AN ABSTRACT OF THE THESIS OF

Daniel Patrick Smith for the degree of Bachelor of Science in
Bioresource Research presented on May 31, 2005.
Title: Identifying microRNAs by Automated Phylogenetic and
Structural Analysis.

Abstract approved:

_____

Dr. James Carrington

Motivation: It can take considerable amounts of time and effort to use
existing computational tools to analyze a small RNA sequence for the
purpose of determining if it is characteristic of microRNA. It is
similarly tedious to search for homologs to known microRNA
sequences in the rapidly growing collection of genomic libraries.
Results: Here I present a tool for automating the procedure used to
locate loci for microRNA genes. Located sequences form a hairpin
secondary structure and are conserved within the queried database(s).
Availability: http://rnadev.science.oregonstate.edu/mirpal

Identifying microRNAs by Automated
Phylogenetic and Structural Analysis

by
Daniel P. Smith

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Bachelor of Science

Presented May 31st, 2005
Commencement June 2005

Bachelor of Science thesis of <u>Daniel P. Smith</u>
presented on <u>May 31, 2005</u>.

APPROVED:

_____
Primary Faculty Mentor

_____
Secondary Faculty Mentor

_____
Director, representing Bioresource Research

I understand that my thesis will become part of the permanent
collection of Oregon State University libraries. My signature below
authorizes release of my thesis to any reader upon request.

_____
Daniel P. Smith

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

MicroRNAs (miRNAs) are a class of 21 – 24 nucleotide long small RNA molecules which are in part responsible for post-translational repression of messenger RNA (mRNA). When bound to an RNA-Induced Silencing Complex (RISC), miRNAs guide cleavage of specific mRNAs through perfect or near-perfect complementary binding with a region on the target mRNA. This regulatory action has been shown to affect epigenetic control relating to cellular development and differentiation. It has therefore become important to identify a species' different miRNAs and the mRNAs they target in order to more completely recognize and understand the developmental mechanisms that are present in that organism.

Initial studies to identify miRNAs relied upon chemical assays of expressed small RNAs in cells of varying tissue types. This approach successfully located many of the miRNAs that are present in the accepted set of miRNAs at the time of this writing. However, this approach is limited in that miRNAs expressed at low levels are not effectively revealed in these screenings. To overcome this limitation, computational techniques were used to locate genomic sequences that are characteristic of miRNA loci. More specifically, these are regions from which a transcribed piece of RNA could spontaneously fold back on itself to form a hairpin secondary structure – a motif common to all known miRNA precursors.

Prior studies have found that miRNA genes are evolutionarily well conserved, sometimes within an entire kingdom. Thus with the recent addition of the assembled genomes for *Oryza sativa* and *Populus trichocarpa* to the model plant organism *Arabidopsis thaliana*, researchers are now able to incorporate ortholog analyses as a means to further verify or refute miRNA genes that have been computationally identified in an individual species. In addition, comparisons of homologous miRNA genes across multiple organisms serve to date the appearance of those miRNA genes in an evolutionary timeline.

Together these two approaches are a powerful method for identifying miRNA genes with fast computer programs. However, such techniques are a slow and tedious process for researchers who must facilitate the transfer of sequence data between several different computer programs, repeat the procedure for each potential miRNA, and make manual adjustments on a case-by-case basis. To effectively utilize the potential of computational resources, data must be seamlessly transferred from the output of one program to the input of the next while at the same time automatically performing all the necessary checks and modifications.

To accomplish this task, the author has developed a computer program, called microRNA Phylogenetic Assembler (MiRPal), which efficiently facilitates the processing of small RNA sequences through a pipeline of several freely available

computer programs in order to locate miRNA genes using only a miRNA's name or sequence as the input from the user.

This program was developed with three primary uses in mind:

1) To find miRNA gene loci for known miRNA sequences in newly available genomes. In this case, the user may enter just the name of the miRNA to be tested. MiRPal then retrieves the mature sequences to use as input from the RFam microRNA Registry [3] via the Internet. Additional checks are performed to relate miRNA genes located by MiRPal to annotated loci.

2) To probe genomes for miRNA genes from which a candidate miRNA sequence could have arisen. MiRPal can help answer the question "Is this small RNA sequence a mature miRNA?" In an ideal case, MiRPal can produce strong evidence indicating 'yes' by locating well-conserved hairpin-producing regions from multiple genomes.

3) To generate phylogenetic trees for determining evolutionary relationships among microRNA genes. Complemented by bootstrap values, these graphs depict the degree of homology between an unannotated locus and a known locus. MiRPal also attempts to locate rooting miRNA loci from ferns, mosses, and liverworts as well as allowing the user to provide outgroups with which to root the phylogenetic tree.

It should be noted that this software does not locate mature miRNA sequences which are diverged from those that are entered by the user.

SYSTEMS AND METHODS

MiRPal acts to facilitate data processing through the pipeline shown in Figure 1.

