A fast *ab-initio* method for predicting miRNA precursors in genomes

Sébastien Tempel and Fariza Tahi*

Laboratoire IBISC, Université d'Evry-Val d'Essonne/Genopole, 23 Boulevard de France, 91034 Evry, France

Received September 2, 2011; Revised January 13, 2012; Accepted January 23, 2012

ABSTRACT

miRNAs are small non coding RNA structures which play important roles in biological processes. Finding miRNA precursors in genomes is therefore an important task, where computational methods are required. The goal of these methods is to select potential pre-miRNAs which could be validated by experimental methods. With the new generation of sequencing techniques, it is important to have fast algorithms that are able to treat whole genomes in acceptable times. We developed an algorithm based on an original method where an approximation of miRNA hairpins are first searched, before reconstituting the pre-miRNA structure. The approximation step allows a substantial decrease in the number of possibilities and thus the time required for searching. Our method was tested on different genomic sequences, and was compared with CID-miRNA, miRPara and VMir. It gives in almost all cases better sensitivity and selectivity. It is faster than CID-miRNA, miRPara and VMir: it takes ~30 s to process a 1 MB sequence, when VMir takes 30 min, miRPara takes 20 h and CID-miRNA takes 55 h. We present here a fast *ab-initio* algorithm for searching for pre-miRNA precursors in genomes, called miRNAFold. miRNAFold is available at http://EvryRNA.ibisc.univ-evry.fr/.

INTRODUCTION

MicroRNAs (miRNAs) are non-coding RNAs which are only 21–25 nt in sequence length and are present in all sequenced higher eukaryotes (1,2). They are involved as negative regulators of gene expression at the post-transcriptional level by binding to specific mRNA targets whose translations are inhibited or downregulated (2,3). According to the current understanding of miRNA biogenesis, miRNA genes are transcribed first as long pri-miRNAs and then are cleaved into 60–80 nt long precursors of miRNA sequences (pre-miRNAs) by the Drosha/Pasha complex. The pre-miRNA, structured as a hairpin, is transported into the cytoplasm by Exportin5 and cleaved by Dicer into the mature miRNA (1). In the RISC complex, a miRNA binds to a specific mRNA transcript and leads to the cleavage or the degradation of the mRNA.

Since the detection of pre-miRNAs by experimental techniques is difficult, expensive and requires a large amount of time, computational methods represent the first step in pre-miRNA identification. These methods can be divided into three approaches: comparative genomics, homology-based approaches and *ab-initio* approaches.

The phylogenetic conservation of some pre-miRNAs in their primary sequence and/or their secondary structure (1,4) is used in comparative genomics approaches. These approaches consider multiple alignments of sequences where conserved pre-miRNAs are searched for. Several algorithms based on this approach were developed, for example miRseeker (5), MiRFinder (6), RNAmicro (7), BayesMiRNAfind (8), miRRim (9).

The increase of known pre-miRNAs in miRBase (www.mirbase.org) (10) permits homology-based approaches to exploit information from both sequence and structure. For example, miRAlign (11) uses sequence and structure filters to predict new pre-miRNAs. ERPIN (12) uses RNA alignments as weight matrices to look for homologous pre-miRNAs.

Comparative genomics and homology-based approaches cannot detect pre-miRNAs of unknown families and/or pre-miRNAs with no close homologues in genomes. Furthermore, comparative approaches do not work on new genomes that do not have a closely related species sequenced. *Ab-initio* methods are needed to predict new pre-miRNAs in genomes.

Almost all existing *ab-initio* algorithms use an early step secondary structure predictor like RNAFold (13), RNALFold (14), Mfold (15) or UNAFold (16). Different methods and filters are then applied for predicting pre-miRNAs.

*To whom correspondence should be addressed. Tel: +33 1 64853463; Fax: +33 1 69470604; Email: fariza.tahi@ibisc.univ-evry.fr

We can classify the *ab-initio* methods into three categories:

- Methods that take as input the sequence of a pre-miRNA candidate and then classify it as true or false pre-miRNA.
- Methods that take as input a genomic sequence and some other information in order to predict (several) pre-miRNAs in the given sequence.
- Methods that are completely *ab-initio*, since they take as input a genomic sequence only (without any other information) and then search for all possible pre-miRNAs occurring in the sequence.

In the first category, we have Triplet-SVM (17), mir-KDE, (18), miPred (19) and microPred (20). Triplet-SVM and miPred are algorithms that classify real and pseudo pre-miRNAs using, respectively, a support vector machine (SVM) and a random forest prediction model. mir-KDE transforms the secondary structure produced by RNAFold and RNAspectral (21) into a vector of 33 features that are then estimated with a relaxed variable kernel density estimator (RVKDE) (21). microPred classifies human pre-miRNAs using a SVM with 48 features, including 6 folding criteria (20).

