scoring scheme is greater than the maximal cutoff score of the previous scoring scheme, then the new scheme is accepted. If the score decreases, the new scheme is rejected. If the scores are equal, stage two is used.

In stage two, the average score of the Match Set is subtracted from the average score of all False Positives. An improvement in this score (over the previous score) means the new score, while not good enough to change the number of TP-FP, is moving the scores 'in the right direction', and is accepted. Otherwise, it is rejected.

RVDLS runs based on a set of convergence criteria, a set of values that informs the algorithm when it is ok to stop looking for a better solution. A common convergence criteria is the number of iterations performed. Other such criteria may include the number of iterations without an improvement.

Our algorithm was given an upper bound of 500 iterations, with the ability to end prematurely if 20 consecutive rounds do not yield an increase in the score of the classifier. In training, it converged in 57 steps. The progression in classifier ability during the training phase is shown in figure 4.7. At each step, the precision and recall of the classifier was determined when using the maximal cutoff as a binary classifier. The precision is the percentage of correct pairings reported out of the entire number (both true and false pairs) reported. The recall is the percentage of correct pairing reported out of total number of correct pairing in the entire set (all true pairings). Both precision and recall show a steady increase as training progressed, ending near 100% (a perfect score) for each metric. The scoring scheme determined is reported in Table 4.1.

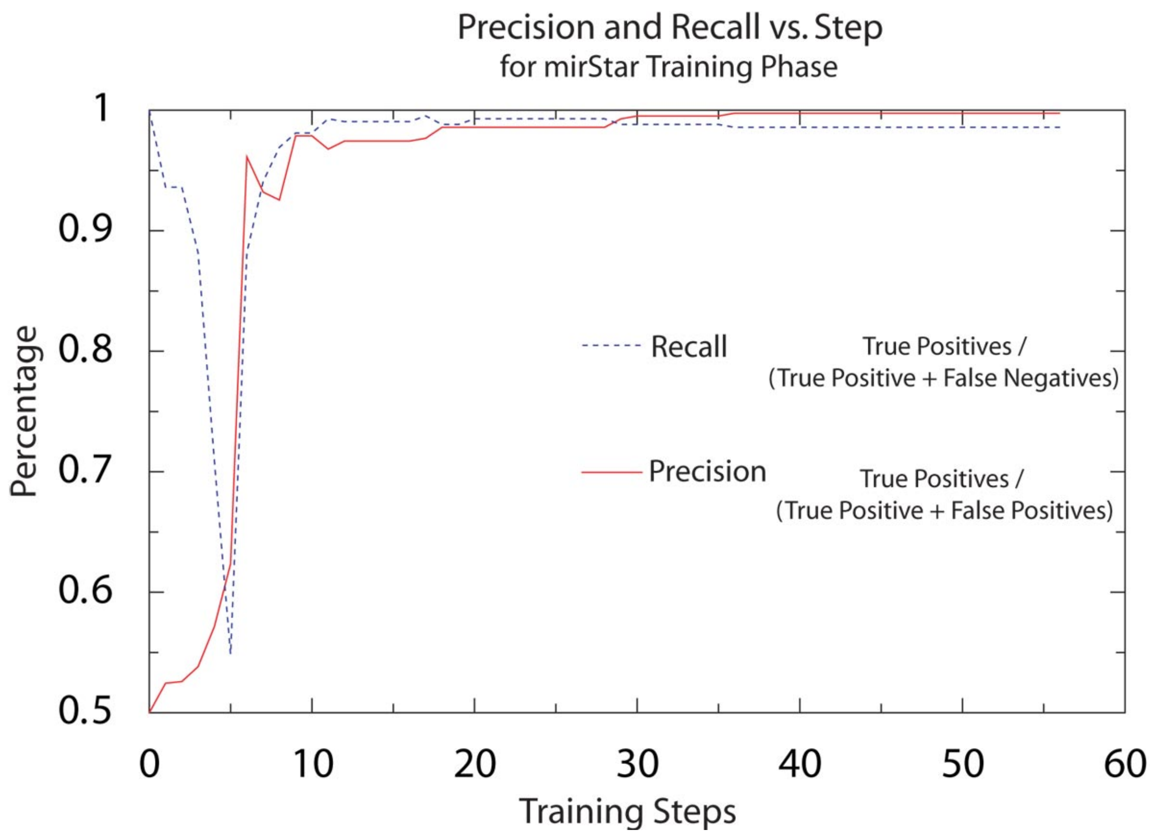

Figure 4.7: In training, mirStar converged in 57 steps. At each step the precision and recall the classifier was evaluated. By using randomized variable-depth local search, an improvement is seen as the algorithm 'learns' optimal values for the scoring matrix.

Table 4.1: We chose a substitution matrix and affine gap penalty to generate our alignment score. Values for the scoring scheme were learned during training. Parameters were trained to differentiate legitimate miRNA:miRNA* duplexes from a background model using randomized variable-depth local search.

	A	U	C	G
A	-1	.6	-.5	-.5
U		-.4	-.4	-.1
C			-1.3	.8
G				-2

	score
Gap Open	-1.2
Gap Extend	-.1

Table 4.2: A confusion matrix relates the true number of positives and negatives to the the predicted number of positives and negatives. The number of predicted true miRNA:miRNA* pairings and false miRNA:miRNA* pairings, associated with the real values, are reported as sums over all 10 testing sets for mirStar.

	Predicted Negatives	Predicted Positives
Negatives	413	10
Positives	8	415

4.3 Classifier Performance

The performance of the approach was assessed through the use of 10-fold cross validation, a technique whereby the raw data is divided into 10 equally sized subsets, alternatively training the classifier on 9 of these subsets and testing on the 10th. This process was performed once for each of the 10 subsets, providing an estimate of the performance of the classifier.

We evaluated our classifier using standard metrics calculated from the set of true miRNA:miRNA* pairs (Positives), and false pairs (Negatives). Those Positives which our classifier correctly labels are called true positives (TP). Negatives which the classifier labels as positive are called false positives(FP). Negatives which are correctly classified are called true negatives(TN). Positives classified as negatives are false negatives(FN). Together, these values are reported in the form of a confusion matrix, that shows the true number Positives and Negatives vs. the predicted number of Positives and Negatives.

For each of the 10 cross-validation test sets, the 'maximal cutoff' as determined in the training phase for that set was used to classify sequence pairs as true or false. To observe the performance of the classifier across all 10 sets, we formed a confusion matrix from the sum totals of TP, FP, TN, and FN across each set (Table 4.2).

Using the confusion matrix, key terms for evaluating a classifier may be formally defined as,

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ \text{Specificity} &= \frac{TN}{TN+FP} \end{aligned}$$

Informally, the precision tells how many things your classifier returns are 'correct', the recall tells you how many out of the entire number of 'correct' things you can expect to get, and the specificity explains how well the classifier correctly throws out 'incorrect' things. Precision and recall often have a give/take relationship. Any classifier can achieve 100% recall if it labels everything as 'correct'. However, in this case it will not achieve 100% precision (it would have also mislabeled all 'incorrect' elements of the set). Likewise, the classifier can be tuned to only return 'correct' elements (100% precision), but may in the process not capture the entire 'correct' set (lessening recall). This trade-off can be related to choosing the correct cutoff value for mirStar. In the training phase, the maximal cutoff (Figure 4.7) is chosen to optimize these two values. If the maximal cutoff is further to the right, the percentage of TP/FP will increase, but some TP will also be lost. If the maximal cutoff is further to the left, the percentage of TP/FP will decrease, but more TP will be captured.

