

A High-Throughput Approach for Associating MicroRNAs with their Activity Conditions

Chaya Ben-Zaken Zilberstein ^{1,*} Michal Ziv-Ukelson ^{1,*,[!]} Ron Y. Pinter ¹ Zohar Yakhini ^{1,2}

* These authors contributed equally to the paper.

¹ Dept. of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel. email: {chaya, michalz, pinter}@cs.technion.ac.il

² Agilent Technologies, Tel Aviv, Israel. email: zohar_yakhini@agilent.com

[!]Corresponding author. email: michalz@cs.technion.ac.il, phone: 972-4829-4883, fax: 972-3636-7566.

Abstract

Biological background: Plant microRNAs (miRNAs) are short RNA sequences that bind to target mRNAs and change their expression levels by redirecting their stabilities and marking them for cleavage. In *Arabidopsis thaliana*, microRNAs have been shown to regulate development and are believed to impact expression both under various conditions, such as stress and stimuli, as well as in specific tissue types.

Methods: We present a high throughput approach for associating between microRNAs and conditions in which they act, using novel statistical and algorithmic techniques. Our new tool, *miRNAXpress*, at first computes a (binary) matrix T denoting the potential targets of microRNAs. Then, using T and an additional predefined matrix X indicating expression of genes under various conditions, it produces a new matrix that predicts associations between microRNAs and the conditions in which they act.

Thus, the program comprises two main modules that work in tandem to compute the desired output. The first is an efficient target prediction engine that predicts mRNA targets of query microRNAs by evaluating the optimal duplex that could be formed between the two: given a short query RNA, a long target RNA, and a predefined energy cut-off threshold, the program finds and reports all putative binding sites of the query RNA in the target RNA with hybridization energy bounded by the predefined threshold. The second module realizes an *association* operation that is computed by a method which relies on an efficient *t*-test to compute the associations.

The calculation of the matrix of microRNAs and their potential targets is the computationally intensive part of the work done by *miRNAXpress* and therefore an efficient algorithm for this portion facilitates the entire process. Thus, the target prediction engine is based on an efficient approximate hybridization search algorithm whose efficiency is the result of utilizing the sparsity of the search space without sacrificing the optimality of the results. The time complexity of this algorithm is almost linear in the size of a sparse set of locations where base-pairs are stacked at a height of three or more.

Results: *miRNAXpress* is a novel tool for associating between microRNAs and the conditions in which they act. We employed it to conduct a study, using the plant *Arabidopsis thaliana* as our model organism. By applying *miRNAXpress* to 98 microRNAs and 380 conditions, some biologically interesting and statistically strong relations were discovered. For example, *mir159C* activity is possibly a factor in the misresponse of *nph4* mutants to phototropic stimulations.

1. Introduction

Genes in plants may be expressed in specific locations (*e.g.* leaf-specific genes), at specific times (*e.g.* seedling-specific genes), or in response to environmental stimuli (*e.g.* light-responsive genes). The cellular expression levels of genes are largely influenced by transcription rates as well as by the degradation rates of their mRNAs (messenger RNA). Plant microRNAs (miRNAs) are non coding RNA molecules (of size ≈ 22 nucleotides) that regulate gene expression in plants by moderating mRNA degradation rates. Plant microRNAs bind to mRNAs and mark them for cleavage [10, 12, 23]. For example, *Arabidopsis thaliana* microRNA 39 interacts with mRNAs of several transcription factors to reduce their expressions and to direct the plant development process.

In this paper we propose a high throughput approach for associating between plant microRNAs and conditions in which they act, given a set of microRNAs and a set of conditions. A condition can be, for example, a certain tissue in which the gene expression is to be measured or exposure of the plant to a long darkness period. Note that there are other factors which influence gene expression, including transcription factors which regulate mRNA transcription rates and proteins which mediate mRNA degradations, but they are beyond the scope of this study.

Our framework for associating a microRNA and a specific condition can be formalized as follows (see the example in Figure 1). First note that since there is a correspondence between genes and mRNAs, we use the term “mRNA” to represent the corresponding gene. Suppose that mRNAs $mRNA_1$, $mRNA_4$ and $mRNA_6$ are targets of microRNA Y , yet other mRNAs $mRNA_2$, $mRNA_3$ and $mRNA_5$ clearly do not bind to Y . Furthermore, assuming there is evidence that $mRNA_2$, $mRNA_3$ and $mRNA_5$ are highly expressed under some given condition, yet $mRNA_1$, $mRNA_4$ and $mRNA_6$ are expressed at a low level under the very same condition, then it may be possible to statistically assert that microRNA Y is active under this condition, contributing to the degradation and low expressions of $mRNA_1$, $mRNA_4$ and $mRNA_6$. This framework is extended to its high-throughput formalism, using the following three matrix definitions, which are exemplified in Figure 2.

Given a set of p candidate microRNAs $\alpha_1 \dots \alpha_p$, another set of q conditions $\beta_1 \dots \beta_q$, and a third set of r genes $\gamma_1 \dots \gamma_r$, the association between microRNAs and their predicted activity conditions can be formally captured in an *Association Matrix*.

Definition 1. *The Association Matrix* A is a $q \times p$ matrix of real numbers such that $A[i, j]$ reflects the

association of microRNA α_j and condition β_i .

Table 1 contains highlight entries from the association matrix computed using our method by the experimental setup described in Section 5.

The matrix A is computed by combining information (operating on the rows and columns) from two pre-computed matrices: the Expressions Matrix X and the Targets Matrix T , defined as follows.

Definition 2. *The Expressions Matrix X is a matrix of real numbers such that $X[i, j]$, for $i = 1 \dots q$ and $j = 1 \dots r$, represents the expression of gene γ_j under condition β_i .*

(This is a standard representation of the results of expression profiling studies).

Definition 3. *The Targets Matrix T is a binary matrix where $T[i, j] = 1$ means that gene γ_i is a predicted target of microRNA α_j .*

In our approach $A[i, j]$ is computed by performing an operation between row X_i of the *Expressions Matrix* and column T_j of the *Targets Matrix*, that is $A[i, j] = F(X_i, T_j)$. We focus on the following instance of an association operation F :

Definition 4. *The Association Operation F : Let \bar{z} denote the binary complement of z . $F(X_i, T_j)$ returns a p -value which is the result of a statistical test comparing two sets of numbers, $\{X[i, \ell] \cdot T[\ell, j]\}_{\ell=1}^r$ and $\{X[i, \ell] \cdot \overline{T[\ell, j]}\}_{\ell=1}^r$ (where “ \cdot ” denotes binary selection) and assessing whether they are significantly different (whether one set contains significantly lower values than the other set).*

The operation F reflects the intuition explained above: it measures the differential expression between potential targets ($\{X[i, \ell] \cdot T[\ell, j]\}_{\ell=1}^r$) of a microRNA and non targets ($\{X[i, \ell] \cdot \overline{T[\ell, j]}\}_{\ell=1}^r$). Thus, $A[i, j]$ is set to the p -value which quantifies the association between a microRNA and a condition. This p -value is the result of a statistical test intended to reject the hypothesis that the atypical low expression of the targets is due to factors other than quick degradation induced by the microRNA activity. Clearly, the lower the p -value, the higher the assumed likelihood that the expression of the targets is indeed influenced by the microRNA activity (see Table 1).

The *Expressions Matrix* X contains expression values measured experimentally in a microarray assay. A wealth of such information has already been collected by various laboratories and is now available to the public. Naturally, the construction of this matrix does not involve any significant computational task.

However, such is not the case with the *Targets Matrix* T . In order to calculate $T[i, j]$ one needs to assess whether mRNA i is a target of microRNA j in vivo. This can be done computationally, as will be discussed in the next section.

1.1 Computing the Target Matrix T : Approach and Background

Here, we aim to assess whether two sequences, the first being a short microRNA and the second a full length mRNA, bind to each other in vivo. The most intuitive measure for this binding is the free energy of the most stable duplex which could potentially form between the two. This duplex may contain mismatches, bulges and interior loops, since although the complementarity between plant microRNAs and their targets is usually quite high, it is not perfect. For example, the complementarity between the sequence of the mRNA DCL1 and the sequence of microRNA miR162 is not perfect, and it contains a bulge nucleotide. (DCL1 is subject to negative feedback regulation of miR162 which marks it for cleavage [26].) Another example is microRNA miR-JAW with 4-5 mismatches to the mRNAs of several transcription factors [16].

Moreover, a recent study by Yekta *et al.* [27] has shown that plants are not unique in using microRNAs to regulate mRNA target cleavage. This study demonstrated that natural metazoan microRNAs direct mRNA cleavage as well. Since, in animals, microRNAs often display limited complementarity to their targets, designing our algorithms to allow mismatches, bulges and loops of various size is important in order to keep them general enough to support animal studies as well. Thus, the core of our approach is an efficient algorithm for computing the optimal free energy of a duplex between a microRNA and an mRNA target, using scoring rules derived from the nearest neighbor thermodynamic approach [28, 13]. Performing this calculation for the entire set of microRNAs and genes of interest yields the targets matrix T . Thus, even though our study here focuses on plant microRNAs, the tool we developed is general enough to apply to animal microRNAs as well.

Work to date on microRNA target prediction consists of methods that either rely solely on edit distance [19], or combine sequence similarity information with secondary structure prediction by energy minimization. The combination is achieved via a two-stage approach [4, 17, 22]: During the first stage potential binding sites are identified by searching for near-perfect base complementarity to the 5'-end of the microRNAs (or some heuristically set "nucleus" of base pairs). For each such match a candidate consisting of a stretch of about twice the microRNA size from the target is extracted, anchored at the highly matched nucleus. In the second stage, the secondary structure is computed, for the candidates suggested by the first

stage, by applying the standard folding program *Mfold* by Zuker *et al.* [28] to the concatenation of potential binding site and microRNA. If this results in a score below a predefined threshold then a microRNA/mRNA target relation is determined.