In the second category, we have miR-abela (22), and MIReNA (23). miR-abela predicts new pre-miRNAs that are close in the sequence to a given known pre-miRNA; it searches for pre-miRNA clusters in human, mouse and rat genomes. In case of MIReNA, it is necessary to enter approximative positions of pre-miRNAs.

To our knowledge, there are very few *ab-initio* algorithms of the third category that search for pre-miRNA structures in whole genomes, without any given additional information. There are CID-miRNA (24), miRPara (25), miRPred (26), miRANK (27), Virgo (28) and VMir (29). CID-miRNA (24) uses a Stochastic Context Free Grammar (SCFG) model for predicting pre-miRNAs built for the human genome. miRPara (25) uses UNAFold (16) first to predict the secondary structure of the given sequence, and select the pre-miRNA candidates through 77 parameters such as the ratio of GC, the number of internal loops, the number of GU pairings and the number of unpaired nucleotides. miRPred (26) identifies pre-miRNA structures in the human genome using linear genetic programming, and miRANK (27) uses a ranking algorithm based on Markov random walks, a stochastic process defined on weighted finite state graphs. Virgo (28) uses RNAFold and hairpin and energy filters before using a SVM classifier, called SVM^{light} (30). Finally, VMir (29) was created for predicting pre-miRNAs in viruses; it uses RNAFold and calculates a score for RNAFold hairpins using several parameters like the size, the number of copies and the number of sliding windows where a same hairpin is detected.

With the new generation of genome sequencing technologies, it is nowadays important to have *ab-initio* automatic methods for quickly analyzing the newly sequenced genomes, and an important aspect of this analysis is the prediction of pre-miRNAs.

In this article, we present a new *ab-initio* method (belonging to the third category), called miRNAFold, for predicting pre-miRNA structures in any genome. We developed an algorithm that, given a genomic sequence (of any length), searches directly for pre-miRNA hairpins occurring in that sequence. It targets more precisely pre-miRNA structures by taking into account their characteristics, in order to (i) better select the true pre-miRNAs and (ii) reduce the search time. The main idea is to first search for a long hairpin stem, which is considered as an anchor allowing to predict the hairpin structure [the idea of anchor was initially used in Tfold (31) for RNA secondary structure prediction and in ModuleOrganizer (32) for the detection of modules in repeated sequences].

miRNAFold was tested on an artificial sequence and on several real genomic sequences. It was compared with CID-miRNA (24), miRPara (25) and VMir (29). We show in this article that our algorithm predicts successfully almost all known pre-miRNAs in genomic sequences of different species. It gives better or at least similar sensitivity and selectivity than CID-miRNA, miRPara and VMir. We also show that our algorithm is very fast; it takes <30 s to process a 1 Mb sequence, while VMir takes >30 min, miRPara ~20 h, and CID-miRNA >55 h.

MATERIALS AND METHODS

Pre-miRNA features

Our first objective was to find features of pre-miRNAs. For this purpose, we downloaded the last version of miRBase database (Release 17, April 2011) that contains 16 772 pre-miRNAs (10) and we studied the pre-miRNAs contained in this database. We then observed several characteristics:

Pre-miRNA hairpins contain long stems. We observed that pre-miRNAs are almost always composed of at least one long exact stem. An exact stem is a couple of subsequences (p, p') such that:

$$\text{(i)} \quad |p| = |p'| = m$$

$$\text{(ii)} \quad p[k]\ R_c\, p'[m-k+1], \qquad \forall k, 1 \leq k \leq m$$

where R_c is the relation of complementarity between nucleotides: AR_cU, GR_cC and GR_cU. In other terms, an exact stem is a succession of base pairings A-U, C-G and G-U.

We observed that all pre-miRNA hairpins of miRBase have at least one exact stem of length greater or equal to 5. And as we can see in Figure 1A, the longest exact stem in pre-miRNA hairpins of miRBase is often between 5 and 10 nt.