The ability of a classifier can be measured without regard to cutoffs with a Receiver Operator Characteristic (ROC) curve. A ROC curve relates the recall of a classier to the inverse its specificity. It demonstrates the behavior of the classifier as the recall is increased from 0 to 1, or from returning no positives in the set to returning all positives in the set. Good classifiers will have curves in the upper left hand corner of the graph, showing that high recall rates can be achieved with a minimum of false positives. A naive classifier (such as a coin toss) would result in a flat curve splitting the graph from bottom left to upper right, representing no discrimination. Curves were generated for each of the 10 test sets used in Cross Validation. These are shown in Figure 4.8.

For ROC curves, the Area Under Curve (AUC) measure is a standard means of evaluating the quality of a classifier. A classifier whose AUC is greater than 90 percent is deemed excellent. The average AUC for mirStar during cross validation was 98.9 percent. The average AUC, precision, and recall for the Cross Validation are reported in Table 4.3. Precision and recall were calculated using the maximal cutoff for each training phase as the threshold for classification in the test set.

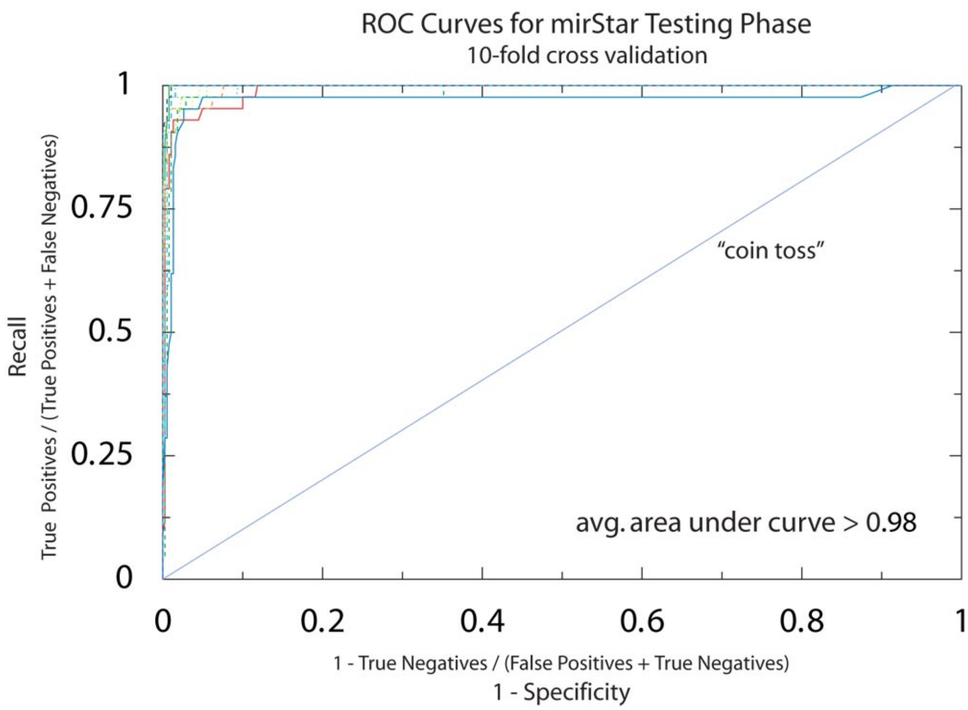

Figure 4.8: A Receiver Operator Curve (ROC) curve was generated for each of the 10 test sets used in Cross validation. Good classifiers will have curves in the far upper left of the graph, relating to the classifier's ability to obtain high recall with a minimal number of false positives. A common metric to compare classifiers is the area under their ROC curves (AUC). For comparison, if classification was based on a coin toss, no discrimination, the ROC curve would be a flat curve from the bottom left to upper right of the graph (the grey line). The AUC score of such a classifier is .5.

Table 4.3: The performance of the classifier was evaluated via 10-fold cross validation. The Precision, Recall, and ROC Curve (Recall vs. Specificity) ws calculated for each test set of 10-fold cross validation. The AUC (Area Under Curve) score for each classifier was calculated by the least square measure by dividing the graph into 100 intervals, forming a lower rectangle for each interval, and summing across all intervals. This provides a lower bound for each AUC measure.

Summary Statistics for 10-fold Cross Validation

	Average	Variance
Precision	99.7	1.95
Recall	98.1	2.06
AUC	98.9	1.17

Chapter 5

Results and Discussion

MirMiner was applied to a set of ongoing studies spanning 13 different species. By identifying known and discovering novel miRNA across these organisms, researchers hope to more accurately infer the evolutionary history of both the individual organism and their families.

MirMiner was used to process the raw sequencing data generated for these studies. It cataloged known miRNA and known non-miRNA elements, and produced a ranked list of potential miRNA for researchers to evaluate. To date, six of the species have been investigated, from which 341 known miRNA families and 101 novel miRNA families were identified.

5.1 Experiments

13 different species (see Table 5.1) are being investigated as part of an ongoing set of studies targeting miRNA. One study focuses on the evolution of miRNA in metazoan (animal) life, and the other is using miRNA as markers to determine the phylogenetic position of the lophophorates (organisms that possess a lophophore, a type of feeding tub).

RNA libraries were created for each of the species, and sequenced by 454 Life Sciences (Hartford, CT) in two separate runs as indicated in Table 5.1. Species barcodes, tags, were added to each library to allow simultaneous sequencing of multiple species.

From the set of raw 454 reads, known and potentially novel miRNA were extracted as described below.

First, 5' and 3' species identification tags and primers were removed by applying a

Table 5.1: MirMiner was applied to 13 different species across two 454 runs, encompassing two different ongoing miRNA studies. The scientific and common names (if available) are listed. Where convenient, species will be referred to by an abbreviation of their scientific name.

Run One	Run Two
Cerebratulus lactus(Ribbon Worm)	*Bugula neritina*
Saccoglossus kowalevskii(Acorn Worm)	*Glottidia pyrmidata*
Nereis diversicolor(Ragworm)	*Mytilus californianus*(California mussel)
Haliclona caerulea(Blue Caribbean sponge)	*Neocrania anomala*
Haliotis rufescens(Red Abalone)	*Phoronis architecta*
Symsagittifera roscoffensis(Acoel flatworm)	*Priapulid*
Hydra magnipapillata(Hydra)	

21 nt cutoff on either end of each sequence read. The resulting sequence reads were organized by species, based on the removed species tag, and a 17-25nt length cutoff enforced. Short or long reads, those reads without a matching 5' and 3' species tag, or reads whose average base-pair quality score did not exceed 20 were removed from the data set. Within each species, duplicate reads were eliminated, the number of duplicates annotated as the read's frequency count. In each non-redundant set, reads that were identical to reads with a higher frequency count when ignoring differences on the 5' and 3' end were grouped. From each such group, the read with the highest expression level was chosen as representative, and the others excluded from further analysis.

Known miRNA were identified by searching for similar mature and star miRNA sequences in miRBase [33] [34] [35] release 10.1. Blastn (version 2.2.27) was used to generate a list of hits, which were then evaluated by mirIdentity to extract known miRNA. The default parameters as described in Chapter 3 were used. Sequence reads which matched a known miRNA or miRNA star sequence within the above criteria were annotated and removed from the data set.

Known non-miRNA reads were removed from the dataset by comparison to NCBI's 'nt' nucleotide database. Megablast was used to generate a list of hits, and the hits for each sequence read were evaluated by mirFilter to determine appropriate classification, potential miRNA or known non-miRNA. Parameters were set as described in Chapter 3, and sequences annotated something other than a miRNA were removed from the data set. Annotations matching miRNA represented sequences similar to known miRNA, but which failed to meet the criteria of mirIdentity. These still represent viable novel miRNA, and were not removed from the data set.