Note that there are clear drawbacks to this approach. The first drawback is that the sequences have to be concatenated with a short linker sequence that can lead to artifacts in the prediction, as explicitly demonstrated by [18]. The second problem is that mRNA or microRNA self-structure can occur. However, this self-binding should not be allowed in the specific subsequences which engage in the duplex, for the following reason: microRNAs function as guides for RISC (RNA-induced silencing complex) to cleave mRNA, and thus the whole microRNA/mRNA duplex is incorporated into the RISC complex [3]. Base pairing between target nucleotides or microRNA nucleotides would result in a structure that would prevent this incorporation due to steric interference. Moreover, the microRNA is so short that it is natural to assume that no two loops cross and that no multiple loops are formed. The third drawback is that for prediction of multiple bindings in one target, as in the case of animal microRNAs, the appropriate potential binding sites have to be cut out and folded separately. Furthermore, these algorithms rely on (heuristic) guiding parameters such as the size of the required “nucleus” of base pairs or the predefined limit on the number of target nucleotides to participate in the duplex. All these could result in false negatives.

In Rehmsmeier *et al.* 2004 [18] a program is described that directly predicts multiple potential binding sites of microRNAs in large target mRNAs, self-structures not allowed. In general, the program finds the energetically most favorable bindings of a small RNA to a large RNA using the nearest-neighbor thermodynamic scoring rules. However, it uses brute-force dynamic programming, whose time complexity would naively be $O(m^2n^2)$, where m is the size of the microRNA and n the size of the target (see figure 3). The program is sped up by heuristically restricting the size of the allowed gap to 15.

1.2 Our Results

miRNAXpress is a novel tool for associating between microRNAs and the conditions in which they act. The program is composed of two main modules that work in tandem to compute the desired output.

The first component is a microRNA/mRNA *target prediction engine* that, given a query microRNA, a text mRNA and a predefined energy cutoff threshold, finds and reports all targets (putative binding sites) of the query in the text with binding energy below the predefined threshold. The process does not allow self-structures. The target prediction engine is based on an efficient algorithm that exploits the sparsity of the

search space without sacrificing the optimality of the results (no heuristics are used). The time complexity of the algorithm is almost linear with the size s of a sparse set of stacked base pair locations. It is based on the approaches of Eppstein, Galil, Giancarlo, and Italiano [7] and Miller and Myers [14, 15], and is extended to form an algorithm which utilizes the score-bounding as well as the convexity of the loop-cost function to speed up the search. Further reduction of the complexity is shown for the target prediction decision problem with discrete scores. The algorithm is described in Section 2. We refer the reader to Section 4 and Figure 14 for a comparison between the *miRNAXpress* target prediction engine versus the naive dynamic programming approach in terms of practical run-time.

The second component takes a pre-defined *Expression Matrix* (based on a gene set and a condition set which are given as input to the program) and a *Targets Matrix*, which is computed by the *target prediction engine* component (given the gene set and a set of input microRNAs), and applies the *Associating Operation F* to the two matrices (see Figure 2). The resulting *Association Matrix* is the output of the program. The formation function is described in Section 3.

Note that the prediction of the targets is the heavier part of the above process in terms of computational complexity. The efficient approach to this part facilitates the entire process. Also note that the algorithm described in this paper avoids the artifacts associated with the concatenation of the microRNAs to the targets, which was a drawback of previous target prediction algorithms which applied a two-stage approach.

We employed *miRNAXpress* to conduct a study, using the *A. thaliana* plant as our model organism, as follows. An *Expression matrix* was assembled from 5800 genes and 380 conditions. These conditions included various tissues, hormonal treatments, ionic and cationic stress, and pathogens. A *Targets Matrix* was then constructed by testing the *Approximate Hybridization Prediction (AHP) Existence* (See Definition 6.2) of each one of 98 previously discovered *A. thaliana* microRNAs in each of the 5800 mRNAs. Some of the associations established as a result of our study are discussed in Section 5. These are plant microRNA α_i such that their association with specific condition β_j got a significant *p-value*.

As an additional result, this study led to the discovery of some tissue specific microRNAs. These microRNAs yielded a significant *p-value* only in specific tissues of the plant (see Figure 15).

2. The Target Prediction Engine

The problem of target prediction is similar in essence to that of finding RNA secondary structure (see e.g.[20]). However, in contrast to the traditional problem of RNA secondary structure prediction, we seek

optimal alignments between a short microRNA sequence (of size < 30) with a long mRNA sequence (of size varying from ~ 1000 to ~ 3000). Following the discussion of Section 1.1, we assume that no two loops cross and that no multiple loops are allowed (as explained in Section 1.1).

Therefore, let T denote a text (mRNA) of size n and P denote a pattern (microRNA) of size m . In the spirit of the extensive studies on predicting RNA secondary structure, pioneered by Zuker *et al.* [13, 28], we use nearest-neighbor thermodynamics to estimate RNA duplex energy. The energy of a duplex will therefore be computed as the sum of all its component energies: base-pairs which create stability, versus mismatches, interior-loops and bulges, which reduce it (see Figure 4). Let $A = a_1 \dots a_n$ and $B = b_1 \dots b_m$ be two RNA sequences. Let $es(i-1, i, j-1, j)$ denote a predefined energy score (usually negative) for the base-pair (i, j) (such that the stacked base-pair (i, j) follows the base-pair $(i-1, j-1)$). Let $w(i', j', i, j)$ denote the energy score (usually positive) of an interior loop with one side of length $i - i'$ and the other of length $j - j'$, closed by pair a_i, b_j at one end and by $a_{i'}, b_{j'}$ at the other. Similarly, let $w'(i', i)$ denote a predefined energy score (usually positive) for a bulge of length $i' - i$, closed by pair a_i, b_j at one end and by $a_{i'}, b_{j'}$ at the other. Let \mathcal{D} denote the set of base pairs in the duplex and let \mathcal{W} denote the set of index quartets defining interior loops in the duplex. Let \mathcal{W}'_P denote the set of start and end index-pairs of bulges in P and let \mathcal{W}'_T denote the set of start and end index-pairs of bulges in T .

Definition 5. Given two hybridized RNA sequences, R_1 and R_2 , which form a duplex, the **Duplex Hybridization Energy Score** of R_1 and R_2 , denoted $DHES[R_1, R_2]$, is

$$DHES[R_1, R_2] = \sum_{\forall (i,j) \in \mathcal{D}} es(i-1, i, j-1, j) + \sum_{\forall (i',j',i,j) \in \mathcal{W}} w(i', j', i, j) + \sum_{\forall (i,i') \in \mathcal{W}'_P} w'(i', i) + \sum_{\forall (i,i') \in \mathcal{W}'_T} w'(i', i)$$

In this section we address a new and challenging problem, based on RNA secondary structure prediction: finding all approximate helix bindings of a short RNA sequence (microRNA) in a long one (mRNA), given a threshold bound on the allowed score. This challenge is formally defined as the following search and decision problems.

Definition 6. Approximate Hybridization Prediction (AHP):

1. *The Search Problem:*

The **AHP Search** problem is, given a predefined score threshold e , to find all the approximate predicted hybridization sites of P in T , where the $DHES$ score of the duplex formed by P and the corresponding

putative binding site in T is at most e .

2. The Decision (Existence) Problem:

The **AHP Existence** problem is, given a predefined score threshold e , to predict whether or not there exists a hybridization site of P in T where the DHES score of the duplex formed by P and the corresponding putative binding site in T is at most e .

Note that the *AHP* search problem is applicable to target discovery in animals, where microRNAs display complementarity to multiple sites in a single mRNA. In plants, however, microRNAs display complementarity to only a single site, and therefore applying the decision version of the problem is more appropriate in this case.

The thermodynamic parameters used by our target prediction engine are experimentally derived and are similar to those used by *Mfold Version 3* for RNA folding [28], scaled to accommodate the relevant plant and animal conditions. The loop energy cost w sums up three terms: a purely entropic term that depends on the loop size, the terminal stacking energies for the mismatched base pairs adjacent to both closing base pairs, and an asymmetric loop penalty for non-symmetric interior loops. In both w and w' (bulge energy cost) the loop destabilizing energy grows logarithmically with the total size of the loops, *i.e.* $i - i' + j - j' - 2$ for interior loops, $i - i'$ for bulges in P and $j - j' - 1$ for bulges in T .

Thus, the energy calculation can be modeled by the following dynamic programming equations, based on the formalization of Waterman and Smith [25]:

$$D[i, j] = \min\{D[i - 1, j - 1] + es(i - 1, i, j - 1, j), V[i, j], H[i, j], E[i, j]\} \quad (1)$$

where

$$V[i, j] = \min_{0 < i' < i} \{D[i', j - 1] + w'(i', i)\} \quad (2)$$

$$H[i, j] = \min_{0 < j' < j} \{D[i - 1, j'] + w'(j', j)\} \quad (3)$$

$$E[i, j] = \min_{0 < i' < i, 0 < j' < j} \{D[i', j'] + w(i', j', i, j)\} \quad (4)$$

Note that the heavier part of the computation of Equation 1 is the $E[i, j]$ term, as computed in Equation 4. For the sake of simplicity we will therefore assume throughout this section that Recurrences 2 and 3 are simplified instances of Equation 4 and focus our explanations on the handling of Equation 4. Also, from

now on we will use the term “arc” when referring to both bulges and loops.