Pre-miRNA hairpins are symmetric. We also observed that most pre-miRNAs either have very few bulges or bulges of one side almost compensate with bulges of the other side (i.e. there is a similar number of nucleotides on both sides of the hairpin from the terminal loop to

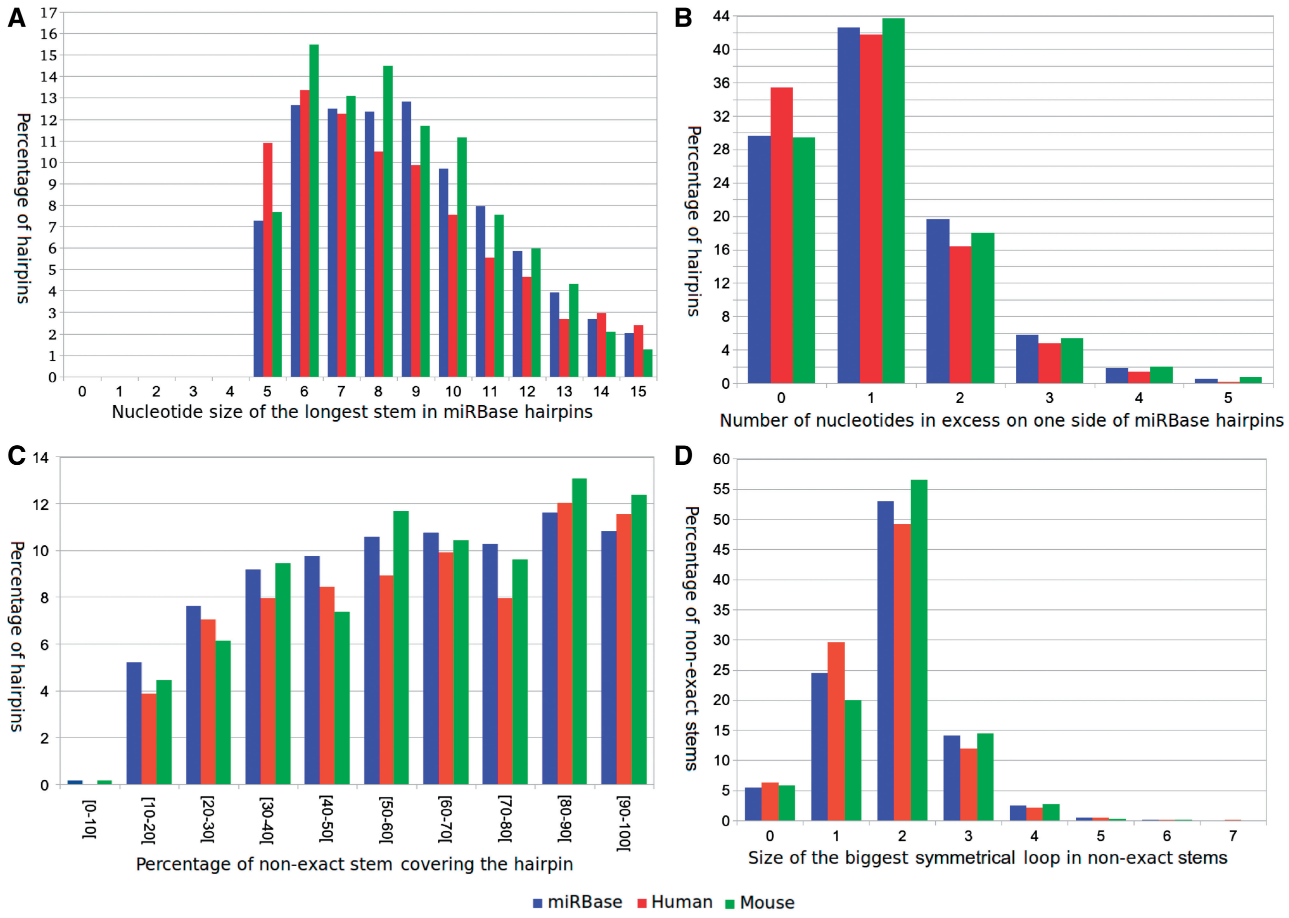


Figure 1. (**A**) Almost all known pre-miRNAs contain at least one long stem. Percentage of pre-miRNA hairpins in human genome, mouse genome and in all miRBase, in function of the length of their longest stem. (**B**) Almost all known pre-miRNAs do not contain big bulges. Percentage of pre-miRNAs in human genome, mouse genome and in all miRBase, having a gap of a given size, i.e. having an excess of nucleotides on one side of the hairpin. A gap of zero corresponds to a same number of nucleotides on both sides. (**C**) Almost all known pre-miRNAs are covered by a non-exact stem. Percentage of pre-miRNAs in miRBase in function of the percentage of nucleotides covered by a non-exact stem. (**D**) Size of the biggest symmetrical loop in a non-exact stem. Percentage of non-exact stems from miRBase in function of the size of their biggest symmetrical loop.

the extremities). Figure 1B shows the number of hairpins decreases when the gap increases. In all, 90% of pre-miRNAs have less than 3 nt in excess on one side. In other words, pre-miRNAs do not form a ‘curved’ hairpin but form an ‘almost straight’ hairpin.