The data are routed through a series of programs, listed in the right-hand column

and along the bottom. These programs must be installed before running MiRPal.
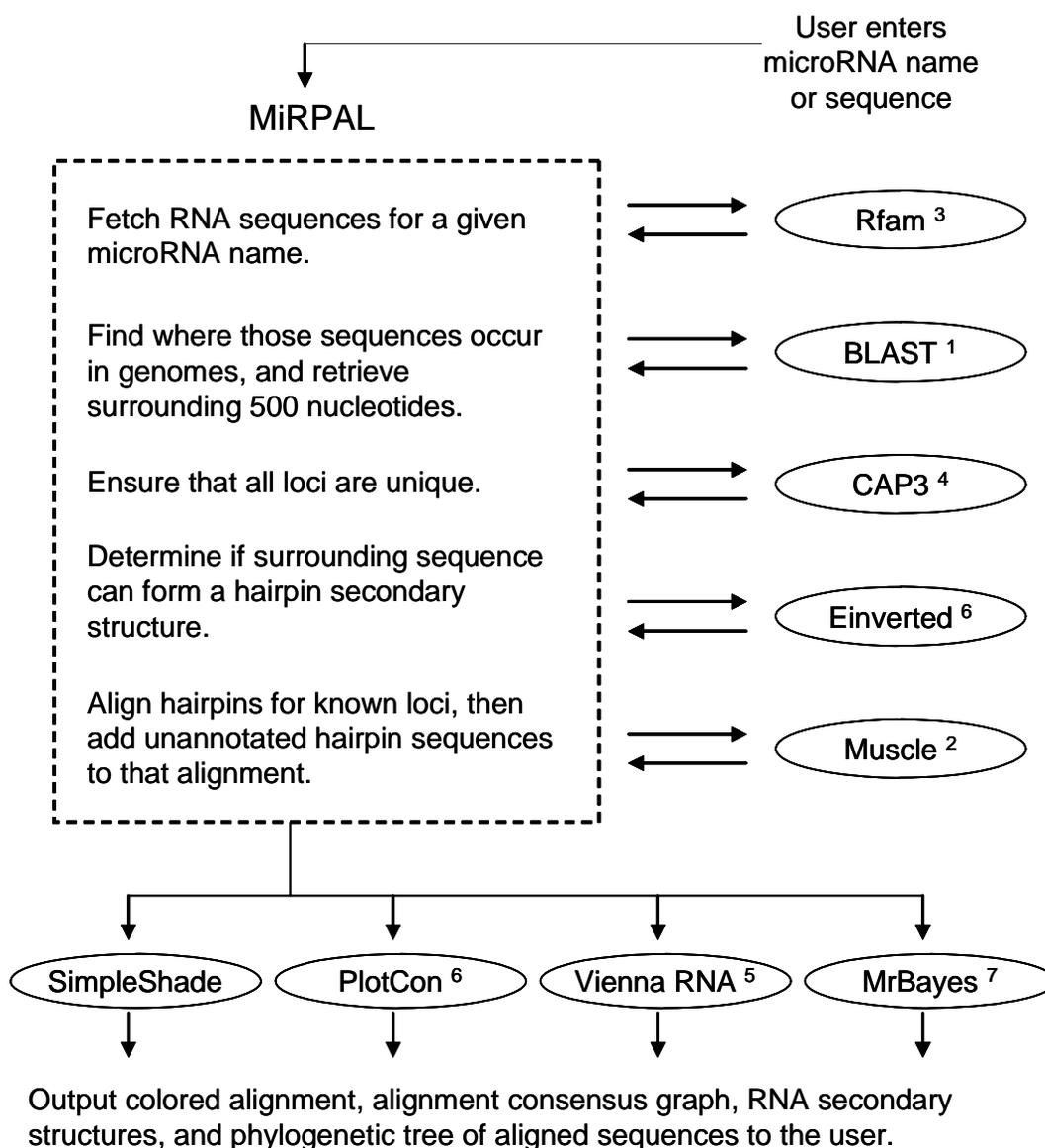
Figure 1. Overview of MiRPal's algorithm.

MiRPal is written in Perl and was developed to run under Linux, but can be modified to run in comparable environments.

Algorithm

The steps of MiRPal's pipeline shown in Figure 1 are a very high-level description of the program's methodology. At each step, MiRPal employs additional checks and optimizations to generate the best possible result set, as described below.

Sequence Retrieval from RFam

When the user enters a microRNA name, MiRPal locates all relevant sequences by the following method:

1. Loads the file specified by the config.ini file, by default ftp://ftp.sanger.ac.uk/pub/databases/Rfam/miRNA/mature.fa (all mature sequences in the RFam microRNA Registry).
2. Stores all sequences which match the user's specification.
3. Removes duplicates in the set of stored sequences.

Locate Mature Sequences' Loci with BLAST

The number of BLAST [1] searches performed by MiRPal equals the number of mature sequences being queried times the number of genomes specified by the user. That is, each sequence is queried against each genome.

Finding Unique Loci with CAP3

Because some genomes are not completely assembled, MiRPal ensures that duplicate loci are filtered out before proceeding by the method outlined below.

1. For every genomic region homologous to a mature miRNA sequence, the sounding 500 nucleotide sequence is retrieved using NCBI's fastacmd program.