Remaining sequence reads were investigated for intra-species and inter-species conservation and searched for potential miRNA:miRNA* sequence pairs using mirCluster and mirStar. Potential miRNA:miRNA* pairs were identified using a threshold of 60%. Default parameters (same criteria as mirIdentity) were used to group sequences. Reads conserved across multiple species were removed and placed in a separate file, ranked by the highest frequency count in each group. Reads not conserved across multiple species were ranked by frequency count and separated by organism.

Table 5.2: Six species were investigated for novel miRNA. They were chosen for the existence of a viable reference genome. Two species, *Hydra* and *Saccoglossus*, have complete genomes. For the remainder, the closest ancestor(s) were used.

species	reference genomes
Hydra	*Hydra magnipapillata*
Saccoglossus	*Saccoglossus kowalevskii*
Nereis	*Branchiostoma floridae*
Haliclona	*Capitella sp.*
Haliotis	*Ciona intestinalis*
Cerebratulus	*Daphnia pulex*
	Ixodes scapularis
	Lottia gigantea
	Nematostella vectensis
	Amphimedon queenslandica
	Strongylocentrotus purpuratus

To date, additional steps have been taken with six of the species to discover novel miRNA. These six species (see Table 5.2 were chosen for the existence of a usable reference genome. Genomic trace sequences for two species were available via the NCBI trace archives (performed at NCBI with default settings using the BLAST network service), Saccoglossus and Hydra. For the remaining sequences, trace archives for the closest ancestor were used. Using information from an ancestral genome allows the computational verification of miRNA gained up to that period of evolution. The reference genomes used are listed in Table 5.2.

Researchers used blastn to search reference genomes for the existence of potential miRNA sequence. MFOLD was used to generate candidate folds of all sequences with hits to a genome with a sufficiently low e-value (generally $< .1$). Sequences for which no genomic match could be found were ignored. Matches were annotated as novel miRNA if they met the structural criteria described by [24], reviewed in the Background chapter. Reported potential miRNA:miRNA* pairs have not yet been investigated to assist in discovering novel miRNA.

5.2 Results

In the first 454 run, 427,970 total sequence reads were obtained from six different libraries. In the second run, 368,366 more reads were generated from seven additional libraries.

From this set of 796,336 raw sequence reads, approximately one third (264,248 reads) contained a readable species tag whose insert was between 17 and 25 nts long. Of these, 82,269 sequences remained after 'redundancy' was removed with mirSorter and mirReducer. Applying the identification module, 2,454 unique sequences were identified as known miRNAs, and 17,584 as known non-miRNA. These results are tabulated and broken down by species in Table 5.3.

5.2.1 Automated annotation

In total, MirMiner automatically annotated approximately 80% of the raw sequence reads across the 13 species. However, the remaining 20% accounted for 60% of the total sequence diversity (the number of unique sequences) in the data. While MirMiner only identified about 40% of the unique sequence reads, the vast majority of these unidentified sequences were cloned only once (See Figure 5.1), making them unlikely to represent legitimate genomic inserts, but originating from cloning errors, sequencing errors, or contamination.

From the remaining sequences, mirCluster identified 2,892 inter and intra-species groups (a total of 8,032 sequences), reducing the complexity of the set of unknown sequences to 59,211 unique potential miRNA. 1,132 groups (a total of 3,910 sequences) showed conservation across multiple species.

5.2.2 Novel miRNA

Within the initial set of six species investigated for novel miRNA, researchers used computational techniques to examine all sequences with a frequency count of two or greater, a total of 3,043 potential miRNA. The investigation began with the 17 sequences conserved across at least two of the six species. In this small set, nine were verified as novel miRNA.

Table 5.3: The number of 454 Reads processed by MirMiner over 13 different species throughout the automated annotation and discovery process of two 454 Runs. ‘Reads’ are the total number of matching tags found. ‘nt’ are those sequences which passed the length cutoff of 17 to 25 nucleotides. ‘NR’ are the number of non-redundant sequences after exact redundancy was eliminated. ‘Red’ was the number of unique sequences remaining after near identity was eliminated. ‘miR’ are the number of unique sequences that hit a good match in miRBase 10.1. ‘gen’ are the number of sequences that were automatically filtered by MirMiner when compared to the NCBI nt database. ‘Unkn’ are the total number of unique unknown sequences after all automatic annotation as been completed. ‘Group’ are the number of sequences remaining unique to each taxon. ’NOVEL’ are the current number of novel miRNAs identified in the data set. Species with a blank in this category have not been investigated for their novel miRNAs.

Species	Reads	$17 < nt > 25$	NR	Red	miR	GEN	Unkn.	NOVEL
Acoel	40403	7039	2929	2028	13	231	1838	
Bugula	28189	26502	7228	5633	190	1956	3487	
Cerebratulus	60171	13609	4395	3602	232	405	2965	5
Glottidia	16816	31870	3320	2407	450	486	1471	
Haliclona	15514	11332	8275	6952	19	1847	5086	4
Haliotis	145505	37303	14417	10687	377	2925	7385	23
Hydra	7518	2441	1773	1498	13	251	1234	10
Mytilus	40039	19469	2982	2511	225	424	1862	
Neocrania	12854	11295	6423	4773	114	1672	2987	
Neries	54806	18337	7532	6181	308	602	5271	15
Phoronis	27722	10701	4827	3992	167	1273	2552	
Priapulid	54141	30122	28498	27203	178	2667	24358	
Saccoglossus	28261	20209	6076	4748	168	698	3882	57
Total	796336	264248	127140	82215	2454	19037	64378	114

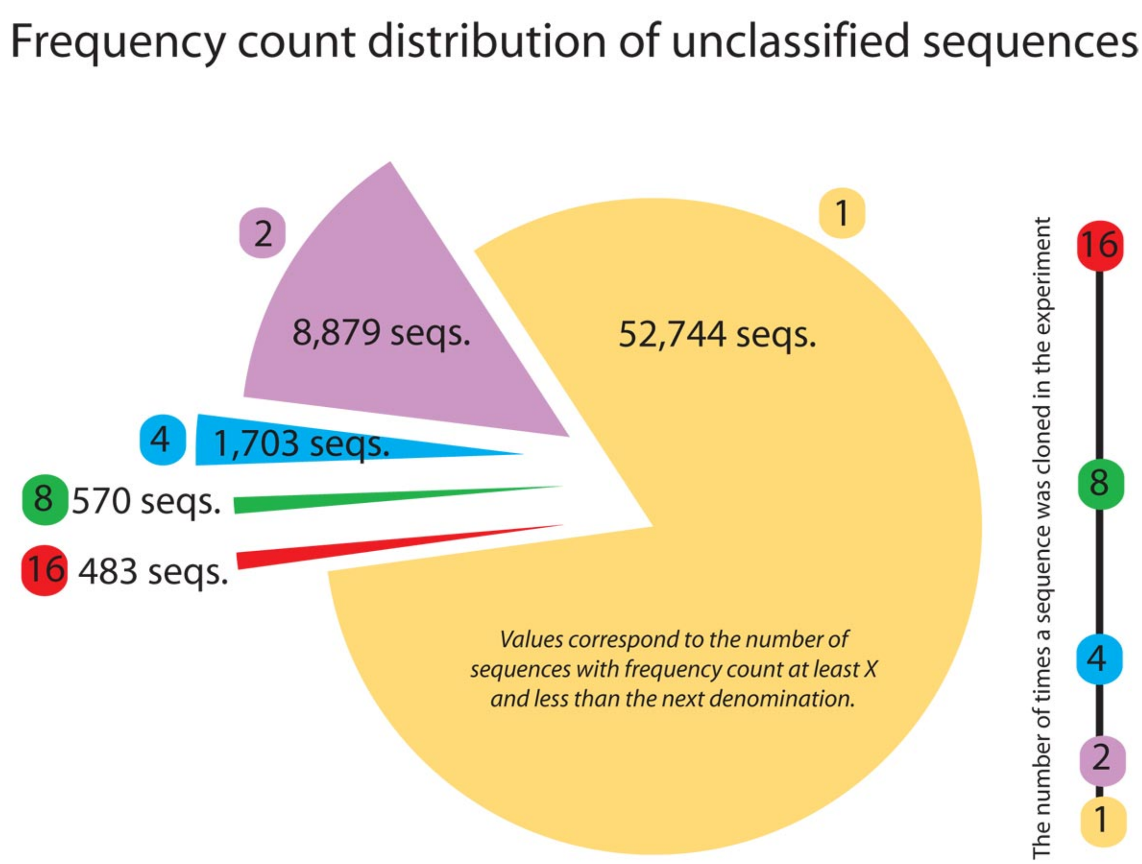

Figure 5.1: After automated annotation, 64,378 sequences remained unclassified. However, only a fraction of these were observed at more than trivial levels, suggesting the majority of these unique sequences are sequencing errors, cloning errors, or contamination.