Pre-miRNA hairpins can be approximated by a non-exact stem. We have observed that in almost all pre-miRNAs of miRBase, there is a non-exact stem. A non-exact stem is composed of a succession of exact stems separated by symmetrical loops such that the size of each symmetrical loop is less than the length of the exact stems surrounding it. We define a symmetrical loop as follows:

(i) $|p| = |p'| = m$

(ii) $p[k]\ \overline{R_{nc}}\ p'[m-k+1], \qquad \forall k, 1 \leq k \leq m$

where $\overline{R_{nc}}$ is the relation of non complementarity between nucleotides. The size of a symmetrical loop is the number of unpaired nucleotides on one side of the loop.

A non-exact stem forms an important part of the structure, often more than 40% (Figure 1C). This percentage corresponds to the ratio of the size of the non-exact stem size and the size of the hairpin (without the terminal loop). More than 75% of pre-miRNAs in miRBase have a non-exact stem that represents at least 40% of their length (Figure 1C).

Almost all miRBase pre-miRNAs have short symmetrical loops. In all, 91.5% of pre-miRNAs have symmetrical loops whose length ranges from 1 to 3 nt. Only 0.15% of miRBase pre-miRNAs (24 of 16,772) have a symmetrical loop bigger or equal to 6 (Figure 1D).

Other pre-miRNA features. By studying pre-miRNAs of miRBase, we observed several other characteristics. We split the features into two categories: global characteristics that are present in all species in miRBase and local characteristics that are species dependent.

For the global features, we observed that the longest stems are composed of at least three base pairings and have a ratio of GU base pair always lower than 33.33%. We also observed that the average size of the exact stems that make up non-exact stems is >3.

For the species-dependent features, we used some usual characteristics like the hairpin size, the minimum free energy (MFE) and the ratio of A, C, G and U nucleotides. MFE is calculated in the same way as in Mfold (15). We also calculated some characteristics from Helvik *et al.* (33) and van der Burgt *et al.* (34) like the MFE adjusted (i.e. ratio between MFE and length), the ratio of G-U and G-C base pairings and the ratio of G over C. All features and thresholds are listed in the Supplementary File 1.

Our approach

Our goal was to develop an algorithm which is able to find efficiently pre-miRNAs in whole genomes in an acceptable time. For this purpose, we adopted the following approach, which was motivated by the different observations (presented above) we made on miRBase pre-miRNAs.

We consider a sliding window of a given size L sufficiently long to contain a pre-miRNA, in which we search for pre-miRNA hairpins.

In a first step, we search for long exact stems that verify some criteria, so they are considered as anchors of possible hairpins. In a second step, we extend the selected stems in order to get the longest non-exact stems verifying some criteria. Each selected non-exact stem can correspond to a large portion of a pre-miRNA. It is therefore considered as a good approximation of a pre-miRNA hairpin, and gives the hairpin position. Possible pre-miRNA hairpins are then searched for in the subsequence associated to the selected non-exact stem, considering the middle position of the non-exact stem as the middle position of the hairpin. Hairpins verifying some criteria are then selected.

Thus, our approach consists of three main steps applied on each window subsequence:

(1) search for longest exact stems;
(2) extend the selected stems and select the longest non-exact stems; and
(3) predict the secondary structure of the hairpins corresponding to the selected non-exact stems.

At each step, several selection criteria are used, corresponding to several features observed on the pre-miRNA hairpins of miRBase and on their exact stems and non-exact stems. Because a pre-miRNA can present some of these features but not all, an exact stem, a non-exact stem or a hairpin is selected when a certain percentage of the criteria are verified. This percentage is a parameter which could be set by the user. There are 12 criteria for the longest exact stem, 17 criteria for the longest non-exact stem and 26 criteria for the hairpin. For example, if the user choses to select a hairpin with 80% of verified criteria, this means that the hairpin must have at least 9, 13 and 20 criteria that are verified for, respectively, the longest exact stem, the longest non-exact stem and the hairpin.

After the three steps, the sliding window is shifted by 10 nt. The overlapping sequence between two sliding windows allows the algorithm to find the complete secondary structure of the hairpin.

The algorithm

Given a genomic sequence of any size, for each subsequence delimited by the sliding window, a triangular base pairing matrix M is built such as:

$$M(i,j) = \begin{cases} M(i-1, j-1) + 1 & \text{if } M(i) \text{ and } M(j) \text{ form a base pair} \\ 0 & \text{otherwise} \end{cases}$$

The algorithm performs then the three following main steps.