2. The sequences found in step 1 are passed to CAP3 [4] where two or more perfect, or near perfect, overlapping sequences are merged together.

Locate Hairpin 2° Structure with EInverted

The 500 nucleotide sequence located with fastacmd in the previous step for each BLAST hit is sent to EInverted [6], a program that locates regions of a nucleotide sequence that may fold back on itself to form hairpin structures, based on scoring criteria. The parameters MiRPal uses for EInverted are:

match = 3, mismatch = -3, gap = 4, threshold = 45, maxrepeat = 350

In addition, MiRPal checks that the mature sequences are on the stem of the hairpin structure; mature sequences in the hairpin's loop are inconsistent with known microRNA genes.

When a single locus is found to contain more than one stem-loop hairpin structure where the mature miRNA sequence is on the stem, only the "best" stem is kept. The "best" stem is defined as the stem in which the mature miRNA sequence has the most base-pairings to its antiparallel, or miR*, sequence. This method can potentially select the incorrect stem-loop sequence. If this occurs then the stem-loop sequence could be failed at the alignment stage, and the locus lost to the analysis.

Loci that have no discernable hairpin sequence are excluded from further analysis. However, when no hairpin sequence is found by EInverted for a locus that is annotated in the RFam microRNA Registry (or other file specified by the user), then the hairpin sequence specified by RFam is used.

Alignment of Hairpin Sequences using Muscle

Muscle [2] is a multiple sequence alignment program used by MiRPal to align hairpin sequences. To generate the best possible alignment, MiRPal performs an alignment in two stages. In the first stage, all of the hairpins found with EInverted that have annotations in the RFam microRNA Registry (or another file specified in config.ini) are aligned all at once by Muscle. In the second stage, the remaining unannotated hairpins are added individually to the alignment from the first stage. After each addition, the alignment is scored. Additions are discarded when the new score is less than 96% of the old score.

The alignment scoring algorithm determines the average parsimony among all the sequences in the alignment. An example of the scoring procedure is shown in Figure 2.
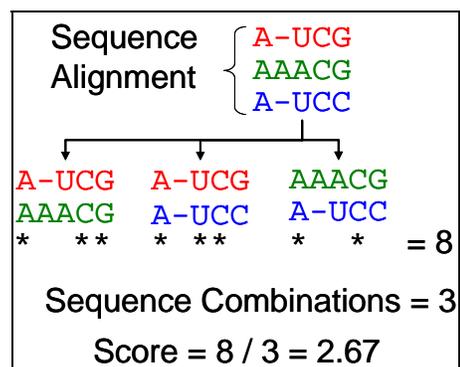
Figure 2. Alignment scoring example.

By this process, MiRPal is able to avoid including sequences in the hairpin alignment which greatly decrease the overall alignment.

Phylogenetic Analysis with MrBayes

MiRPal generates phylogenetic trees using MrBayes [7], a Bayesian approach to phylogenetic analysis of sequence alignments. MiRPal provides two additional pieces of information to MrBayes to optimize its analysis.

First, MiRPal passes the distance tree generated by Muscle to MrBayes to provide a starting point for tree generation. Without a starting tree MrBayes would begin with a random tree and require a longer burn in before beginning to generate meaningful trees.

The second piece of information MiRPal provides to MrBayes is an exclusion

mask of the hairpin alignment. All columns in the alignment where 30% or more

of the positions are gaps are marked for exclusion from analysis. This serves dual

purposes of decreasing the time required for analysis as well as filtering out noise

from non-conserved regions.

By default, MiRPal sets the following parameters for MrBayes:

Note: $sLen is the length in nucleotides of the sequences in the hairpin alignment.

```
charset bt1 = 1-$sLen\\3;
charset bt2 = 2-$sLen\\3;
charset bt3 = 3-$sLen\\3;

partition bygene = 3: bt1, bt2, bt3;
set partition=bygene;

lset applyto=(1) nst=6  rates=invgamma;
lset applyto=(2) nst=6  rates=invgamma;
lset applyto=(3) nst=6  rates=invgamma;
mcmc ngen=100000 samplefreq=10 nchains=4 savebrlens=yes;

sumt filename=run.nxs.t contype=allcompat burnin=500;
sump filename=run.nxs.p burnin=500;
```

In practice, the number of trees that MrBayes actually samples is largely

dependent on the number of sequences in the data set. In 11 runs mediated by

MiRPal, the number of resamplings ranged from 17 (using 6 sequences) to 4,995

(using 19 sequences). These sampled trees are not only used to create the final

consensus tree, but also to assign bootstrap values to each of its nodes. The

bootstrap value is the percentage of sampled trees that have an identical grouping

of sequences on either side of a given node.

Graphical Output

To allow the user to better visually analyze the results from an analysis, MiRPal

generates three types of graphical representations. First, it colors the hairpin

alignment using a built-in shading component called SimpleShade (example output

may be seen in Appendix I). This allows the user to scan through an alignment and

quickly find well-conserved regions. Secondly, MiRPal utilizes the Vienna RNA

software suite [5] to produce images of the hairpin sequences' secondary structures

(example output may be seen in Figure 6). Lastly, MiRPal uses PlotCon [6] to

create a similarity plot of the hairpins' alignment (example output may be seen in

Figure 4). All together, these visual representations allow a user to quickly

comprehend MiRPal's results and identify key elements that are of interest or need

to be removed from the analysis.