A naive dynamic programming algorithm solves Recurrence 1 for sequences of length n in $O(n^4)$ time. A lower complexity of $O(n^3)$ can be achieved with no assumptions on loop destabilizing functions [25]. Eppstein *et al.* [5] considered loop destabilizing functions satisfying certain convexity or concavity conditions, and developed an $O(n^2 \log^2 n)$ algorithm for this case. This was later improved to $O(n^2 \log n)$ [1], and finally to $O(n^2 \alpha(n))$ (where α is the inverse of Ackerman’s function) for logarithmically growing destabilizing functions [11].

2.1 Sparsification

For the parameters and cost functions that are standard for base stacking and loops, one can show that the energy cost of breaking a loop will be more than the energy benefit from a single base pair or two stacked base pairs. Thus, we ignore base pairs that cannot be stacked without gaps at height 3 or more. This insight can be used to greatly reduce the number of possible pairs, yielding a sparsification of the dynamic programming matrix for P versus T . Instead of nm entries we drop to about $nm/64$ (the probability of an occurrence of a consecutive complementary triplet), without sacrificing the optimality of the results. (In fact in our data the number of relevant base pairs was analyzed to be about one percent on average.) Thus the computation and minimization needs to be taken only over positions (i, j) which end a triplet of complementary base-pairs, taking into account wobble pairs *i.e.* $G-U$ pairs as well. Let S be the set of such positions, to be denoted “points” in the rest of this paper (see Figure 5), and let $s = |S|$. Note that the set S can be computed in $O(s + n \log \Sigma)$ time, where Σ is the size of the alphabet (four nucleotides in this case), using standard string matching techniques (suffix trees). The effects of sparsity on the alignment of RNA has been studied, in a unifying framework, by Eppstein *et al.* [6, 7]. With Johnson’s data structure [7] and a special implementation of the binary search, an $O(n + s \log s \log \min(s, n^2/s))$ bound can be obtained. For simple destabilizing functions, it decreases to $O(n + s \log s \log \log \min(s, n^2/s))$. Larmore and Schieber [11] have improved one of the algorithms of Eppstein *et al.* [7] to $O(n + s \log \min(s, n^2/s))$ for concave w and $O(n + s \alpha \log \min(s, n^2/s))$ for convex w . However their algorithm uses matrix searching techniques, which lead to a high constant factor in the time bound. Therefore, we chose the algorithm of [7] as the basis for our target prediction engine, to be described in the next section.

2.2 The AHP Algorithm (for both the Search and Existence Problems)

Following Definition 6.2, the *AHP* algorithm searches for potential binding sites of P in T that may form

a duplex with P of score below a predefined *DHES* cutoff threshold e . Our cutoff threshold is a constant fraction (specifically 0.85) of the optimal energy possible for the microRNA/target duplex (this optimal energy is calculated by aligning the microRNA with its exact complement sequence).

Observation 1. *The score threshold e imposes a lower bound on the number of microRNA nucleotides that must hybridize in order to achieve the threshold score, and therefore an upper bound on the number of deletions allowed from the microRNA in the sought alignment.*

For example, assume that matches contribute -2 to the free energy and consider a microRNA of size 10 and a threshold of -18 (that is, -18 is an upper bound on the minimal energy allowed for an accepted duplex). In that case the number of allowed deletions is 1, because more deletions would result in less than 9 matches, yielding a hybridization energy that is higher than -18 . For each microRNA, knowing the content of its nucleotides, we can easily derive the minimal number of matches necessary to achieve the required threshold.

Therefore, let k denote a pre-defined bound, based on the *DHES* threshold e and on the maximum number of deletions from P allowed in aligning P and T . Note that k is immediately a bound on the maximal size of an allowed gap in P . Also note the asymmetric nature of the gap binding: the acceptance threshold score does not impose a limit on the number of deletions from the long sequence T .

2.2.1 A Division of the *DP* Table Based on the Gap Bound in P

In this section we demonstrate how the k bound on gaps in P , as imposed by the pre-defined score threshold, can be used to restrict the search space and speed up the search, under the sparse dynamic programming model, without sacrificing the optimality of the scores. The dynamic programming table is divided to m/k slices of k rows each (see Figure 6). To each block we apply a divide-and-conquer recursion on the points of S which are included in its rows. The invariant is that at each level of the recursion the gaps in P are confined to a size that is twice the size allowed in the previous level. For each level of the recursion, having t rows in the subproblem of that level, we partition the rows into two blocks, the first consisting of the first $t/2$ rows of the block and the second consisting of the bottom $t/2$ rows of the block.

Definition 7. *Given two points $(i', j'), (i, j) \in S$ such that $i > i'$ and $j > j'$, the **arc-contribution** of point (i', j') to point (i, j) is the contribution of the score term suggested by an optimal alignment that consists of an arc from (i', j') to (i, j) to the score minimum computation of $E[i, j]$ in Equation 4.*

The score minima for the points of each sub-block are computed as follows:

1. Recursively solve the problem for the top sub-block.
2. Compute the arc-contribution of the points in the top sub-block to the points in the bottom sub-block.
3. Recursively solve the problem on the bottom sub-block.

Note that since the gap size in P does not exceed k , all arcs are confined to a maximum of two consecutive k -blocks (i.e. blocks of k rows each). Therefore, the k -blocks are processed in the following order. First the divide and conquer computation is applied to the first block. Then the arc-contribution of the first block to the second block is computed; only then is the divide and conquer computation applied to the second block. Now the arc-contribution of the second block to the third block can be computed, followed by a computation of the third block, and so on: for each pair of consecutive blocks the bottom block in the pair is only computed after the computation of the top one has been completed and the arc-contribution of the top block to the bottom block has already been resolved.

Lemma 1. *Imposing the block division as above will not sacrifice the optimality of the scores.*

Proof. *No false negatives:* All correct arcs of sizes up to $2k$ will be considered by the algorithm, and this set includes of course all potential arcs of size up to k . *No false positives:* The block division allows the consideration of arcs of sizes up to $2k$ (two consecutive blocks). However, since the k bound is derived from the score cutoff threshold e , we know that all hybridizations with gaps in P that are greater than k will by definition be ruled out by the search algorithm. □

2.2.2 Computing the Arc-Contribution of One Set of Points to Another

In this section we describe a point-traversal order which allows us to efficiently compute the arc contribution of a top sub-block to a consecutive bottom sub-block, following the partitioning of the dynamic programming table as described in Section 2.2.1. This point-traversal order will support dynamic minimization and is based on the approach described in [7]. We say that point (i', j') *precedes* point (i, j) , denoted by $(i', j') \prec (i, j)$, if and only if $i' < i$ and $j' < j$. The *diagonal index* of a point (i, j) is defined as $i + j$. Let d_k be the set of points (i', j') in S whose diagonal index is k .

Let S_1 be the set of points on the top sub-block, and S_2 be the set of points on the bottom sub-block in a given arc-contribution computation. Points in S_1 and S_2 are processed in order of their column indices first and only then by their row indices. Within a given column we first process the points of S_2 , and then the points of S_1 (see Figures 8 and 10).

Claim 1 *If the points of $S_1 \cup S_2$ are scanned in the order described above, then:*

1. *When a point $p \in S_2$ is reached, all points $p' \in S_1$ such that $p' \prec p$, and only those points, have already been traversed and analyzed.*
2. *The diagonal index of any point $p \in S_2$ that has not yet been scanned is greater than the diagonal index of any of the points already scanned in S_1 .*

Proof.

1. Clearly $i > i'$ since S_2 is the lower part of the block, and we also impose $j > j'$.
2. Consider any pair of points $\{(i', j') \in S_1, (i, j) \in S_2\}$ such that point (i', j') has already been scanned and processed in S_1 and point (i, j) has not yet been scanned in S_2 . Clearly $i > i'$ due to the block separation. Also $j' > j$ due to the column-scanning order. Therefore, the indices of the diagonals corresponding to the two points fulfill $i + j > i' + j'$. □

Note that the recurrence of Equation 4 has the property that the function w depends only on the differences between the two diagonals defined by its four argument points. Thus, during the computation of $E[i, j]$, the source points (i', j') can be grouped into sets of points with identical diagonal numbers, such that each set will contribute only one term to the minimization formulated in Equation 4 (see figure 7). For the set of source points (i', j') such that $i' + j' = x$ on a given diagonal x , we have only to consider the minimum among the $D[i', j']$ entries in d_x . Furthermore, by Claim 1, Equation 4 could be reformulated in terms of diagonal indices as follows. Let $x = i' + j'$, $y = i + j$, and let x_max denote a dynamic variable which, at any moment of the point traversal, stores a value which is greater by one than that of the largest diagonal in S_1 which was scanned so far. This yields

$$E'[y] = \min_{x < x_max} \{D'[x] + w(x, y)\} \text{ for } x_max \leq y \leq n + t, \quad (5)$$

Using the point-traversal order described above, the arc-contribution of points in S_1 to points in S_2 can be computed as a minimization problem with dynamically changing input values, based on Equation 5, as follows.

1. A point (i', j') in S_1 is processed by performing the operation of decreasing $D'[x]$ to $\min(D'[x], D[i', j'])$.
2. A point (i, j) in S_2 is processed by performing the operation of computing $E'[y]$.

2.2.3 Using the Convexity and Simplicity of w to Compute $E'[y]$ Efficiently

Note that, for the application at hand, it is customary to use the \ln function for quantifying the growth of w and w' when increasing the loop/bulge size. Figure 11 demonstrates three functions: $D'[3] + w(3, y)$, $D'[5] + w(5, y)$ and $D'[7] + w(7, y)$, where w is the \ln function, $D'[3] = 1$, $D'[5] = 0.25$ and $D'[7] = 0.5$. Also note that the \ln function is convex.

Definition 8. A weight function w is convex w.r.t. the y minima computation if $\forall x < x' < y < y'$, $w(x, y') - w(x, y) \leq w(x', y') - w(x', y)$.