Longest exact stem searching. Stems of length greater than a minimal size *lmin* are searched for in the matrix. For example in Figure 2A, three stems (surrounded by blue) are selected. The 10 longest stems verifying a certain percentage of criteria [set by default to 70% (see ‘Results’ section or given by the user)] are then selected. The 12 exact stem criteria are section the size of the exact-stem, the MFE, the size of the terminal loop, the percentage of A, C, G and U, the number of consecutive A, C, G and U and the ratio of GU pairings in the stem (Supplementary File 1).

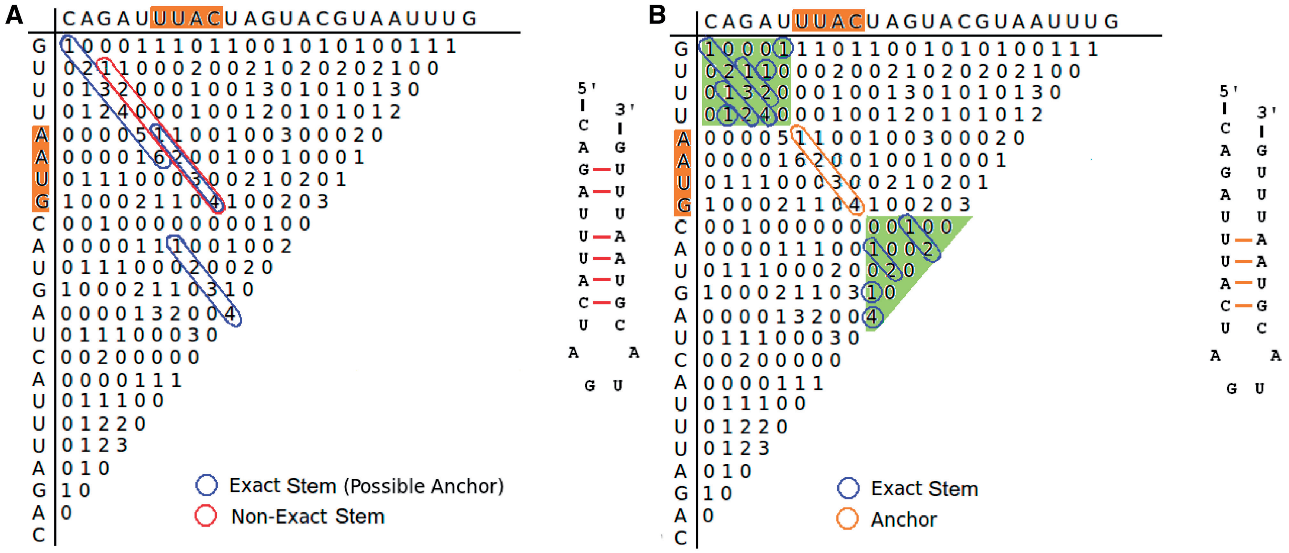
Longest non-exact stem searching. When an exact stem is selected, it is used as an ‘anchor’ for finding a non-exact stem. The extension of the exact stem is done by considering only the diagonal containing the exact stem. The diagonal is searched to the left and right of the anchor. For example, in Figure 2A, a non-exact stem is indicated by red and corresponds to two exact stems. Once a longest non-exact stem is extended, the 17 non-exact parameters are calculated: the total length, the number of exact stems composing it, the MFE, the length of the terminal loop, etc. The stem is selected if it meets a certain percentage of these parameters.

Each selected non-exact stem can correspond to a large portion of a pre-miRNA. It is therefore considered as a good approximation of a pre-miRNA hairpin, and gives the hairpin position. Possible pre-miRNA hairpins are then searched for in the subsequence associated to the selected non-exact stem, considering the middle position of the non-exact stem as the middle position of the hairpin.

Hairpin formation. The hairpins are predicted from selected non-exact stems. We consider that a selected non-exact stem approximates well a pre-miRNA hairpin structure. The anchor of the considered non-exact stem is positioned in the matrix, and then is extended inwards left and right (Figure 2B, light green areas) on different diagonals, in order to allow here bulges and non-symmetrical internal loops.

The hairpins determined from the matrix must verify a set of criteria in order to be selected (26 criteria): length, MFE, size of the terminal loop, ratio of base pair GU and GC, etc. (Supplementary File 1). Only hairpins where the percentage of verified criteria is higher than a certain percentage are selected.