To facilitate access to the program, the author has also created a web-based

interface to accept input and present MiRPal's results in an easy-to-navigate
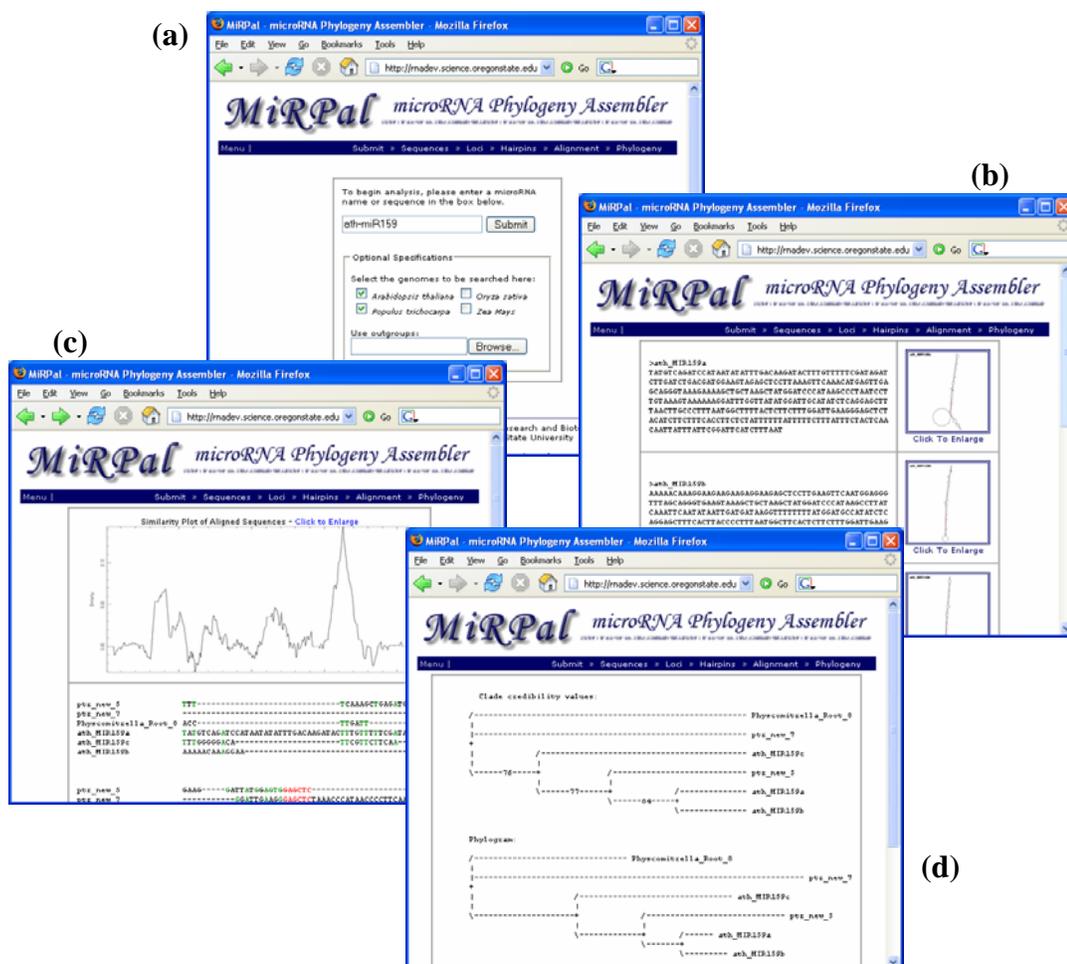
fashion.

Figure 3. Selected pages of MiRPal's web interface.

(a) Data entry by the user. (b) Genomic loci that meet fold-back criterion. (c) Alignment of hairpin sequences. (d) Phylogenetic tree.

This tool is currently hosted by the Center for Gene Research and Biotechnology

at Oregon State University and may be accessed from the URL

http://rnadev.science.oregonstate.edu/mirpal.

RESULTS

MiRPal was given the microRNA name "mir319" and told to use the genomic

databases for *A. thaliana* (ath), *O. sativa* (osa), and *P. trichocarpa* (ptr). It fetched

the mature miRNA sequences listed in Table 1 from the RFam microRNA

Registry and used them to ultimately find the potential miRNA gene loci listed in

Table 2. The alignment, similarity plot, phylogenetic tree, and hairpin secondary

structures MiRPal produced for these loci can be found in Appendix I, Figure 4,

Figure 5, and Figure 6, respectively. Total CPU time was 8min, 30sec.