The computation of $E'[y]$ is viewed as a competition among a sparse set of $|S_1|$ candidate diagonals from the range $0, 1, \dots, x_{max}$ for the minimum in Equation 5, to be computed for a sparse set of target diagonals in the range $x_{max} + 1 \dots n + t$.

Suppose that we wish to resolve which of the two graphs, candidate source diagonal 5 and candidate source diagonal 7 in Figure 11, will yield the minimum value for each of the target diagonals $y = 8, \dots, 16$. In the next Lemma we will show that the competing source diagonals (e.g. $x = 5$ and $x' = 7$) divide the target diagonals into intervals, to be denoted *leadership intervals*. In each such interval only one of the source diagonals from S_1 will yield the minima values for Equation 5. (For example, after the first three points of S_1 have been scanned, the leadership interval of source diagonal $x = 7$ in the example of Figures 10 and 11 yields the minima for target diagonals $8, \dots, 13$).

Lemma 2. Given a function w which is convex with respect to the minima of Equation 5.

1. For any x, y and x' , with $x < x'$, if $D[x] + w(x, y) \leq D[x'] + w(x', y)$, then for all $y' > y$, $D[x] + w(x, y') \leq D[x'] + w(x', y')$.
2. Conversely, if $D[x] + w(x, y) > D[x'] + w(x', y)$, then for all $x_{max} < y' < y$, $D[x] + w(x, y') > D[x'] + w(x', y')$.

Proof.

1. By the Definition of convexity, $w(x, y') + w(x', y) \leq w(x, y) + w(x', y')$. Subtracting $w(x, y') + w(x', y) + D[x'] - D[x]$ from both sides and rearranging yields $(D[x] + w(x, y)) - (D[x'] + w(x, y')) \leq ((D[x] + w(x', y)) - (D[x'] + w(x', y')))$. But by the assumption $((D[x] + w(x, y)) - (D[x'] + w(x, y')))$ is positive, and therefore $(D[x] + w(x', y)) - (D[x'] + w(x', y'))$ must also be positive and the first statement holds.

2. Similar to the former statement. The only difference is that here we encounter an interval $x \leq y < x'$ where $w(x', y)$ is not defined (see Figures 12 and 13) and therefore x rather than x' would yield the diagonal minima in this interval. However, by Claim 1.2, at the stage when $E(i, j)$ is computed this interval is no longer relevant to the minimization computation. \square

We point out that a similar lemma was stated and proven for the *concave* case in [7]. The *convex* case, however, is more complicated since $w(x', y)$ is undefined for any $x \leq y < x'$. Therefore, x rather than x' yields the diagonal minima in this interval. This imposes a fragmentation of the leadership-intervals (see figure 12 and Conclusion 1 below) which could strongly affect the efficiency of the algorithm. However, we resolve this problem for applications which follow the divide and conquer approach and traversal order described in Section 2.2.1, as follows: By Lemma 2.2 at the stage when $E(i, j)$ is computed, the fragmented leadership-intervals, all of which by definition fall to the left of x_max , are no longer relevant to the minimization computation and are therefore discarded (see figure 13).

Conclusion 1. *At any given point in the traversal of $S_1 \cup S_2$, the values of $D[x]$ supplying the minima for the positions of $E[y]$ partition the possible indices of y , $x_max < y < n + t$, into a sequence of “leadership intervals”. If $x' < x$ and if y is in the interval in which $D[x] + w(x, y)$ is best, and y' is in the interval in which $D[x'] + w(x', y)$ is best, then $y' > y$. Also, if both x' and x have active leadership-intervals, then $D[x] > D[x']$.*

Based on Conclusion 1, the algorithm maintains in a data structure a subset of candidates which satisfies the property that $E'[y]$ depends only on these candidates. Diagonals carrying points from S_1 , whose potential arc-contribution term is no longer a candidate to yield the minimal E' (as computed by Equation 5) for some future point in S_2 , are discarded from this subset. Intervals corresponding to target diagonals that are smaller than the dynamically growing x_max are also discarded. The interested reader is referred to the candidate list algorithms of ([5], [7], [8], [9], [14]) which also utilize convexity/concavity properties.

Definition 9. *Given x and x' , $x < x' \leq n + t$, $\text{Intersect}(x, x')$ is the minimal index y , $x' < y \leq n + t$, such that $D[x] + w(x, y) \leq D[x'] + w(x', y)$. $\text{Intersect}(x, x')$ is ∞ if there is no such index y .*

(In the example of Figure 11: $\text{Intersect}(5, 7) = 13$.)

By Conclusion 1, the source diagonal numbers giving the minima for E' , computed for increasing target diagonal numbers, are nonincreasing when w is convex. Furthermore, for log functions $\text{Intersect}(b, a)$ can

be computed in constant time. A candidate diagonal x is *live* if it supplies the minimum for some $E'[y]$. The algorithm maintains live diagonals and their leadership intervals in which these diagonals give the minimum (see Figures 9 and 11) using a priority queue data structure. Computing $E'[y]$ then reduces to looking up which leadership interval contains y . Decreasing $D'[x]$ involves updating the interval structure by deleting some neighboring live diagonals and finally a constant time *Intersect* computation at each end.

The psuedo-code for the algorithm is given in the attached appendix.

2.3 The Complexity of Sparse AHP

Theorem 1. *The algorithm described in this section computes sparse AHP with logarithmically growing destabilizing functions in time $O(s \log \log s)$.*

Proof. During a preprocessing stage, the set S of stacked triplets can be computed in $O(s + n \log \Sigma)$ time, where Σ is the size of the alphabet (four nucleotides in this case), using standard string matching techniques (suffix trees).

Within each level of the recursion, we will need the points of each set to be sorted by their column indices and only then by their row indices. To achieve this we initially bucket-sort all points, and then at each level of the recursion perform a pass through the sorted list to divide it into the two sets. Thus the order we need is achieved at a linear cost per level of the recursion. We also need a data structure to efficiently support the following two operations:

1. Compute the value of $E'[y]$ for some $y \in S$.
2. Decrease the value of $D'[x]$ for some $x \in S$.

Note that Operation 2 involving one value of $D'[x]$ may simultaneously change $E'[y]$ for many ys . A candidate list can be implemented to maintain the live diagonal candidates and support union, find and split operations on their leadership intervals. If a balanced search tree is used for implementing this list the amortized time per operation is $O(\log n)$. The bound can be improved to $O(\log \log n)$ with van Emde Boas's data structure [24]. Note that the time for each data structure operation can be taken to be $O(\log s)$ or $O(\log \log s)$ rather than $O(\log n)$ or $O(\log \log n)$. This is because only diagonals of the dynamic programming matrix that actually contain some of the s points in the sparse problem need to be considered. The van Emde Boas flat trees can be set up at $O(s)$ preprocessing time and then be reused at different levels of the recursion. Thus the arc-contribution of S_1 to S_2 can be computed in $O(s \log \log s)$ time. Multiplying by the number of levels of recursion, and adding the work spent on arc-connecting consecutive k -block pairs, the

total time is $O(n + s \log k \log \log s) \stackrel{k \leq s}{=} O(s \log \log s)$. \square

Theorem 2. *The sparse AHP Decision problem with logarithmically growing destabilizing functions and discrete DHES cost can be computed in time $O(s)$.*

Proof. A score function f is said to be discrete only if there exists some constant r such that every element of f is some integral multiple of r . Let $min_triplet$ denote the minimal cost of a triplet of base pairs. When computing AHP Existence (see Definition 6.2), the algorithm immediately halts once an e -scoring duplex is found. Therefore, the range of scores does not exceed $e - min_triplet$. Let $c = (e - min_triplet)/r$. Clearly, at any given moment during the execution of the algorithm there can be only c different DHES scores which are shared by all candidate diagonals from S_1 . By Conclusion 1, for any two diagonal candidates x' and x from S_1 such that $x' < x$, if $G[x'] = G[x]$, where $G[x]$ denotes the maximal score of a point from S_1 on diagonal x , then only x is a live candidate. Therefore, the total number of live candidates in a given moment is c and the time complexity of the algorithm in this case is therefore $O(s \log \log c) \stackrel{c \text{ is constant}}{=} O(s)$. \square \square

3. Computing the Association Operation F

The association operator (see Definition 4 in Section 1), $F(X_i, T_j)$, represents the probability that miRNA j is active under condition i . It is set to the confidence level by which we reject the hypothesis that $\{X[i, k] \cdot T[k, j]\}_{k=1}^r$ and $\{X[i, k] \cdot \overline{T[k, j]}\}_{k=1}^r$ were sampled from the same distributions. We use the following theorem to calculate a statistic t for the difference between $\{X[i, k] \cdot T[k, j]\}_{k=1}^r$ and $\{X[i, k] \cdot \overline{T[k, j]}\}_{k=1}^r$ and then set $F(X_i, T_j)$ to $\Phi(t)$, the corresponding level of significance associated with t . Note that $\Phi(t)$ represents the area below the normal distribution curve corresponding to t .

Theorem 3 [2]. *Let x_1 and x_2 be two observations and let n_1 and n_2 be their sizes respectively. Let \bar{x} denote the mean of sample x , and SD_x its standard deviation.*

Let $SD_x = \sqrt{\frac{SD_{x_1}^2(n_1-1) + SD_{x_2}^2(n_2-1)}{n_1+n_2-2}}$. If x_1 and x_2 were sampled from the same normal population and assuming that $n_1 + n_2 \geq 30$ then $t = \frac{\bar{x}_1 - \bar{x}_2}{SD_x \sqrt{1/n_1 + 1/n_2}}$ has an approximate standard normal distribution.