Complexity of the algorithm. The algorithm uses a sliding window of a given size L on the sequence of size N.



```
 | C A G A U U U A C U A G U A C G U A A U U U G
G| 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 1 1 1
U| 0 2 1 1 0 0 0 2 0 0 2 1 0 2 0 2 0 2 1 0 0
U| 0 1 3 2 0 0 0 1 0 0 1 3 0 1 0 1 0 1 3 0
U| 0 1 2 4 0 0 0 1 0 0 1 2 0 1 0 1 0 1 2
A| 0 0 0 0 5 1 1 0 0 1 0 0 3 0 0 0 2 0
A| 0 0 0 0 1 6 2 0 0 1 0 0 1 0 0 0 1
U| 0 1 1 1 0 0 0 3 0 0 2 1 0 2 0 1
G| 1 0 0 0 2 1 1 0 4 1 0 0 2 0 3
C| 0 0 1 0 0 0 0 0 0 0 0 1 0 0
A| 0 0 0 0 1 1 1 0 0 1 0 0 2
U| 0 1 1 1 0 0 0 2 0 0 2 0
G| 1 0 0 0 2 1 1 0 3 1 0
A| 0 0 0 0 1 3 2 0 0 4
U| 0 1 1 1 0 0 0 3 0
C| 0 0 2 0 0 0 0 0
A| 0 0 0 0 1 1 1
U| 0 1 1 1 0 0
U| 0 1 2 2 0
U| 0 1 2 3
A| 0 1 0
G| 1 0
A| 0
C|
```

```
 | C A G A U U U A C U A G U A C G U A A U U U G
G| 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 1 0 0 1 1 1
U| 0 2 1 1 0 0 0 2 0 0 2 1 0 2 0 2 0 2 1 0 0
U| 0 1 3 2 0 0 0 1 0 0 1 3 0 1 0 1 0 1 3 0
U| 0 1 2 4 0 0 0 1 0 0 1 2 0 1 0 1 0 1 2
A| 0 0 0 0 5 1 1 0 0 1 0 0 3 0 0 0 2 0
A| 0 0 0 0 1 6 2 0 0 1 0 0 1 0 0 0 1
U| 0 1 1 1 0 0 0 3 0 0 2 1 0 2 0 1
G| 1 0 0 0 2 1 1 0 4 1 0 0 2 0 3
C| 0 0 1 0 0 0 0 0 0 0 0 1 0 0
A| 0 0 0 0 1 1 1 0 0 1 0 0 2
U| 0 1 1 1 0 0 0 2 0 0 2 0
G| 1 0 0 0 2 1 1 0 3 1 0
A| 0 0 0 0 1 3 2 0 0 4
U| 0 1 1 1 0 0 0 3 0
C| 0 0 2 0 0 0 0 0
A| 0 0 0 0 1 1 1
U| 0 1 1 1 0 0
U| 0 1 2 2 0
U| 0 1 2 3
A| 0 1 0
G| 1 0
A| 0
C|
```

Figure 2. (**A**) Example of a symmetrical matrix for searching for exact and non-exact stems in a given genomic subsequence. Three stems are selected with a threshold of minimum length equal to 4 (surrounded by a blue circle). One of the three stems has been extended to a non-exact stem (surrounded by a red circle). (**B**) Search for hairpins. The anchor (surrounded by an orange circle) of the non-exact stem shown in (A) is positioned in the matrix, and then is extended in left and in right (green areas) on different diagonals, in order to allow bulges and internal loops.

The search of hairpins in the window is of time and space complexity of $O(L^2)$. Therefore, the total time complexity of the algorithm is $O(L^2 \cdot N)$, where N is the sequence length. The space complexity is of $O(L^2 + N)$.

RESULTS AND DISCUSSION

The data

To test our method, we considered two types of sequences: (i) an artificial sequence obtained by concatenating known pre-miRNAs with mRNA sequences, and (ii) real genomic sequences where a great number of pre-miRNAs are known.

Artificial sequence. The artificial sequence was created by the concatenation of human mRNAs and the insertion of 100 human pre-miRNAs. The mRNA sequences came from the Human genome (build 37.2) of the NCBI Website (www.ncbi.nlm.nih.gov) and the pre-miRNAs came from miRBase database (release 17) (www.mirbase.org). Pre-miRNA lengths are from 63 to 110 nt and the start position of pre-miRNAs begins every 300 nt, the first pre-miRNA starting at position 300. The total length of the artificial sequence is 30 500 nt. The obtained sequence is given in Supplementary File 2.