Name	Sequence
ath-miR319a	UUGGACUGAAGGGAGCUCCC
ath-miR319c	UUGGACUGAAGGGAGCUCCU
osa-miR319a	UUGGACUGAAGGGUGCUCCC

Table 1. miR319 mature RNA sequences

Unique Name	Database	Accession	Start	End	Sense
ath_MIR319a	ath	NC_003075	12353119	12353138	sense
ath_MIR319b	ath	NC_003076	16677698	16677717	antisense
ath_MIR319c	ath	NC_003071	17036948	17036967	sense
osa_MIR319a	osa	lcl|9629	26601789	26601808	antisense
osa_MIR319b	osa	lcl|9629	6674362	6674381	antisense
ptr_new_6	ptr	LG_XIX	10852733	10852752	antisense
ptr_new_7	ptr	LG_XIII	12992460	12992479	sense
ptr_new_8	ptr	LG_III	12635218	12635237	sense
ptr_new_9	ptr	LG_I	7044536	7044555	sense
ptr_new_10	ptr	scaffold_125	778448	778467	antisense
ptr_new_12	ptr	LG_XIII	9782577	9782596	sense

Table 2. Locations of miR319 genes as found by MiRPal.
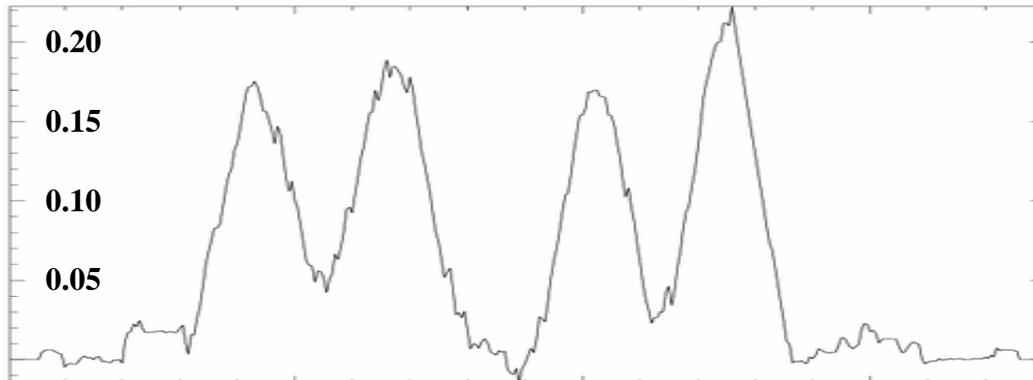
Figure 4. Plotcon similarity plot of aligned miR319 hairpin sequences.

    X-axis = nucleotide position (averaged over a 21 nucleotide window).

    Y-axis = similarity.
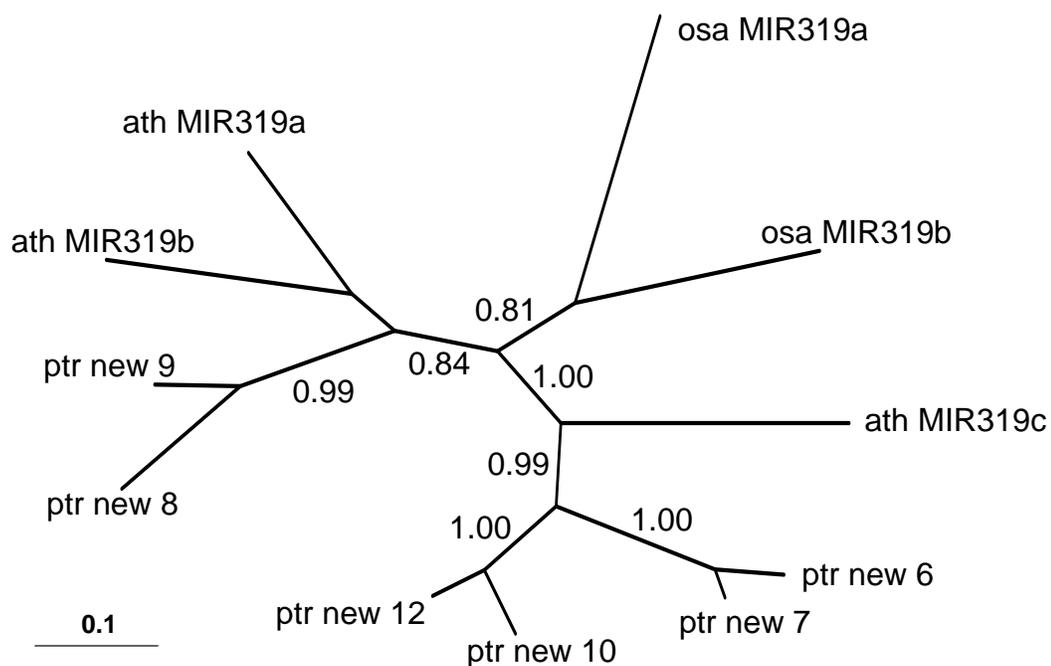
Figure 5. Phylogenetic tree of the miR319 hairpin sequences.