Within the remainder of the sequences, uniquely expressed in their individual species, 92 additional novel miRNA families were identified. In this fashion, 114 sequences representing 101 novel miRNA families were discovered. Table 5.3 shows the number of novel miRNA identified per species.

5.2.3 Predicted miRNA:miRNA* pairs

mirStar was applied to each of the 13 species, using a threshold of 60% (raw cutoff score of .36). Candidate miRNA:miRNA* pairs were formed using sequences with a frequency count of at least four as the candidate mature strand, and sequences of lesser frequency counts as the candidate star strand. This condition was imposed to mimic the differences in observed expression between mature and star sequences found in sequencing libraries.

A total of 992 probable candidate miRNA:miRNA* pairs were returned, ranked by raw score. The distribution of raw scores is shown in Figure 5.4.

Probable pairs are impossible to verify without experimental tests or a reference genome. To evaluate the effectiveness of the classifier, hand annotations generated during the novel miRNA investigation were compared to the applicable set of probable miRNA:miRNA* pairs (393 total across six specie, encompassing 83 verified novel miRNAs) to identify verified miRNA and non-miRNA sequences.

Three miRNA:miRNA* pairs in three different species(all belonging to the same miRNA family) were identified during the novel miRNA investigation, Table 5.4, and mirStar correctly reported all three. In addition, 48 pairs were reported where one strand matched a verified miRNA, thus correctly identifying a novel miRNA. 31 reported pairs contained at least one sequence annotated as a non-miRNA, 21 pairs contained sequences annotated as not present in a reference genome, and the remaining 296 pairs were unclassified.

Taking into account only hand verified sequences, and counting all non-miRNA annotations as false positives (both non-genomic and verified non-miRNA), mirStar achieves a precision of 49.5%. A random guess within the same set of sequences (83 out of 393 verified as novel miRNA) yields a precision of approximately 22%. The recall of the classifier may not be calculated, even if restricting the view to verified miRNAs, given the presence of a miRNA:miRNA* sequence pair for all miRNAs is not a practical assumption.

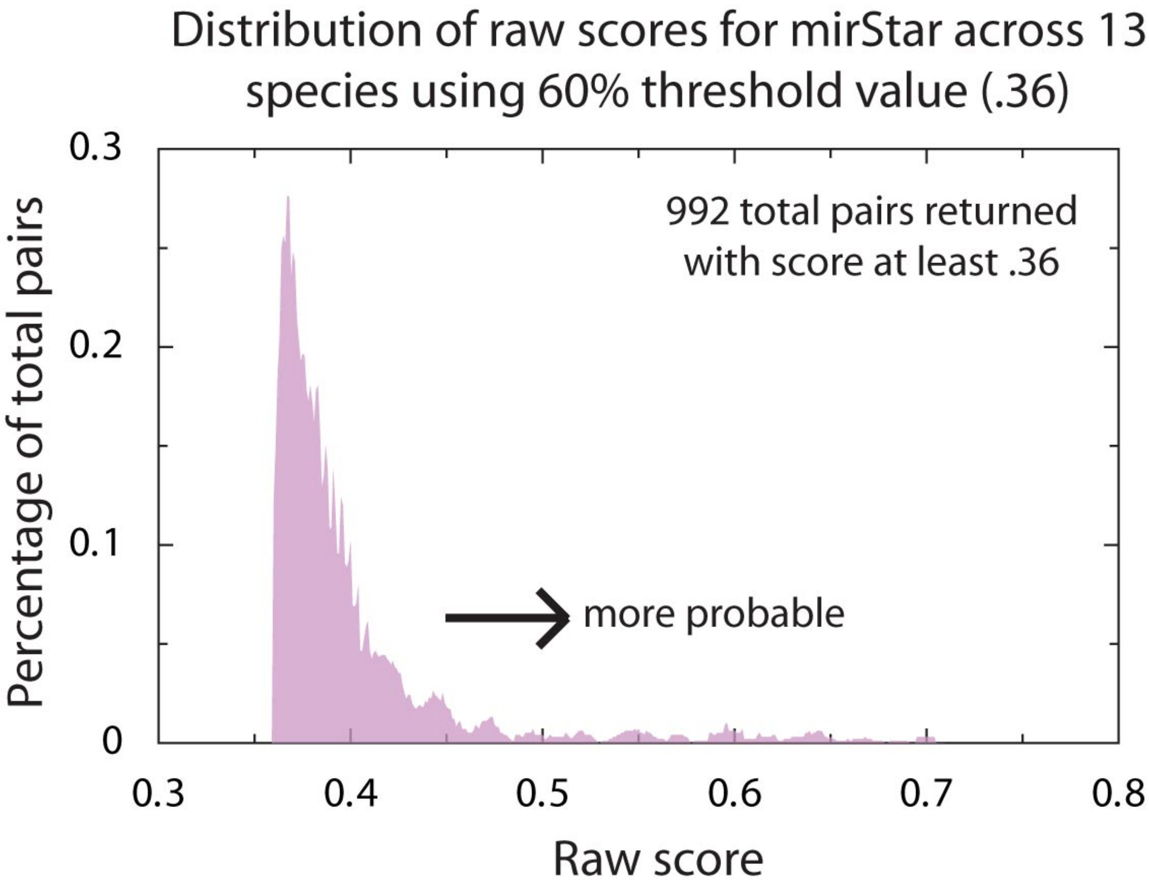

Figure 5.2: At at threshold of 60% a total of 992 candidate miRNA:miRNA* pairs were returned. Sequences with higher raw scores are more likely to represent legitimate miRNA:miRNA* pairings.

Table 5.4: Three miRNA:miRNA* pairs were verified during the novel miRNA investigation, a single novel miRNA family expressed in three different species. All three of these pairs were returned as probable candidates by mirStar.

species	sequences	mirStar raw score
Nereis	5'-UGAGAUUCAACUCCUCCAACUGC-3' 3'-AGUGGAGAGAGUUCUAUCUCAU-5'	.36
Haliotis	5'-UGAGAUUCAACUCCUCCAACUGC-3' 3'-AGUGGAGAGAGUUUUAUCUCAU-5'	.399
Cerebratulus	5'-UGAGAUUCAACUCCUCCAACUGC-3' 3'-AGUGGAGAGAGUUCUAUCUCAUC-5' *	.399

5.3 Discussion

MirMiner was developed in response for the need to automate the analysis of a prohibitively large set of potential miRNA sequences resulting from the previously described studies. Beginning as a in-house software tool, the opportunity to create a more flexible tool for a wider audience was realized when the tool needed to accommodate and combine the analysis of multiple sequencing runs. The goal was to produce a simple, customizable tool to reduce the number of sequences researchers needed to investigate by hand. In its current form, MirMiner is a set of python scripts that operate either from a Python Interpreter or through the set-up of configuration files passed to command-line utilities. Basic options are modifiable via the command-line, and more intensive changes may be introduced in the Python Interpreter.