Real genomic sequences. We considered for our tests four genomes: human, mouse, zebrafish and sea squirt. We chose these genomes as they present a large cluster of known miRNAs:

- The human chromosome 19 (strand '+') has a cluster of 50 pre-miRNAs, the first pre-miRNA starting at position 54,169,933 and the last one ending at position 54,485,651.
- The mouse chromosome 2 (strand '+') has a cluster of 71 pre-miRNAs, the first one starting at position 10,388,290 and the last one ending at position 10,439,906.
- The zebrafish chromosome 4 (strand '–') has a cluster of 50 pre-miRNAs, the first one starting at position 34,353,975 and the last one ending at position 34,481,435.
- The sea squirt chromosome 7q (strand '–') has a cluster of 46 pre-miRNAs, the first one starting at position 5,400,066 and the last one ending at position 6,168,570.

For each of these genomes, we extracted from NCBI Website the sub-sequence that includes the considered pre-miRNAs cluster.

Tested algorithms

In order to evaluate our algorithm, we compared it to existing algorithms of the same category (the third category of *ab-initio* algorithms), i.e. with *ab-initio* algorithms searching for pre-miRNA structures in genomes. We can cite five programs in this category: CID-miRNA (22), miRPara (23), miRPred (24), miRANK (25), Virgo (28) and VMir (26).

Unfortunately, we could not access the source code, binary or web server of miRPred and miRANK. Virgo has only a web server (http://miracle.igib.res.in/virgo/) that limits the size of input sequences to 5000 pb. This size is too short for our data sets and for genome screening. We therefore considered CID-miRNA, miRPara and VMir for our tests. All have their binaries available and take as input the genome sequence in a FASTA format. To our knowledge, only CID-miRNA has a Webserver (http://mirna.jnu.ac.in/cidmirna/).

CID-miRNA, which is dedicated to human genome, uses a sliding window for parsing the genomic sequence. On each subsequence delimited by the window, it uses a Cocke-Younger-Kasami (CYK) parser to build the most likely secondary structures and uses a classification tree [obtained with WEKA (http://www.cs.waikato.ac.nz/ml/weka) on training data] to determine if the secondary structure is a pre-miRNA.

miRPara also uses a sliding window and on each subsequence uses first UNAFold (16) for predicting the secondary structure of the pre-miRNA candidate. miRPara calculates 77 parameters and uses a SVM classifier to select or not the candidate.

VMir, which is dedicated to viruses pre-miRNA prediction, also uses a sliding window and on each subsequence delimited by the window, performs RNAFold (13) to predict the secondary structures. VMir calculates a score based on the size of the terminal loop, the number of base pairs and the size of the bulges for each hairpin in the sliding window. The hairpins with a score higher than a given threshold are then selected.

For the three *ab-initio* software CID-miRNA, VMir and miRNAFold, we used a sliding window of 150 nt. We also considered their default parameters, except for CID-miRNA, where we changed the parameter value of the hairpin minimal size. By default, CID-miRNA sets the minimal size of hairpin to 60 bp, but the ciona intestinalis cluster we chose has some pre-miRNAs range from 47 bp to 61 bp, so we chose as minimal size 40 nt.

A predicted hairpin does not always correspond exactly to the functional hairpin *in vivo*. Therefore, we consider that a known pre-miRNA is correctly predicted if the returned position is correct. The position of a hairpin is considered as its center and we assume that a predicted pre-miRNA corresponds to a known pre-miRNA if the distance between the known and the predicted center is lower than 10% of the hairpin size.

Statistical measures

In order to evaluate and compare the tested programs, we used the measures of sensitivity and selectivity (specificity). The sensitivity measures the capability of the software to find known pre-miRNAs. The selectivity represents the probability that a predicted hairpin corresponds to a pre-miRNA. The sensitivity and the selectivity are given by the following equations:

$$\text{Sensitivity} = 100 \cdot \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Selectivity} = 100 \cdot \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where TP (true positives) is the number of correctly predicted pre-miRNA, FN (false negatives) is the number of non correctly predicted pre-miRNA and FP (false positives) is the number of wrong pre-miRNAs predicted.

Results on the artificial sequence. An important parameter of miRNAFold is the minimal percentage of criteria that must be verified at each step of the algorithm. To select this important parameter, we ran miRNAFold on the artificial sequence.

Table 1. Results obtained by miRNAFold, CID-miRNA, miRPara and Vmir on an artificial sequence

	Sensitivity	Selectivity	Time (min : *s*)
CID-miRNA	97	11.72	90:49
miRPara	97	9.7	5:24
VMir	28	1.32	2:32
miRNAFold$_{50}$	**98**	18.77	0:0.88
miRNAFold$_{60}$	**98**	18.96	0:0.88
miRNAFold$_{70}$	97	19.17	0:0.84
miRNAFold$_{80}$	96	22.91	0:0.76
miRNAFold$_{90}$	65	**52**	**0:0.68**

Comparison of prediction results obtained on an artificial sequence by miRNAFold, CID-miRNA, miRPara and Vmir. miRNAFold was run with different values for the parameter of minimal percentage of verified criteria: 50, 60, 70, 80 and 90. Values in bold denote best results.