Values at the nodes were obtained by bootstrap analysis based on 105 resamplings. Bootstrap values over 80% are shown. The scale bar represents 10% estimated sequence divergence.

ptr_new_6

ptr_new_7

ptr_new_8

ptr_new_9

ptr_new_10

ptr_new_12
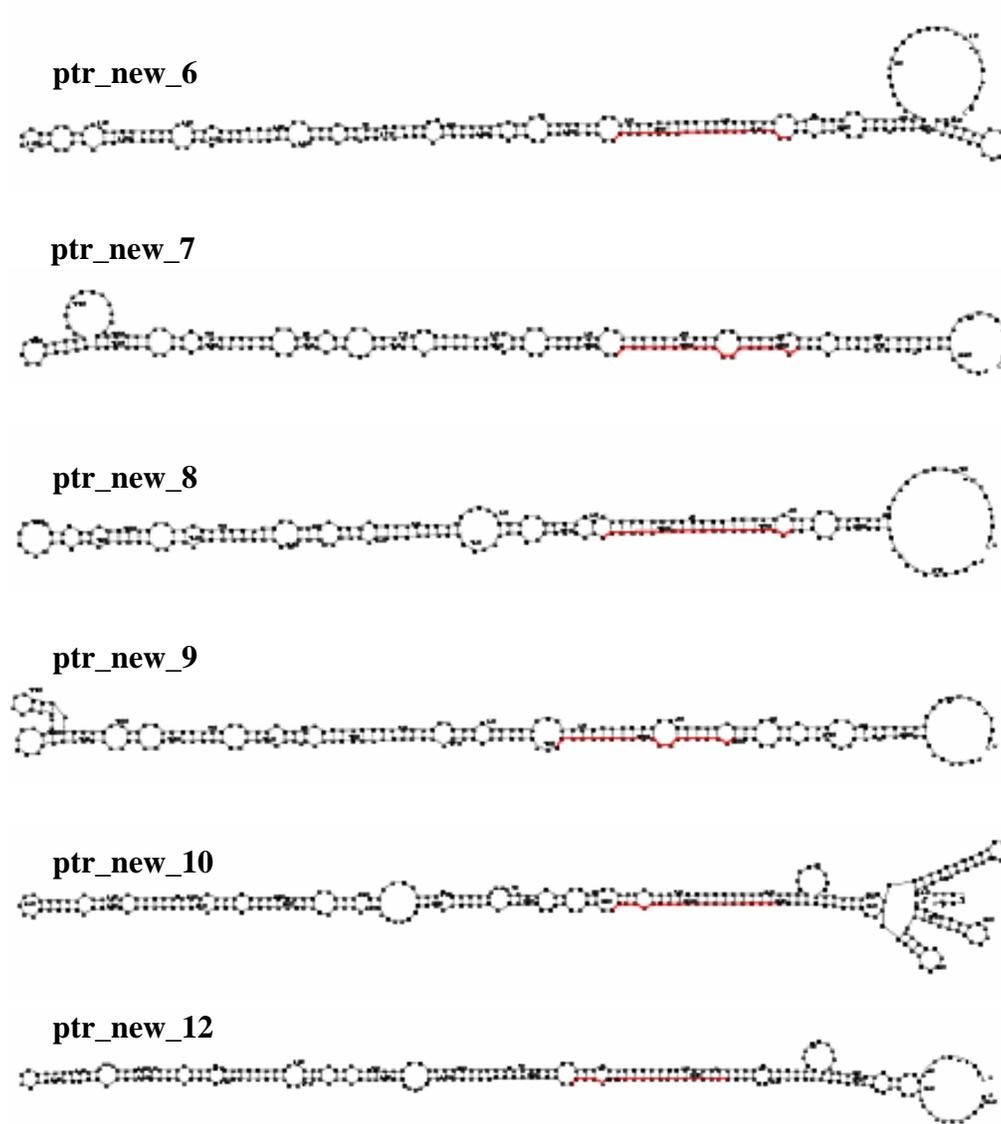
Figure 6. Secondary structures for the unannotated miR319 genes in *P. trichocarpa* as found by MiRPal.

Structures were generated with the Vienna RNA software suite.

DISCUSSION

In this experiment with miR319, MiRPal found the known miRNA gene loci in *A. thaliana* and *O. sativa* with zero false positives and zero false negatives. However, in this case all of the loci for the 20-nucleotide long mature miRNA sequences were in fact miRNA gene loci, thus making it impossible to produce a false positive. To ensure that the zero false negative rate was not purely an artifact of the override for RFam microRNA Registry-annotated loci, the author re-ran the analysis after removing the code for the override and found the results to be unchanged.

As evident by the similarity plot in Figure 4 and by the fact that all but one of the clades in the phylogenetic tree (Figure 5) were supported by bootstrap values of 81% and above, MiRPal was successful in creating an alignment (Appendix I) of highly related sequences containing the known mature sequences for miR319.

Most notably, the program found six unannotated loci for mir319 genes in *P. trichocarpa*, a recently assembled genome. From a visual inspection of the secondary structures of these loci (Figure 6), they do indeed appear to be consistent in their structure to plant miRNAs. Their alignment with known miRNAs from *A. thaliana* and *O. sativa* provide further evidence that they are likely orthologs for those genes. However, chemical assays will be necessary to verify these suppositions.

CONCLUSIONS

MiRPal has shown that it is capable of locating previously known microRNA gene loci; therefore it has the potential to locate unannotated loci as well, assuming that the characteristics for microRNA genes are constant.