4. miRNAXpress Performance

In order to benchmark the performance of miRNAXpress we compared its running times to those of the naive dynamic programming (DP) approach (see Figure 3 and Section 1.1).

We constructed a large dataset consisting of pairs of sequences to which we applied the two methods. Each pair contains two sequences: the first — of size 20 — corresponding to a microRNA, and the second corresponding to a mRNA. The size of the second sequence varied between the pairs and was between 200 and 3200 nucleotides. In each pair, both sequences were randomly generated by a toss of a coin of their symbols.

The table in figure 14 shows the running times of the two algorithms (miRNAXpress target prediction engine versus the naive) for several representative sizes of the second sequence (*i.e.* the mRNA); the graph in Figure 14 also demonstrates the relationship between the running times of the two algorithms. Each entry of the table (or point in the graph) is an average over 1000 corresponding pairs. Both the graph and the table clearly indicate that miRNAXpress is indeed much quicker in practice than the naive dynamic programming method and that this trend becomes stronger as the size of the mRNA sequences grows.

5. A Study Associating *A. thaliana* microRNA with various Activity Conditions

Different gene expression profiles are part of how a plant grows, develops, and adapts to environmental changes. In this section we use our computational approach to shed light on the contributions of various microRNAs to these expression profiles in *A. thaliana*. The Expression Matrix X is based on 380 conditions and 5800 genes. The expressions were measured using a two dye microarray assay where the condition sample was labeled red and the reference sample was labeled green. The red/green ratio presents the abundance of the gene's mRNA in the condition, in comparison to the reference. A low red/green ratio can indicate that the mRNA is more degraded in the condition than in the reference, and vice versa for high ratios. We used the average values of \log_2 of the normalized red/green ratio. The reference samples changed between the conditions and are indicated at the relevant positions. The microarray data for the 380 different conditions was retrieved from the *TAIR* database, <http://www.Arabidopsis.org/>.

The T matrix is based on 98 previously discovered microRNAs. These were retrieved from the RFAM database, <http://www.sanger.ac.uk/Software/Rfam/>, and were discovered either experimentally (by isolating, reverse transcribing, cloning, and sequencing small cellular mRNAs) or computationally (by seeking microRNA precursors conserved between *A. thaliana* and other related organisms [19]). The mRNA sequences of all 5800 *A. thaliana* genes were retrieved from the *TAIR* database. To construct the T matrix we used the untranslated regions (3'UTRs) of these molecules (since 3'UTRs control mRNAs stabilities).

Table 1 describes some relations corresponding to significant entries of the Association Matrix A , and Table 2 includes the corresponding predicted targets. The results presented in Table 1 were selected based on their p -values. From all the microRNAs with p -value better than 10^{-6} we chose 6 for further investigation. The symbol $\bar{x}_{targets}$ in Table 1 represents the average expression of the set of the potential microRNA targets, i.e. $\sum_{l=1}^r \{X[i, l] \cdot T[l, j]\} / \sum_{l=1}^r \{T[l, j]\}$, and \bar{x}_{others} is the average expression of all other genes i.e. $\sum_{l=1}^r \{X[i, l] \cdot \overline{T[l, j]}\} / \sum_{l=1}^r \{\overline{T[l, j]}\}$. Since we use relative expressions, if $\bar{x}_{targets} < \bar{x}_{others}$ we hypothesize that the microRNA is active at the condition and not in the reference. On the other hand, if $\bar{x}_{targets} > \bar{x}_{others}$ we hypothesize that the microRNA is silenced at the condition and is active at the reference. Both cases can result in a significant p -value as represented by column $A[i, j]$. Moreover, to assess both cases one naturally needs to use a two tail test such as the t-test we used which is described in Section 3. Note that negative values of $\bar{x}_{targets}$ and \bar{x}_{others} reflect higher expression in the reference compared to the condition, since we are using the \log_2 of the red/green ratio.

$A[i, j]$ is a p -value for whether the expression of the targets is distinguished from the expression of all other genes. For this p -value to be correct the expression levels of the genes should be normally distributed (see Section 3). Although in practice this is the case for most microarray measurements, we further asserted our p -values by conducting the following shuffling simulations. We first randomly shuffle the mRNA sequences with respect to the genes and then computed a new T matrix. Using this matrix and the E matrix we computed the A matrix again. Repeating this process 1000 times allows us to estimate the *false/positive* rate of our approach. For example, if the lowest number computed for an entry $A[i, j]$ in all 1000 different assays is x , we conclude that the null probability for obtaining a value $A[i, j] \leq x$ is smaller than 10^{-3} . The smallest and average numbers computed for the relevant entries are also presented in Table 1 (see $\min A[i, j]_{shuffled}$ and average $A[i, j]_{shuffled}$, respectively). Note that the p -values computed for the microRNAs are significantly better than these numbers, strengthening the power of our approach and the significance of our results.

Our high throughput approach to computing A enables us to focus on the more interesting associations, e.g.:

1. miR159C was found to be more active when *nph4* mutants were subjected to phototropic stimulation than when wild-type (non-mutant) cells did. A phototropic stimulation is the process of exposing a plant to light after a long period of darkness. In this experiment, for example, plants were grown in the dark for

2.5 days and then exposed to blue light for one hour. NPH4 is a regulatory protein which is crucial for the normal response of the plant to a phototropic stimulation. Our findings indicate that one derivative of this protein action might be miR159C, which was found to be more active in *nph4* mutants than in wild-types, possibly indicating NPH4 pathway silencing of miR159C in response to light stimulus in wild-types. Indeed, many of the predicted targets (as in Table 2) of miR159C are involved in light processes, *e.g.* belong to endomembrane systems, or have photoreceptor activity, or code for chloroplast protein. The loss of the phototropic response in the mutant might be partly due to the loss of suppression of miR159C by NPH4. Intriguingly, miR159C seems to be active also in flowers that do not respond to phototropic stimulations. Moreover, since flowers do not perform photosynthesis they do not need most of miR159C's target products.

2. We found mir168A to be active in response to bacterial pathogen inoculation. This is a condition in which a virus virulent protein is introduced into the plant cells. Supporting evidence for our findings is the fact that AT5G60800, one of mir168A targets (Table 2), is a metal binder. Metals have been shown to facilitate viral attacks in plants [21]. Our findings, therefore, suggest that part of the virus strategy for attacking the cells is in elevating metal levels by activating mir168A and eliminating the metal binder, AT5G60800.

The very same microRNA is also activated in Xanthophyll cycle mutants. Xanthophyll cycle is a photoprotection mechanism. We propose that AT5G60800 is also related to this observation, since it is capable of binding free radicals which are typical to light exposure. We propose that in normal cells mir168A is de-activated by the Xanthophyll cycle in order to elevate the levels of AT5G60800 and reduce light damage.

3. miR399A was identified to be silenced under Auxin response. Concurrently, one of miR399A predicted targets, AT3G07390, is an Auxin responsive gene. This gene should be necessary for the cell's response to that hormone.

It is interesting to note that our findings give rise to hypotheses about microRNA regulation in the positive direction (conditions that induce microRNA activity) as well as in the negative direction (pathways that reduce microRNA activity).

Lastly, we looked for tissue specific microRNAs (see Figure 15). We focused on the subset of rows in *A* corresponding to flowers, stems, siliques, leaf, and root tissue conditions. Within this subset of selected rows we chose columns with only one significant entry. The microRNAs corresponding to such columns are potentially specific to the tissue represented by this row.

Acknowledgements: We thank Eleazar Eskin for fruitful discussions and Uri Levy for contributing code.

The research of M. Z.-U. was supported in part by the Aly Kaufman Post Doctoral Fellowship.

References

- [1] A. Aggarawal and J. Park. Notes on searching in multidimensional monotone arrays. *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 497–512, 1988.
- [2] C. Chatfield. *Statistics for technology, a course in applied statistics*, Sci. Papreback, 1970.
- [3] D.V. Dugas and B. Bartel. MicroRNA regulation of gene expression in plants. *Curr. Opin. Plant Biol.*, 7:512–520, 2004.
- [4] A.J. Enright et al. MicroRNA targets in drosophila. *Genome Biol.*, 5(1), 12 2003.
- [5] D. Eppstein, Z. Galil, and R. Giancarlo. Speeding up dynamic programming. *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 488–296, 1988.
- [6] D. Eppstein, Z. Galil, R. Giancarlo, and G.F. Italiano. Sparse dynamic programming I: Linear cost functions. *JACM*, 39:519–545, 1992.
- [7] D. Eppstein, Z. Galil, R. Giancarlo, and G.F. Italiano. Sparse dynamic programming II: Concave and convex cost functions. *JACM*, 39:519–545, 1992.
- [8] Z. Galil and R. Giancarlo. Speeding up dynamic programming with applications to molecular biology. *Theoretical Computer Science*, 64:107–118, 1989.
- [9] D.S. Hirshberg and L.L. Larmore. The least weight subsequence problem. *SIAM J. Compt.*, 16(4):628–638, 1987.
- [10] K.D. Kasschau et al. P1/HC-Pro, a viral suppressor of RNA silencing, interferes with Arabidopsis development and miRNA function. *Dev. Cell*, 4:205–217, 2003.
- [11] L. Larmore and B. Schieber. On-line dynamic programming with applications to the prediction of RNA secondary structure. *J. Algorithms*, 12(3):490–515, 1991.
- [12] C. Llave et al. Cleavage of scarecrow-like mRNA targets directed by a class of Arabidopsis miRNA. *Science*, 23:2053–2056, 2002.
- [13] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, 1999.
- [14] W. Miller and E. Myers. Sequence comparison with concave weighting functions. *Bull. of Mathematical Biology*, 50(2):97–120, 1988.