Python was chosen for it's cross compatibility in all major computing environments, and relative ease of use when compared to other programming languages. This accessibility makes it feasible to expect non-computer science oriented users to take advantage of the flexibility implementation in an interpreted language provides. Additionally, program additions/modifications and updates are easily handled without requiring entire pieces of the program to be rewritten.

Programming in Python allowed development to start with the initial set raw sequence reads from the first 454 sequencing run, with little idea to the exact parameters or methodologies that would be used to effectively analyze the data. Collaboration with the researchers using intermediate results enacted a trial and error method to programming each module. This occurred as researchers gained a better understanding of their data throughout the development process, leading to better methods to identify redundant sequences, extracting known miRNA, ect.

5.3.1 Identifying the insert

The first hurdle was to identify and extract the correct set of inserts corresponding to the different species tags. In the first run, when a precise tag and primer parsing scheme was used, almost 90 percent of the reads did not match any of the six species. In contrast, 97 percent of the sequence reads had a discernable beginning species tag (first 4 nts).

While over 97 percent of the sequence reads in the first run had a discernable beginning sequence tag, only about 55 percent matched the exact pattern of tags and primers expected. Further analysis of the non-matching reads revealed significant variation in the primers and some variation in the end tags associated with species. Additionally, the majority of the primer variation seemed to occur on the 3' end of the sequence read. 802 different beginning primers were found when applying a straight 17 nt cutoff in the range of where the primer was expected to reside. In contrast, 4424 different variations of the end primer were identified. Of the beginning primers found, the vast majority were correct, with over 95 percent matching the expected sequence. Only about 77 percent of the all end primers found matched the expected. No explanation has been shown for the variation in primers or end tags, but is conjectured to result from sequencing or cloning errors that either prematurely chop the end of the read, and general sequencing errors that insert/delete or change the nucleotides in these sensitive regions.

5.3.2 Problems with identifying known miRNA

Identifying known sequences is perhaps the single most influential step in MirMiner. In the previous modules the goal was to produce the largest number of reasonably accurate sequences for further analysis. This step seeks to eliminate as many sequences as reasonable, classifying them as known miRNA or known non-miRNA elements. Accepted standards of miRNA structure exist, and the highly conserved nature of miRNA make the identification of known elements a generally uncomplicated procedure. Criteria are kept strict so as to avoid erroneous classification, as these identities represent the first real 'result' that may be used with little or no further analysis. However, it is noted that the presence of known miRNA homology within the data set does not mean it is truly present in the organism. Contamination, the infusion of genomic elements that do not originate from the host organism, can cloud the picture of known miRNAs 'found'. In only two (Haliclona and Hydra) of the 13 species the discovered complement of known miRNA investigated for validity. Haliclona and Hydra were singled out because in neither genome were known miRNA prevalent, making the identification of such known homologies contrary to accepted thought. On further investigation, none of the 'found' miRNA sequences could be identified in the reference genomes, marking these sequences most likely contaminants. A distinguishing feature of these contaminants were their relatively low frequency counts versus the known miRNA across the other species. The highest frequency count of a known contaminant was five, with the average frequency approximately two. In species for which a genome is available, validating known miRNA is merely an exercise in finding the sequence in the genome using BLAST or other tool. Verification is difficult in sequences without a genome, requiring the use of Northern Blots or other experimental techniques. Using the relative frequency counts of the contaminates in Haliclona and Hydra, it is hypothesized that experimental verification may be necessary only for sequences with a low frequency count (< 10).

5.3.3 Ambiguity when removing known non-miRNA elements

Extracting known non-miRNA sequences is the most ambiguous of the steps. In general, it is atradeoff between removing too much and removing too little. Initially, a straight E-value (as reported by BLAST) cutoff was used to remove homologies to NCBI's general nt database. However, this broad approach was rejected as sequences were being removed with matches to entire genomes, non-RNA elements, ect. Additionally, degradation products of mRNA or longer RNA molecules might receive a high E-value even if they were a perfect match, letting them slip through our filter. To correct this, the current filtering methodology was put in place, parsing the BLAST results to recognize only RNA molecules, and using percent identity instead of E-value. The question then becomes what makes a good percent identity cutoff, applicable to a wide range of different RNA molecules. A percent identity that is too low will result in a small set of unknown sequences, but potentially catch novel miRNA that by happenstance are remotely similar to a known RNA molecule in an unrelated species. Likewise, a percent identity which is too high will result in a large number of unknown sequences that need to be evaluated with time consuming sequence by sequence techniques. Within the examined data sets, our users preferred the 95% identity cutoff so as to avoid 'missing' any novel miRNA. It is suggested that a more relaxed criteria may be employed, and unknown sequences reported with a value that states how closely they matched a known RNA molecule. The unknown set of sequences may then be ranked inversely by this value, focusing attention on those molecules that have less similarity to known elements, and have a greater chance of representing novel elements. Additionally, elements that show some similarity to known elements would not be automatically discounted, but could be investigated after the initial set of high potential elements had been evaluated.

5.3.4 Identifying novel miRNA

The group of sequences present across multiple species is the best set of sequences in which to search for novel miRNA. In the data set, 17 conserved groups were identified. Of these, nine were annotated as novel miRNAs, two were degraded rDNA, two could not be found in the genome of the organisms, two were degraded transfer RNA, one matched another small class of RNA, and two could not be investigated due to insufficient genomic data.

Within the set of unknown sequences, frequency count appears to be the greatest indicator of a novel miRNA. This can be seen in Saccoglossus, where the greatest number of novel miRNA were identified, because a complete reference genome is available for computational verification of potential miRNA. All sequences with a frequency count of two or greater were verified by comparison to the genome, and the use of mFold to identify miRNA stem-loop structures. Of 268 total sequences reported by MirMiner, 227 were conserved in the genome, and 56 identified as novel miRNA. Figure 5.3 shows the percentage of the set identified as novel miRNA in Saccoglossus broken down by the minimum frequency count of the sequences. As the frequency count increases, so does the percentage of the number of novel miRNA identified.

5.3.5 Finding miRNA:miRNA* pairs

Correctly predicting miRNA:miRNA* pairs without the use of a reference genome has the potential to greatly expand the ability of researchers to study miRNA. Currently, a very real limitation is the lack of reference genomes for all but a handful of model species. In the absence of such a genome, identifying probable miRNA:miRNA* pairs give researchers an alternate tool to infer novel miRNA. Likewise, it can be used to double-check the annotation efforts of researchers to avoid annotating two novel miRNA that are really complementary strands.

When a reference genome is available, RNA folding programs such as MFOLD are clearly superior, however, if the verification process is not automated, mirStar may be used to prioritize investigations to focus effort first on those sequences most likely to represent novel elements. Without a reference genome, the additional information provided by mirStar can justify the use of a laboratory test.