The speed of MiRPal – 8min, 30sec for miR319 in this study – makes this program an excellent exploratory tool for small RNA research, when the goal is to quickly analyze many different RNA sequences.

REFERENCES

1. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389-402.

2. Edgar, Robert C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* **32**, 1792-97.

3. Griffiths-Jones S (2004). The microRNA Registry. *Nucleic Acids Res* **32**, D109-11.

4. Huang X, Madan A (1999). CAP3: A DNA sequence assembly program. *Genome Res* **9**, 868-77.

5. Hofacker I, Fontana W, Stadler P, Bonhoeffer L, Tacker M, Schuster P (1994). Fast Folding and Comparison of RNA Secondary Structures. *Monatsh Chem* **125**, 167-188.

6. Rice P, Longden I, Bleasby A (2000). EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet* **16**, 276-7.

7. Ronquist F, Huelsenbeck JP (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* **19**, 1572-4.

APPENDIX

APPENDIX I – Alignment for miR319 gene sequences.

Color Legend: BLUE ≥ 50% Similarity;   RED ≥ 75% Similarity

```
ptr_new_8    GATGATAAT-T-------------------------------------A
ptr_new_7    ----------------------------CATGAGA-------------A
ath_MIR319a  -------------------------------------------TAGGAAT
ptr_new_9    --------T-CTAGATGATAATT-------------------------A
osa_MIR319b  -------GT-TAACATATTCTTTTAATTTGATGGAAGAAGCGATC----G
ptr_new_10   TACTGTAGTGGTCCGTATTACTTAGTTGTGAGAGAATT----------A
ath_MIR319c  AAA----GT-ATATATATGAATCAAGACCAAAATA-------------G
osa_MIR319a  ------TGT-TTTGCAATTCATTTGGTGCTCCAAGATTGTGGCTTTGACT
ptr_new_6    --------------AAATTTCTTAATTTTCATGAAA------------A
ptr_new_12   ATTGGTTATGAGAGAGAATTAATT-----------------------A
ath_MIR319b  TGTGCTTGT-ATGTATGCGTATATATAGTAACGGAATTGTTAATATGCTT


ptr_new_8    ATAGATA----------------AGAGAGCTTTTTTCGGTCCACTTATAG
ptr_new_7    ATGGTTTA---------------AGAGAGCTTTCTTCAGTCCACTCATGG
ath_MIR319a  ATATATGT-----------AG--AGAGAGCTTCCTTGAGTCCATTCACAG
ptr_new_9    ATAGCTA----------------AGAGAGCTTTCTTCAGTCCACTCATGG
osa_MIR319b  ATGGATGG-----------A---AGAGAGCGTCCTTCAGTCCACTCATGG
ptr_new_10   ATCGTGGG---------------AGAGAGCTTCTTTCAGCCCACTCGTGG
ath_MIR319c  ATAGATAT-----------AG--AAGGAGATTCTTTCAGTCCAGTCATGG
osa_MIR319a  ACATGTGT-----------AAGAAGAGAGCTCTCTTCAGTCCACTCTCAG
ptr_new_6    ATGGTTTA---------------ACAGAGCTTCCTTCAGTCCACTCATGG
ptr_new_12   ATGGTGGG---------------AGAGAGCTTCCTTCAGCCCACTCATGG
ath_MIR319b  ATAGATGTTGAGTTGGTGGAGGAAGAGAGCTTTCTTCGGTCCACTCATGG
             *                       *  ***      **  * *** *     *


ptr_new_8    ATAGTAATA-----TGATTTAATTAGTTACCGACTCATTCATCCAAATAC
ptr_new_7    ACGGGCGAAG----GGTTTGGATTAGCTGCCGACTCATTCATTCAAACAC
ath_MIR319a  GTCGTGATA-----TGATTCAATTAGCTTCCGACTCATTCATCCAAATAC
ptr_new_9    GTGGTAGTA-----GGATTTAATTAGCTGCCGACTCATTCATCCAAATAC
osa_MIR319b  GCGGTGCTA-----GGGTCGAATTAGCTGCCGACTCATTCACCCACATGC
ptr_new_10   ATAGGACAAAG---GGGTTGAACTAGCTGCCGACTCATTCATTCAAGCAC
ath_MIR319c  ATAGAAAAAGAAGAGGGTAGAAATATCTGCCGACTCATCCATCCAAACAC
osa_MIR319a  ATGGCTGTA-----GGGTTTTATTAGCTGCCGAATCATCCATTCACCTAC
ptr_new_6    ACGGGCGAAG----GGTTTGGATTAGCTGCCGACTCATTCATTCAAACAC
ptr_new_12   ATAGGAGAAAG---GGGTTGAATTAGCTGCCGACTCATTCATTCAAGCAC
ath_MIR319b  A--GTAATATGTG-AGATTTAATTGACTCTCGACTCATTCATCCAAATAC
             *        *      * *   * *      *   *** **** **   **      *