- [15] E. Myers and W. Miller. Chaining multiple-alignment fragments in sub-quadratic time. *ACM-SIAM Symposium on Discrete Algorithms*, pages 1–10, 1995.
- [16] J.F. Palatnik et al. Control of leaf morphogenesis by microRNAs. *Nature*, 425:257–263, 2003.
- [17] N. Rajewsky and N.C. Socci. Computational identification of microRNA targets. *Genome Biology*, 5, 2004.
- [18] M. Rehmsmeier, P. Steffen, M. Hochsmann, and R. Giegerich. Fast and effective prediction of microRNA/target duplexes. *RNA*, 10:1507–1517, 2004.
- [19] M.W. Rhoades et al. Prediction of plant microRNA targets. *Cell*, 23:513–520, 2002.
- [20] J. Setubal and J. Meidanis. Introduction to computational molecular biology. 1997.
- [21] A. Shevchenko et al. Plant virus infection development as affected by heavy metal stress. *Archives of Phytopathology and Plant Protection*, 23:139–146, 2004.
- [22] A. Stark et al. Identification of Drosophila microRNA targets. *PLoS. Biol.*, 1(3), 2003.
- [23] G. Tang et al. Framework for RNA silencing in plants. *Genes Dev.*, 17:49–63, 2003.
- [24] P. van Emde Boas, R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Mathematical Systems Theory*, 10:99–127, 1977.
- [25] M.S. Waterman and T.F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Adv. Appl. Math.*, 7:455–464, 1986.
- [26] Z. Xie et al. Negative feedback regulation of dicer-like1 in Arabidopsis by microRNA-guided mRNA degradation. *Curr. Biol.*, 13:784–789, 2003.
- [27] S. Yekta. MicroRNA-directed cleavage of HOXB8 mRNA. *Science*, 304:594–596, 2004.
- [28] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, 31(13):3406–3415, 2003.

An Appendix of Pseudo-Code

Algorithm 1: RNA Approximate Bounded Score Hybridization Search

input : microRNA pattern P of size m , mRNA pattern T of size n , and a score threshold e .

Output: Potential Hybridization sites of P with T , such that $DHES(P, T) < e$.

```
1 find a sparse set  $S$  of possible triplets of stacked base pairs from the two strings;
2 bucket-sort  $S$  by column numbers first and then by row numbers;
3  $num\_segments = m/k$ ;
4 for  $i = 1$  to  $num\_segments$  do
5   Let  $S_i$  denote the base pairs whose row number falls between  $(i - 1) \times k$  and  $i \times k$ ;
6   Let array  $E$  be indexed by members of  $S_i$ ;
7   for  $x \in S_i$  do
8     Let  $\delta[x]$  denote the energy of the triplet of base pairs ending in  $x$ ;
9     set  $E[x]$  to  $\delta[x]$ ;
10  end
11  if  $i > 1$  then
12    ContributeByArcs( $S_{i-1}, S_i$ );
13  end
14  Recurse( $S_i, k$ );
15 end
16 report all  $x \in S$  with score  $< e$ ;
```

Algorithm 2: Procedure $Recurse(S, nr)$

```
1 Let  $i$  denote the middle row of  $S$ ;
2  $this\_nr = nr/2$ ;
3 Let  $S_1$  be the points above or on row  $i$  in  $S$ ;
4 Let  $S_2$  be the points below row  $i$  in  $S$ ;
5 if  $this\_nr > 3$  then
6    $Recurse(S_1, this\_nr)$ ;
7    $ContributeByArcs(S_1, S_2)$ ;
8    $Recurse(S_2, this\_nr)$ ;
9 end
```

Algorithm 3: Procedure *ContributeByArcs*(S_1, S_2)

```
1 Let  $I$  be the candidate leadership interval list (the search key is the index of the largest diagonal in the interval), implemented by the data structure of section 2.3;
2 Let  $L$  be another list of candidates (the search key is the diagonal number of the candidate), also implemented by the data structure of section 2.3;
3 Let  $G[d]$  denote the minimum for diagonal  $d$ ;
4 Let  $X = S_1 \cup S_2$ , maintaining sorted order (the points in  $S_1$  always lag behind  $S_2$ );
5 for  $x \in X$  in order do
6    $d \leftarrow \text{row}(x) + \text{column}(x)$ ;
7   if  $x \in S_1$  then
8      $G[d] \leftarrow \min\{G[d], E[x]\}$ ;
9      $\text{max}_x = \max\{\text{max}_x, d\}$ ;
10    ClearIrrelevantCandidates( $I, L, \text{max}_x$ );
11    UpdateCandidateLeadershipIntervals( $d, G[d], I, L$ );
12  end
13  else
14     $E[x] \leftarrow \min\{E[x], \text{QueryArcContribution}(I, d) + \delta[x]\}$ ;
15  end
16 end
```

Algorithm 4: Procedure *UpdateCandidateLeadershipIntervals*($d, G[d], I, L$)

```
1  $\ell \leftarrow$  the smallest active leader candidate that is greater than or equal to  $d$ ;
2  $i \leftarrow$  the last diagonal in the leadership interval of  $r$ ;
3 while  $G[d] + w(d, i) < G[r] + w(\ell, i)$  do
4   Remove( $i, I$ );
5   Remove( $\ell, L$ );
6    $\ell \leftarrow$  the smallest active candidate that is greater than  $d$ ;
7    $i \leftarrow$  the number of the last diagonal in the leadership interval of  $\ell$ ;
8 end
9  $c \leftarrow \text{Intersect}(d, \ell)$  /*  $c$  is the last diagonal in the new leadership interval of  $d$  */;
10 Insert( $c, d, I$ );
11 Insert( $d, L$ );
12  $\ell \leftarrow$  the largest active candidate that is smaller than  $d$ ;
13 Repeat the above while loop in reverse direction;
```

Algorithm 5: Procedure *QueryArcContribution*(I, d)

```
1  $i = \text{Find}(d, I)$ ;
2  $\ell = \text{GetLeaderByInterval}(i)$ ;
3 return  $G[\ell] + w(\ell, d)$ ;
```

Algorithm 6: Procedure *ClearIrrelevantIntervals*(I, L, max_x)

```
1  $i \leftarrow SmallestIntervalIn(I)$ ;  
2  $\ell \leftarrow LeaderOf(i)$ ;  
3 while  $i < max\_x$  do  
4    $Remove(i, I)$ ;  
5    $Remove(\ell, L)$ ;  
6    $i \leftarrow SmallestIntervalIn(I)$ ;  
7    $\ell \leftarrow LeaderOf(i)$ ;  
8 end  
9 set  $i$  of  $\ell$  to  $max\_x$ ;
```

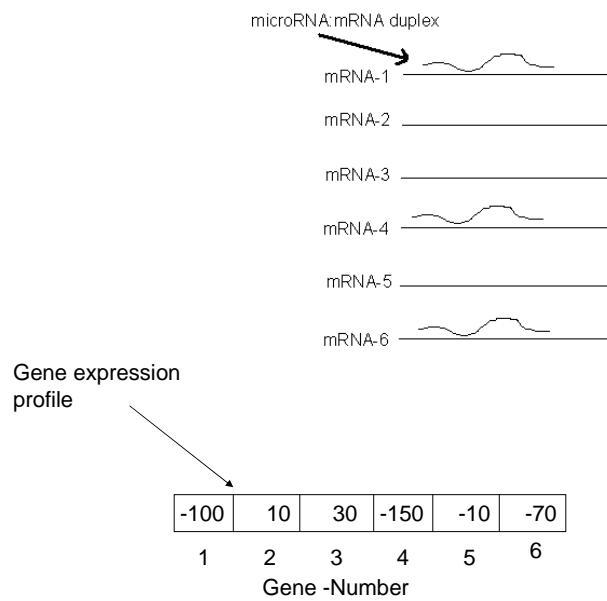


Figure 1: Associating a microRNA and a specific condition. The predicted targets of microRNA Y (*i.e.* computed to have the potential to form a stable duplex with it) are $mRNA_1$, $mRNA_4$ and $mRNA_6$. According to the expression profile at the bottom of the figure, the expression levels of these predicted target mRNAs ($\{-100, -150, -70\}$ correspondingly) are significantly low in comparison to the rest of the mRNAs ($\{10, 30, -10\}$) which participate in this test.

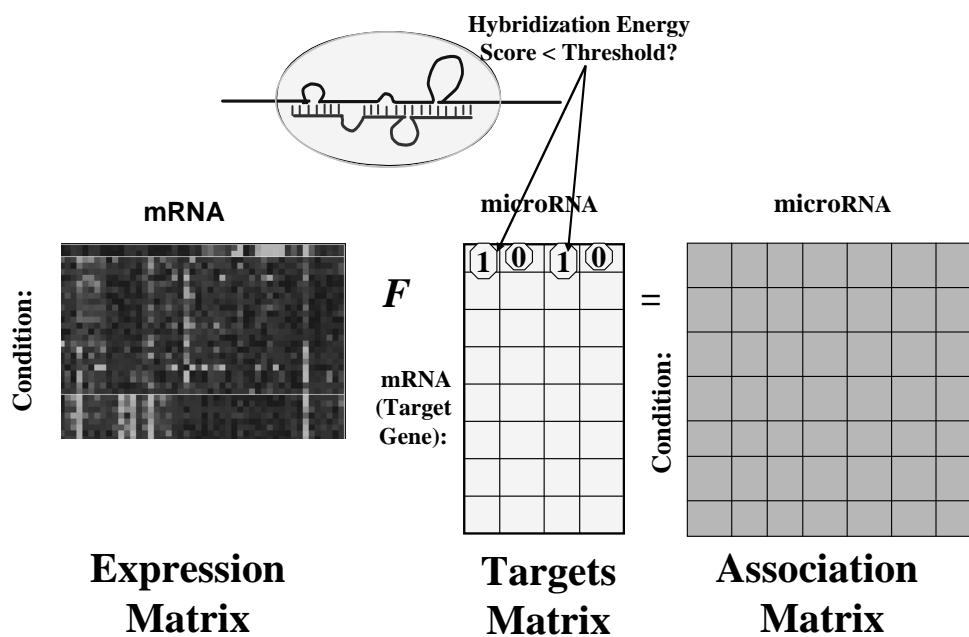


Figure 2: Constructing the Associating matrix A , by applying the association operation F between each row of the target matrix T and each column of the expression matrix E . Note that the 1's in the T matrix correspond to cases where the predicted duplex between the microRNA and the target mRNA were stable (below a predefined energy threshold).