The effectiveness of mirStar is difficult to gage given the inability to confirm or re-

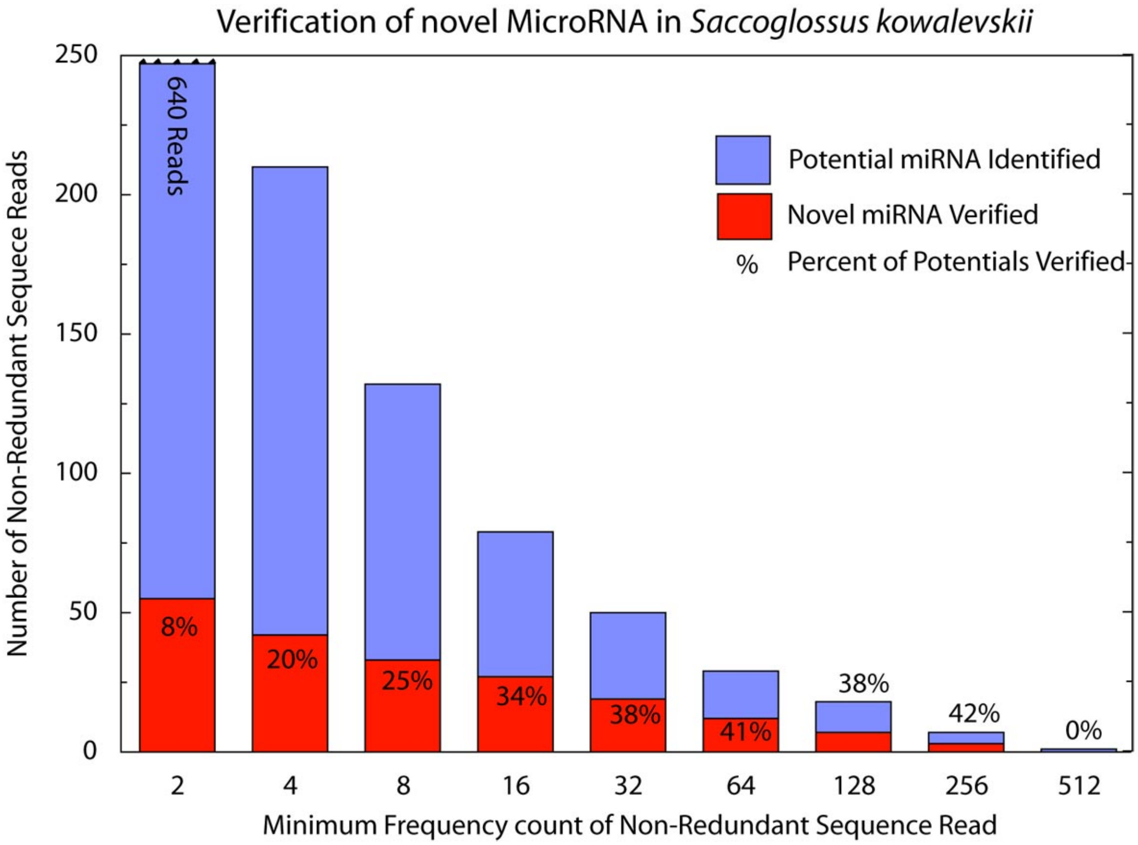

Figure 5.3: When investigating the set of unknown sequence returned from MirMiner, higher percentages of novel miRNA have been verified in sequences which appear more often in the data set. Saccoglossus was chosen as a model because the existence of a reference genome allowed researchers to completely analyze the entire set of potential sequences appearing twice or more in the experiment. A total of 57 novel miRNAs were identified.

ject the majority of candidate pairs returned by the classifier. In training, mirStar proved exceptionally well at distinguishing real miRNA:miRNA* pairs from erroneous miRNA:miRNA*, miRNA:miRNA, and miRNA*:miRNA* pairs. One challenge that is tough to address is the effect of non-miRNA and extra-genomic sequences on the classifier. When applied to the real data set, these sequences were treated as false positives for the purposes of evaluating mirStar, yielding a precision under 50%, much lower than the 98% seen in the cross-validation phase. While desirable to remove non-genomic sequences, this is impossible without a genome, and if a genome were available then mirStar would not be needed.

A tempting alternative use for mirStar is use it as an alignment tool to directly predict RNA folding structure of miRNA:miRNA* pairs. This is doable because mirStar does generate an alignment to determine the score for the classifier, however, it is mistaken to assume its suitability for such a task. Looking at alignments generated by mirStar, Figure 5.4 illustrates this fact.

Alignments returned by mirStar using learned scoring parameters

Raw Score = .5	Raw Score = .44
--UUAUAAUACAACCUGAUAAGUG	--UGUUGUAGUAGUUUAGCCCUGC--
UUAAUGUUAUGUUAGACUAUUC--	GCACAACAUCAUUAAAUC----CGGC

Figure 5.4: It is tempting to try and use mirStar to generate a candidate alignment of potential miRNA:miRNA* pairs, however, it is important to note the algorithm was trained based on the alignment score, and not the alignment itself. Both of the above alignments were generated by mirStar using real miRNA:miRNA* pairs, and were scored very well. As seen in the left hand alignment, for well behaved problems it will return a believable ordering, automatically adding the two nucleotide overlap on either end of the sequence. However, as seen in the right hand alignment, 'good' scores can still be achieved by alignments that do not strictly follow proper secondary structure guidelines.

The root of the matter is that the classifier was trained to learn scores, not structure. As regular structure is big part of miRNA biology, it is to be expected that alignments would tend in the direction of appropriate match-ups. Beyond this casual relationship, the alignment cannot be assumed to produce legitimate RNA secondary structures.

5.3.6 MirMiner is useful for more than miRNA

In addition to its designed task, MirMiner also proves useful for the general analysis of the raw sequence reads. Through straightforward modification of the regular expression for the processing module, different subsets of interest could be easily extracted from the total set of reads. For example, to find the consensus tag and primer sequences, the behavior of mirParser was changed to extract the tags and primers instead of the insert sequence. Simple scripts could than catalogue and summarize the results. This same technique was used to characterize the non-matching sequences and determine the quality of the sequencing libraries used in the experiment. For example, in both 454 runs, sequence reads with the tags GTGC (*Cerebratulus* and *Glottidia*) or CATG (*Haliotis* and *Saccoglossus*) showed an abnormally high number of 'primer dimers', sequence reads that contained no insert sequence and were composed entirely of tag and primer. Across all raw reads matching one of the 13 species, 76892 'primer dimers' were found. Over 99 percent (76463 reads) of these came from the tags CTGC (38558 reads) or CATG (37905 reads). Only 429 'primer dimers' appeared in all reads within the remaining 9 species (five different tags), highlighting a potential area of improvement in subsequent libraries. This same length analysis revealed the presence of what may be piwiRNAs (characterized as 30 nts in length) in some of the libraries, overshadowing the miRNA population. The length distributions for *Cerebratulus* and are shown in Figure 5.5.