ptr_new_8    ----------TGAGTT--ATGAGCTAAGATTA--CCCAGTAAATGAATGA
ptr_new_7    AGTA-GAAATTAAG--GGAGCAGTATGGCT--GCTATTGTGAATGTGTGA
ath_MIR319a  ---C-------GAGTCGCCAAAATTCAAACTAGACTCGTTAAATGAATGA
ptr_new_9    ---T-GAGTTAAAGGACAAGGAGATTA-------CCCAGTAAATGAGTGA
osa_MIR319b  ---C-AAGCAAGAAACGCTTGAGAT--AGCGAAGCTTAGCAGATGAGTGA
ptr_new_10   TAGTAGAAAAAAAGGCGAACGGTGTTTCTTTTGCTACTGTGATTGTGTGA
ath_MIR319c  TCGT-GGTAGAGAAACG-ATAAATTTAAAC----CGCAGTGACTGTGTGA
osa_MIR319a  ---C---AAGAAAGTTGCAGGAGTGTATCT----CTTGGTAGCGGACTGG
ptr_new_6    AGTA-GACAAGGAGTGGCAGCGGCT-------GCTATTGTGAATGTGTGA
ptr_new_12   CAGT--AGAAAAAGGGGAATGGATATTCTTTTGCTACTGTGATTGTGTGA
ath_MIR319b  ---C--------AAATGAAAGAATTTGTTCTCA-TATGGTAAATGAATGA
             *                         *             * **
```

```
ptr_new_8    GTGATGCGGGAGACAA--ATTAAATCTTAAACTTCCTATCATTGGACTGA
ptr_new_7    ATGATGCGGGAGATAA-ATTTCATCCTTTTCTTCTCTGTGCTTGGACTGA
ath_MIR319a  ATGATGCGGTAGACAA--ATTGGATCATTGATTCTCTTTGATTGGACTGA
ptr_new_9    ATGATGCGGGAGACAA--ATTGAATCCTAAGCTTCCTGTACTTGGACTGA
osa_MIR319b  ATGAAGCGGGAGGTAACGTTCCGATCTCGCGCCGTCTTTGCTTGGACTGA
ptr_new_10   ATGATGCGGGAGATAATTTTCCATCCTCTCCTTTTCTGTGCTTGGACTGA
ath_MIR319c  ATGATGCGGGAGATAT--TTTTGATCCTTCTTTATCTGTGTTTGGACTGA
osa_MIR319a  ATGACGCGGGAGCTAA-AATTTAGCTCTGCGCCGTTTGTGGTTGGACTGA
ptr_new_6    ATGACGCGGGAGATTA-ATTTCATCCTTTTCTTCTCTGTGCTTGGACTGA
ptr_new_12   ATGATGCGGGAGATAATTTTACATCCCCTCTTTTTCTGTGCTTGGACTGA
ath_MIR319b  ATGATGCGAGAGACAA--ATTGAGTCTTCACTTCTCTATGCTTGGACTGA
             *** ***  **        *              * *   *********


ptr_new_8    AGGGAGCTCCCTTTACT------------------------------GT
ptr_new_7    AGGGAGCTCCCTTT---------------------------------AA
ath_MIR319a  AGGGAGCTCCCTCTCTC--------------------TTTTGT----AT
ptr_new_9    AGGGAGCTCCCTTTCCT-------------------------------TT
osa_MIR319b  AGGGTGCTCCCTCCTCC---------------TCGATCTCTTCG----AT
ptr_new_10   AGGGAGCTCCTTCCTTC--------------------TCTTGCTTTCAT
ath_MIR319c  AGGGAGCTCCTTCTTTT------TCTACAAATTTTCATTTTTCTCATAAT
osa_MIR319a  AGGGTGCTCCCTTGCTC---------------AAGCACTCGTGC----AT
ptr_new_6    AGGGAGCTCCCTTT---------------------------------AA
ptr_new_12   AGGGAGCTCCTTCCTTC--------------------TATTGCTTTCAT
ath_MIR319b  AGGGAGCTCCCTATTTTTATCTTTCTCAGAGTACCACATTTTAT----AT
             **** ***** *


ptr_new_8    TATACTATATTTT-------------------------------------
ptr_new_7    TCGTTTATTCACGAT-----------------------------------
ath_MIR319a  TC------------------------------------------------
ptr_new_9    TATACTATGTATC------------------------------------T
osa_MIR319b  CTAATTA-------------AGCT-------------------ACCTT
ptr_new_10   TCATTGA------------------------------AGAAACAGTCT
ath_MIR319c  CTGTTTA-------------------------------------------
osa_MIR319a  AAAATCATGCCCTAATGTTAA----------------------------T
ptr_new_6    TTGTTTATTCACGATCTTAGA----------------------------T
ptr_new_12   TTACTGA-------------------------------------------
ath_MIR319b  CTATATATATACAAGCATTGAATTCGTGTGTATAACTCAAGATCAAGTTC


ptr_new_8    -----------
ptr_new_7    -----------
ath_MIR319a  --CAAT-----
ptr_new_9    TATACT-----
osa_MIR319b  GACAT------
ptr_new_10   AATACTGTTTG
ath_MIR319c  -----------
osa_MIR319a  AATAAT-----
ptr_new_6    GACAAGATCC-
ptr_new_12   AGAAAGAATTT
ath_MIR319b  AACATT-----
```