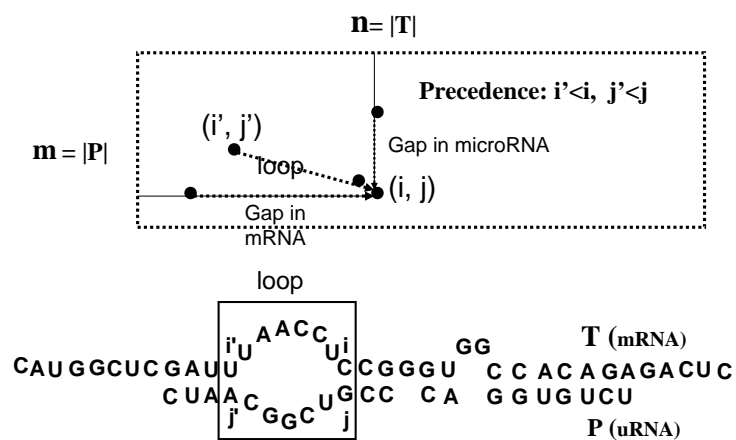


Figure 3: The naive dynamic programming algorithm for computing the energetically most favorable duplex between a short query RNA and a long target RNA using nearest-neighbor thermodynamic scoring rules. Note that computing the score of each cell (i, j) involves the consideration of all the cells (i', j') preceding it (i.e. left and above such that $i' < i$ and $j' < j$) in the DP table, which leads to an $O(m^2n^2)$ time complexity.

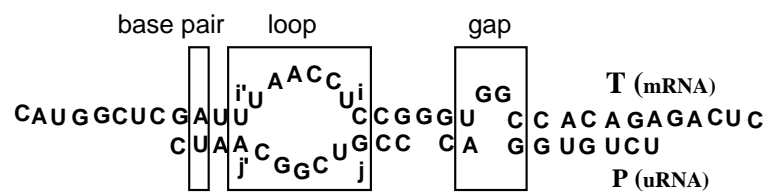


Figure 4: The energy of a duplex computed as the sum of its components: base pairs, loops and bulges.

$$E[i, j] = \min\{ D[i', j'] + \ln(i - i' + j - j') \}$$

for all $(i', j') \mid i' < i, j' < j$

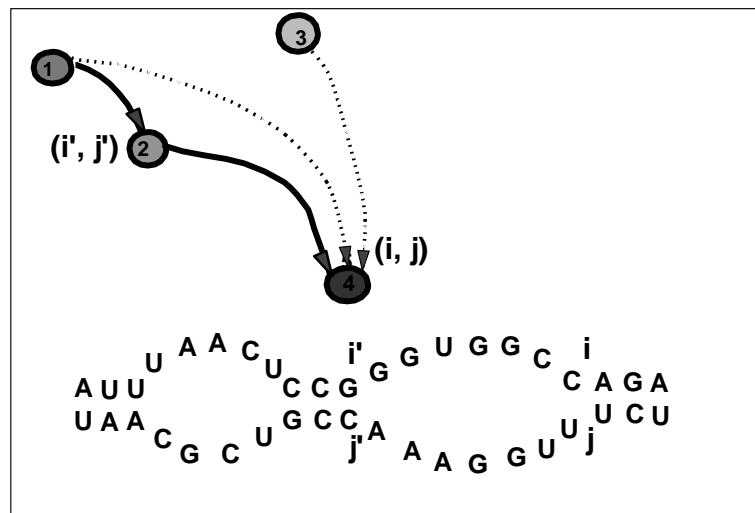


Figure 5: A sparse representation of the dynamic programming table. The score of point 4 is computed as the minimum of three sums of pairs each consisting of a score for a preceding point plus the cost of the loop connecting the preceding point to the new point.

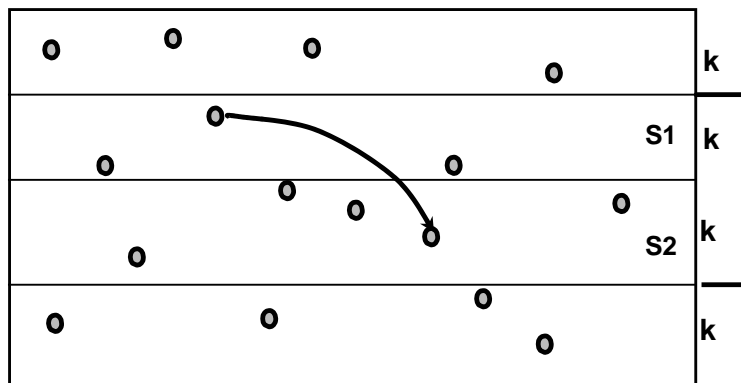


Figure 6: The division of the dynamic programming table into m/k slices of size k each. Each arc is restricted to a sliding window of $2k$ consecutive rows.

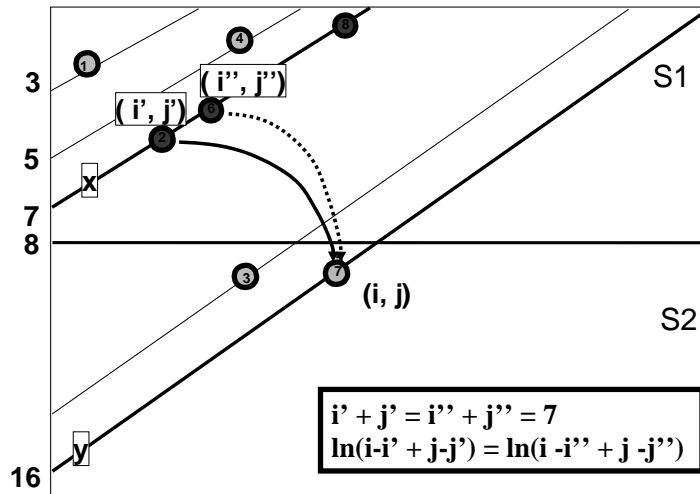


Figure 7: When computing the score of the point $(i, j) \in S_2$ we only need to know its diagonal number (*i.e.* 16 in this example) plus, for each diagonal in S_1 , the value of the point on that diagonal which carries the lowest value among all those preceding point (i, j)

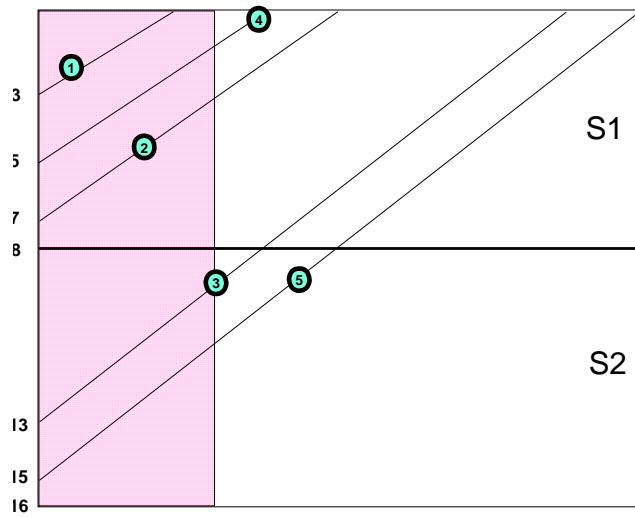


Figure 8: Computing the arc-contribution of the first 2 points in S_1 to the first point in S_2 .

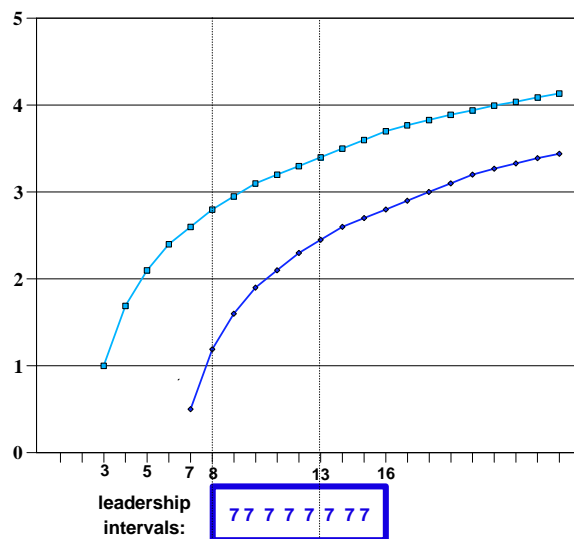


Figure 9: The graphs representing the score terms contributed by Points 1 and 2 to the computation of E' for Point 3 in Figure 8.