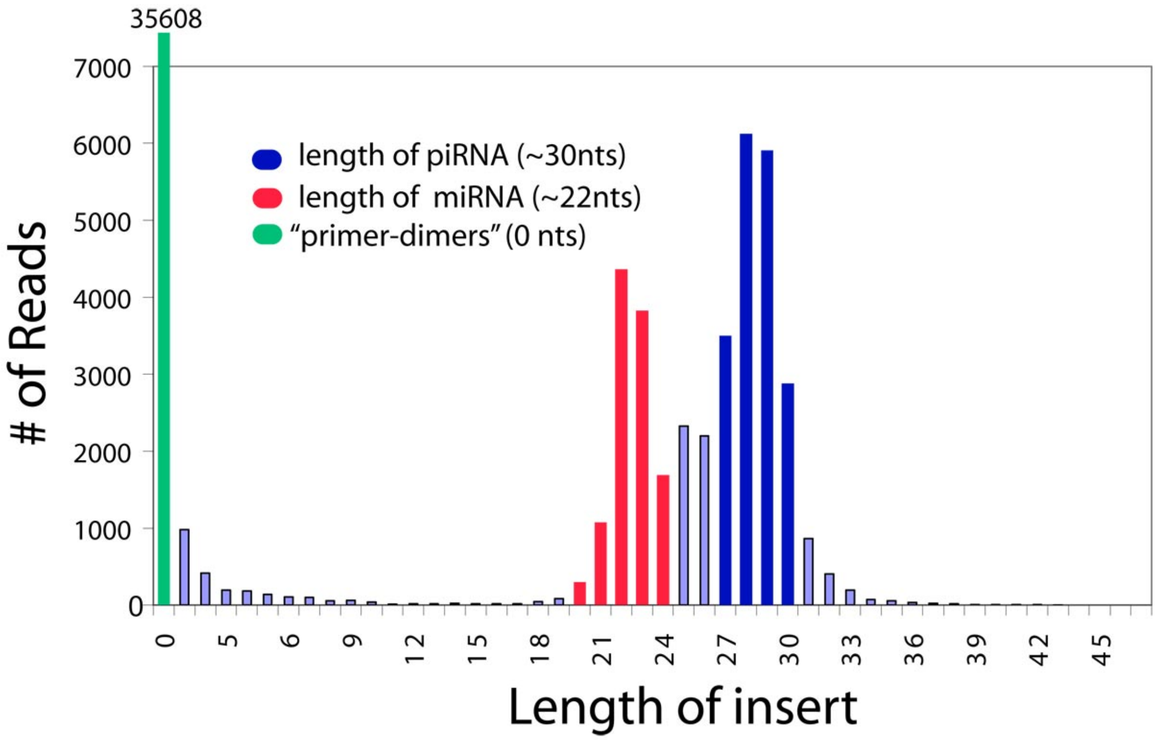

Figure 5.5: In addition to annotating and discovering novel miRNA, MirMiner can be useful in the general analysis of unclassified sequence reads. The mirMiner processing module was modified to capture reads of any length and summary statistics were calculated for the resulting reads. This distribution from *Cerebratulus* tells researchers two important facts. First, the results contain an extremely high number of "primer-dimers", sequence reads that consist solely of tags and primers. This infers a procedural in the sequencing or in construction of the library. Secondly, while the goal of library was to capture miRNA, the focus of the reads is closer to 30nts. Another class of RNA, piRNA, are about 30nts, and this distribution implies significant numbers of these elements may be present in the library.

Chapter 6

Conclusion and Future Work

The goal of this Thesis is to automate the annotation, analysis, and discovery of novel MicroRNAs (miRNA). A corresponding tool, MirMiner, was developed around and applied to two ongoing miRNA studies.

The first study is examining the evolutionary history of metazoan miRNAs, seeking clues to illuminate the role of miRNAs in organism evolution and complexity. A number of organisms across the evolution of eight different animal phyla were sequenced. Known and novel miRNA are being discovered in each species, and the acquisition and development of novel miRNA families is being traced back. The second study uses miRNAs as phylogenetic markers to infer the correct evolutionary position of the lophophorates (organisms that possess a lophophore, a type of feeding tub). MirMiner effectively processed data from 13 different genomes encompassing both projects. It automatically annotated 80% of the sequence reads, and ranked the remaining sequences for investigation by collaborators, directly facilitating the discovery of 101 novel miRNA families across six of the species.

In the majority of the studied species, little or no existing information previously existed regarding miRNAs. In some, such as Haliclona (the demosponge), miRNAs were conjectured not to exist [37]. Few of the organisms had genomic sequences available to computationally verify candidate miRNAs, forcing the reliance on ancestral genomes.

To address this, we developed mirStar, a tool to predict miRNA:miRNA* pairs. Such pairs can be used to infer with high confidence novel miRNAs from a set of unclassified sequences, without the need for a reference genome. To the best of our knowledge, mirStar is the only tool available to predict such pairs.

There is a lack of computational computational tools for the analysis of data resulting from high-throughput sequencing experiments. While new innovations are constantly developed to improve the speed and efficiency of sequencing technology, the same cannot be said of the supporting analysis techniques. MirMiner provides an adaptable tool for researchers to address analysis issues prevalent in large sets of sequence data within miRNA studies.

miRNA have been identified to play key roles in the regulation of cells within a variety of species, but many open questions exist. Many laboratories are focused on examining such questions in different organisms, and each such study brings with it separate challenges.

For example, in studies with flies at the NCSU Entomology department, more efficient and effective techniques are needed to identify common elements between different species. Researcher's are attempting to investigate the evolution of flies using miRNA as phylogenetic markers, similar to our lophophorates project. In flies, the majority of fly specific miRNA have only been examined in Drosophila [38, 39, 40]. This limited set of known miRNA diversity makes it difficult to effectively separate different species of fly via miRNAs. Either more novel miRNA need to be identified (made difficult by the lack of genomes for the majority of flies), or alternative methods need to be developed. In future work, we plan to expand MirMiner to look at other types of RNA molecules, and attempt to examine both known and unknown elements to infer the relationship between two sequenced species. Working from know elements other than just miRNA, such as piRNA or snoRNA, may provided the necessary descriptors to differentiate even closely related flies. Alternatively, the sets of known sequences may be ignored, and sets of unknown elements compared, cross-species groups used to infer phylogenetic relationships.

The Fungal Genomics Laboratory at NCSU is using high-throughput sequencing to study the open question of whether miRNA exist in the fungi kingdom. They have encountered severe quality problems in their sequence reads. Linker/primer sequences from the raw data do no follow a consistent pattern (being either absent, flipped, or only partially present) making correct extraction of the inserts difficult. Similar problems have been seen to lesser degrees (generally the dropping of one or two nucleotides from a primer/linker sequence) in all data sets we have had the opportunity to observe, implying this issue effects most experiments. Currently, MirMiner handles these issues by clustering similar sequences, assuming the most commonly occurring sequence to be the 'correct' copy. This is not sufficient in extreme cases such as with the Fungi study. We plan adapt the mirProcessing module to handle more pervasive quality issues, and extract inserts with a parser capable of separating primer/linker/insert regions by examining the nucleotides of the raw sequence read.

Also in the Fungi project, researchers need to automate the candidate miRNA verification/testing process. Naive methods of similarity based comparison to known miRNAs have proven ineffective, requiring an exhaustive analysis of all sequence reads. A reference genome is available, but there are too many candidates to individually check. Researchers face a similar hurdle with some species in the studies to which MirMiner was already applied. We plan to develop an additional module for MirMiner to automate sequence comparisons to either a genome or set of trace archives. This will be designed initially for Fungi and other current projects (metazoan miRNA, lophophorates), then generalized for application to a wider set of sequences/genomes/archives.

Finally, the need to manage large amounts of sequence data is not limited to miRNAs. While details may be different, the general framework of organizing sequence reads, identifying known elements, and prioritizing/reporting the resulting sequences could be applied to many types of genomic elements. The opportunity to create a general work flow environment, customizable to specific applications, exists to directly address the challenges of high-throughput sequencing technologies. We hope to use MirMiner as a framework from which to create such a work flow environment, providing a tool customizable to a variety of sequencing based studies.

Bibliography

[1] Anderson A Bernberg E Lu C Meyers BC Green PJ Markis M Isaacs G Huang E Morgan RW Burnside J, Ouyang M. Deep sequencing of chicken micrornas. *BMC Genomics*, 9(1):185, April 2008.

[2] Souret FF MuthuValliappan R Tej SS Poethig RS Henderson IR Jacobsen SE Wang W Green PJ Meyers BC Lu C, Kulkarni K. Micrornas and other small rnas enriched in the arabidopsis rna-dependent rna polymerase-2 mutant. *Genome Research*, 16(10):1276–88, October 2006.

[3] Sheridan R Sewer A Iovino N Aravin A Pfeffer S Rice A Kamphorst AO Landthaler M Lin C Socci ND Hermida L Fulci V Chiaretti S Fo R Schliwka J Fuchs U Novosel A Mller RU Schermer B Bissels U Inman J Phan Q Chien M Weir DB Choksi R De Vita G Frezzetti D Trompeter HI Hornung V Teng G Hartmann G Palkovits M Di Lauro R Wernet P Macino G Rogler CE Nagle JW Ju J Papavasiliou FN Benzing T Lichter P Tam W Brownstein MJ Bosio A Borkhardt A Russo JJ Sander C Zavolan M Tuschl T. Landgraf P, Rusu M. A mammalian microrna expression atlas based on small rna library sequencing. *Cell*, 129(7):1401–14, June 2007.

[4] Bartel DP. Micrornas: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–97, January 2004.

[5] Bracht J Pasquinelli AE, Hunter S. Micrornas: a developing story. *Current opinions in Genetics and Development*, 15(2):200–5, April 2005.

[6] Wang Q Cobb GP Anderson TA Zhang B, Pan X. Computational identification of micrornas and their targets. *Computational biology and chemistry*, 30(6):395–407, December 2006.

[7] Griffith M Kuchenbauer F Delaney A Prabhu AL Zhao Y McDonald H Zeng T Hirst M Eaves CJ Marra MA Morin RD, O'Conner MD. Application of massively parallel sequencing to microrna profiling and discovery in human embryonic stem cells. *Genome Research*, 18(4):610–21, April 2008.

[8] Altman WE Attiya S Bader JS Bemben LA Berka J Braverman MS Chen YJ Chen Z Dewell SB Du L Fierro JM Gomes XV Godwin BC He W Helgesen S Ho CH Irzyk GP Jando SC Alenquer ML Jarvie TP Jirage KB Kim JB Knight JR Lanza JR Leamon JH Lefkowitz SM Lei M Li J Lohman KL Lu H Makhijani VB McDade KE McKenna MP Myers EW Nickerson E Nobile JR Plant R Puc BP Ronan MT Roth GT Sarkis GJ Simons JF Simpson JW Srinivasan M Tartaro KR Tomasz A Vogt KA Volkmer GA Wang SH Wang Y Weiner MP Yu P Begley RF Rothberg JM. Margulies M, Egholm M. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–80, September 2005.

[9] Hayashizaki Y. Kawaji H. Exploration of small rnas. *PLoS Genetics*, 4(1):e22, January 2008.

[10] Ambros V Lee RC, Feinbaum RL. The c. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. *Cell*, 75(5):843–54, December 1993.

[11] Mallick B. Ghosh Z., Chakrabarti J. mirnomics-the bioinformatics of microrna genes. *Biochemical and Biophysical Research Communications*, 363(1):6–11, November 2007.

[12] Nilsen TW. Mechanisms of microrna mediated gene regulation in animal cells. *Trends in Genetics*, 23(5):243–9, May 2007.

[13] Sonenberg N Filipowicz W, Bhattacharyya SN. Mechanisms of post-transcriptional regulation by mirnas: are the answers in sight? *Nature Reviews. Genetics*, 9(2):102–14, February 2008.

[14] Stark A Russell RB Cohen SM Brennecke J, Hipfner DR. bantam encodes a developmentally regulated microrna that controls cell proliferation and regulates the proapoptotic gene hid in drosophila. *Cell*, 113(1):25–36, April 2003.

[15] Hay BA Xu P, Guo M. Micrornas and the regulation of cell death. *Trends in Genetics*, 20(12):617–24, December 2004.

[16] Sakai H Aukerman MJ. Regulation of flowering time and floral organ identity by a microrna and its apetala2-like target genes. *Plant Cell*, 15(11):2730–41, November 2003.

[17] Chen X. A microrna as a translational repressor of apetala2 in arabidopsis flower development. *Science*, 303(5666):2022–5, March 2004.

[18] Alvarez J Eshed Y Hawker NP Izhaki A Baum SF Bowman JL Emery JF, Floyd SK. Radial patterning of arabidopsis shoots by class iii hd-zip and kanadi genes. *Current Biology*, 13(20):1768–74, October 2003.

[19] Wu X Schommer C Schwab R Carrington JC Weigel D Palatnik JF, Allen E. Control of leaf morphogenesis by micrornas. *Nature*, 425(6955):257–63, September 2003.

[20] Sanseau P Brown JR. A computational view of micrornas and their targets. *Drug Discovery Today*, 10(8):595–601, April 2005.

[21] Weinstein EG Abdelhakim A Yekta S Rhoades MW Burge CB Bartel DP Lim LP, Lau NC. The micrornas of caenorhabditis elegans. *Genes and Development*, 17(8):991–1008, April 2003.

[22] Zuker M Turner DH Matthews DH, Sabina J. Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. *Journal of Molecular Biology*, 288(5):911–40, May 1999.

[23] Zuker M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–15, July 2003.

[24] Bartel DP Burge CB Carrington JC Chen X Dreyfuss G Eddy SR Griffiths-Jones S Marshall M Matzke M Ruvkun G Tuschi T Ambros V, Bartel B. A uniform system for microrna annotation. *RNA*, 9(3):277–9, March 2003.

[25] Havelda Z Va'rallyay E, Burgya'n J. Microrna detection by northern blotting using locked nucleic acid probes. *Nature Protocols*, 3(2):190–6, 2008.

[26] Coulson AR Sanger F, Nicklen S. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Science of the United States of America*, 74(12):5463–7, December 1977.

[27] Hall N. Advanced sequencing technologies and their wider impact in microbiology. *The Journal of Experimental Biology*, 210(Part 9):1518–25, May 2007.

[28] Schuster SC. Next-generation sequencing transforms today's biology. *Nature Methods*, 5(1):16–8, January 2008.

[29] Wunsch CD Needleman SB. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–53, 2008.

[30] Fitch WM Smith TF, Waterman MS. Comparitive biosequence metrics. *Journal of Molecular Evolution*, 18(1):38–46, 1981.

[31] Manderick B Hamelryck T. Pdb file parser and structure class implemented in pyhton. *Bioinformatics*, 19(17):2308–10, November 2003.

[32] Bassi S. A primer on python for life science researchers. *PLoS Computational Biology*, 3(11):e199, November 2007.

[33] van Dongen S. Enright A.J. Griffiths-Jones S., Saini H.K. mirbase: tools for microrna genomics. *NAR*, 36(Database Issue):D154–D158, 2008.

[34] van Dongen S. Bateman A. Enright A.J. Griffiths-Jones S., Grocock R.J. mirbase: microrna sequences, targets and gene nomenclature. *NAR*, 34(Database Issue):D140–D144, 2006.

[35] Griffiths-Jones S. The microrna registry. *NAR*, 32(Database Issue):D109–D111, 2004.

[36] Madden T.L. McGinnis S. Blast: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res.*, 32:W20–W25, 2004.

[37] McPeek MA Peterson KJ Sempere LF, Cole CN. The phylogenetic distribution of metazoan micrornas: insights into evolutionary complexity and constraint. *Journal of experimental zoology. Part b. Molecular and devlelopmental evolution*, 306(6):575–88, November 2006.

[38] Cohen SM Sandmann T. Identification of novel drosophila melanogaster micrornas. *PLoS ONE*, 2(11):e1265, November 2007.

[39] Parts L Brennecke J Hodges E Hannon GJ Kellis M Stark A, Kheradpour P. Systematic discovery and characterization of fly micrornas using 12 drosophila genomes. *Genome Research*, 17(12):1865–79, Decemeber 2007.

[40] Zhang MQ Li Y Wang X, Gu J. Identification of phylogenetically conserved microrna cis-regulatory elements across 12 drosophila species. *Bioinformatics*, 24(2):165–71, January 2008.