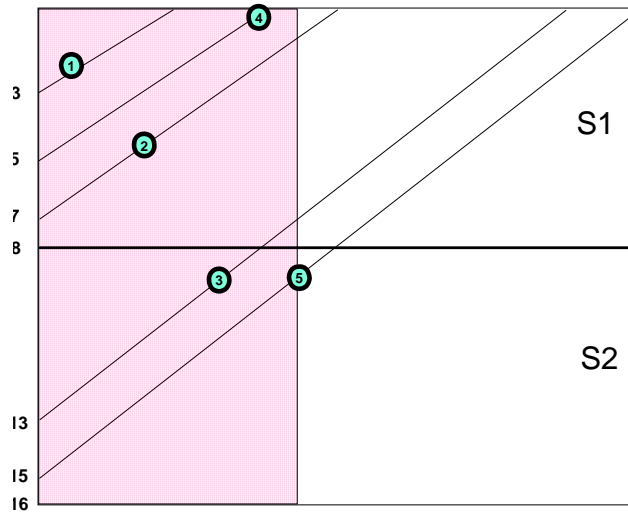


Figure 10: Computing the arc-contribution of the first 3 points in S_1 to the second point in S_2 .

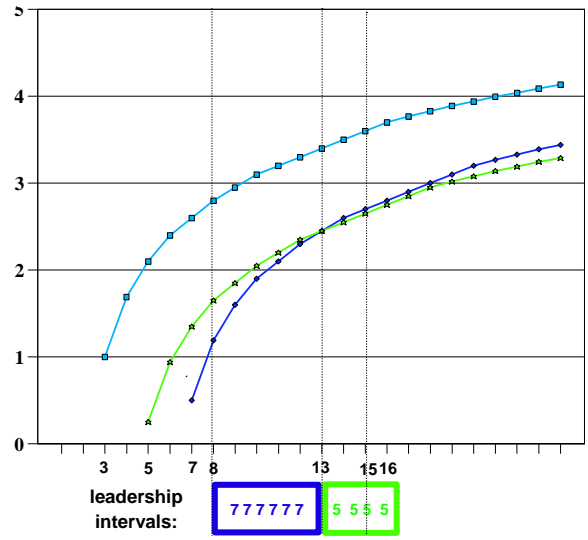


Figure 11: The graphs representing the score terms contributed by points 1, 2 and 4 to the computation of E' for point 5 in Figure 10.

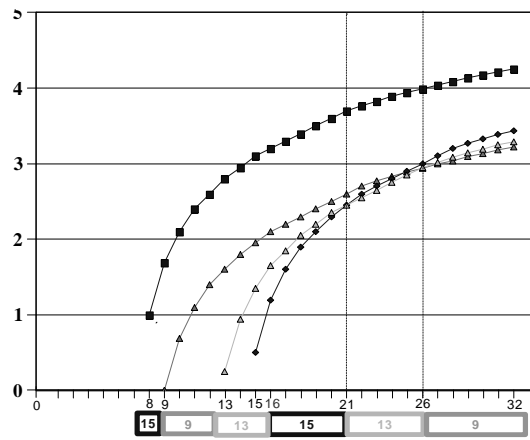


Figure 12: The fragmentation of the leadership interval prior to x_{max} .

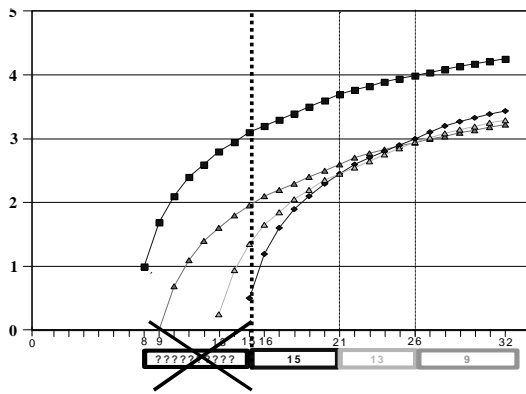


Figure 13: Using the divide and conquer and traversal order described above, the fragmented leadership intervals, which by definition fall to the left of x_{max} , are no longer relevant to the minimization computation.

<i>microRNA</i>	<i>Condition</i>	$A[i, j]$	\bar{x}_{others}	$\bar{x}_{targets}$	<i>min</i> $A[i, j]$ shuffled	<i>average</i> $A[i, j]$ shuffled
miR168A	Xanthophyll mutant vs wildtype	10^{-22}	0.03	-1.82	0.009	0.48
miR168A	Arabidopsis leaves inoculated in bacterial pathogen vs non non inoculated leaves	10^{-6}	0.01	-1.76	0.045	0.47
miR168A	Shoot exposed to light after 1hr darkness vs unexposed shoot	10^{-9}	1	0.41	0.25	0.5
miR156A	Defective DST-mediated mRNA degradation pathway vs wildtype	10^{-14}	22	1.23	0.05	0.52
miR159C	Phototropic stimulation of NPH4 mutants vs phototropic stimulation of wildtype	10^{-11}	0.07	0.62	-0.85	0.1
miR399A	Roots treated with Auxin compare to untreated roots.	10^{-11}	-0.85	0.07	0.1	0.55

Table 1: MicroRNAs and associated conditions. For Bonferroni corrected p-values, entries of the table should be multiplied by 10^4 .

<i>microRNA</i>	<i>Target list</i>	<i>Hybridization energy</i>	<i>GO term</i>
<i>mir168a</i>	AT3G26420 AT5G60800	-43.2 -41.7	RNA binding Metal ion binding
<i>miR159C</i>	AT4G08320 AT3G26420 AT4G02520 AT3G10770 AT1G78880 AT4G01810 AT1G09570 AT1G73060 AT4G22890 AT4G24230 AT5G40760 AT2G24940	-35.90 -36.30 -36.70 -41.40 -37.30 -36.10 -37.30 -36.80 -39.60 -38.40 -36.00 -36.90	Endomembrane system RNA binding Glutathione transferase Glutathione transferase Unknown Photoreceptor activity Photoreceptor activity Unknown Chloroplast protein Acyl-CoA binding Unknown Electron transport
<i>mir399C</i>	AT4G24470 AT1G51200 AT3G13460 AT3G07390 AAT3G27260	-37.50 -38.10 -36.10 -36.80 -36.30	Transcription factor Electron transporter Unknown Response to Auxin DNA binding
<i>mir395</i>	AT3G26420 AT2G31800 AT1G55255 AT1G78080 AT5G61170 AT1G73060 AT5G27650 AT5G35360 AT2G36400 AT3G52800 AT4G24230 AT1G60730 AT2G43970 AT3G18210 AT5G10860 AT1G08990 AT5G54300 AT5G15320 AT3G50960	-38.20 -38.70 -36.80 -36.20 -36.50 -42.20 -36.60 -36.10 -36.70 -36.80 -36.90 -36.10 -37.90 -36.60 -38.40 -37.50 -36.20 -38.50 -36.60	RNA binding Kinase activity Zinc ion binding Transcription factor Ribosomal RNA Chloroplast protein Unknown Chloroplast protein Chemokine receptor Zinc ion binding Endomembrane system Kcl channel RNA binding Unknown Acetolactate synthase Endomembrane system Sodium ion transport Endomembrane system Unknown

Table 2: microRNAs and their potential targets.

size of T	miREX time(sec)	Naïve time(sec)	Naive/miREX mean +- SD
200	0.041	1.10	26.83 +- 0.09
400	0.048	4.44	92.50 +- 0.33
800	0.066	18.70	283.33 +- 2.03
1600	0.097	72.80	750.50 +- 6.09
3200	0.160	288.00	1800.00 +-24.49

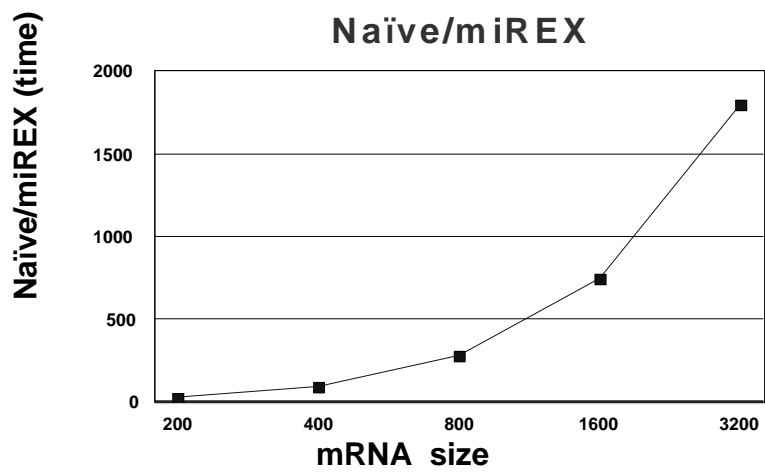


Figure 14: Average Naive/miRNAXpress running times vs. mRNA size. Each entry of the table (or point in the graph) is an average over 1000 corresponding pairs.

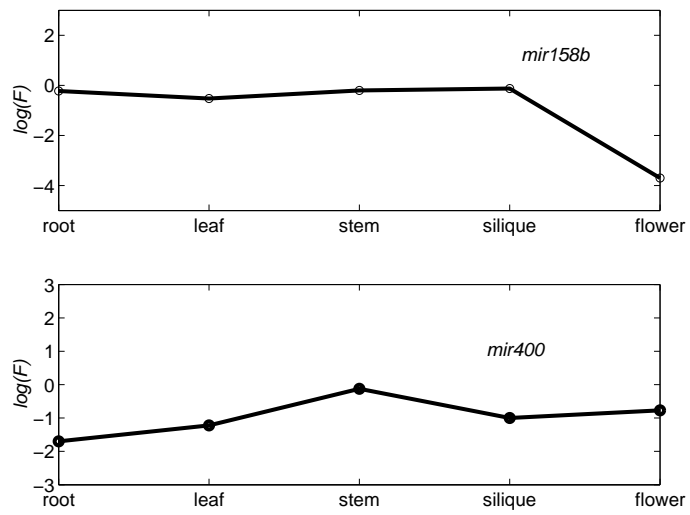


Figure 15: Tissue specific microRNAs. MicorRNAs that had significant $p - values$ only at specific tissues.