The results of sensitivity and selectivity obtained by miRNAFold, CID-miRNA, miRPara and Vmir are given in Table 1. Because the artificial sequence contains 100 pre-miRNAs, the sensitivity shown in Table 1 corresponds to the number of pre-miRNAs correctly predicted (true positives).

CID-miRNA, miRPara, VMir, miRNAFold$_{50}$, miRNAFold$_{60}$, miRNAFold$_{70}$, miRNAFold$_{80}$ and miRNAFold$_{90}$ predicted a total of 288, 2968, 7565, 529, 525, 514, 419 and 126 hairpins, respectively. CID-miRNA, miRPara, VMir, miRNAFold$_{50}$, miRNAFold$_{60}$, miRNAFold$_{70}$, miRNAFold$_{80}$ and miRNAFold$_{90}$ have predicted, respectively, 28, 97, 97, 98, 98, 97, 96 and 65% of the real pre-miRNAs. As expected, the lower the percentage parameter value considered in miRNAFold, the higher the sensitivity. The higher the percentage, the higher the selectivity.

miRNAFold was always more selective than the other methods whatever the considered criteria percentage. miRNAFold$_{50}$, miRNAFold$_{60}$, miRNAFold$_{70}$ found at least 10 times less false pre-miRNAs than VMir whereas they found the same number of true pre-miRNAs. miRNAFold$_{80}$ is 15 times more selective when it losts only one true pre-miRNA. miRPara found the same number of pre-miRNAs than miRNAFold$_{70}$ but it also predicted 2436 more pre-miRNAs, which corresponded to five times the number of pre-miRNAs predicted by miRNAFold$_{70}$.

miRNAFold is faster than the other tested methods: in the worst case (with miRNAFold$_{50}$), miRNAFold was about 120 times faster than Vmir, which is the fastest tested method.

When we increase the percentage of criteria from 50% to 80%, miRNAFold misses only 2 pre-miRNAs but removes more than 109 false pre-miRNAs. These thresholds allow the user, in a simple way, to choose between the discovery of a maximum number of pre-miRNAs with numerous false positives or the discovery of some pre-miRNAs with a lower number of false positives.

In the following, we set to 70% the default value of this parameter as it represents the percentage giving a good compromise between the sensitivity and the selectivity.

Results on the real genomic sequences. We tested CID-miRNA, miRNAFold, miRPara and VMir on the four genomic sequences described above (human, mouse, zebrafish and sea squirt), which contain each a cluster of several known miRNAs. miRNAFold was run with a threshold of 70% for its parameter of minimum percentage of verified criteria.

Sensitivity results

Table 2 shows the sensitivity results obtained with CID-miRNA, miRNAFold, miRPara and VMir in each of the considered sequences.

For the human sequence, miRNAFold and VMir succeeded both to predict all known pre-miRNAs. For the mouse sequence, our algorithm has a better prediction rate than VMir: 98.59% versus 88.73% which means that miRNAFold missed one known pre-miRNA when VMir missed eight pre-miRNAs. For the zebrafish sequence, miRNAFold missed only three known pre-miRNAs (94.74%) when VMir missed nine known pre-miRNAs (84.21%). For the sea squirt sequence, VMir found all known pre-miRNAs while our algorithm missed four pre-miRNAs (91.30%).

In the human sequence, CID-miRNA missed 31 of the 50 known pre-miRNAs where miRNAFold found all of them. For the mouse sequence, CID-miRNA found less than one-third of known pre-miRNAs (21 of 71) when miRNAFold missed only one pre-miRNA. For the zebrafish and the sea squirt sequences, the sensitivity of CID-miRNA decreases to 19.30% and 28.26%.

In the human sequence, miRPara missed one known pre-miRNA while miRNAFold found all of them. In the mouse sequence, the two methods have the same sensitivity (98.58%), which is the best sensibility rate for the mouse sequence. For the zebrafish and the sea squirt sequences, the sensitivity of miRPara dropped to 47.37 and 58.7%, respectively, when miRNAFold has a sensitivity rate greater than 91%.

Thus we can say that miRNAFold has a better or similar sensitivity than VMir in three of the four genomic sequences. However, the sensitivity rate of miRNAFold is always higher than 90% while the sensitivity of VMir can decrease under 85%. CID-miRNA is the least sensitive of the four programs. For the four genomic sequences, the sensitivity of CID-miRNA is lower than 40%. The higher sensitivity value is for the human sequence, which confirms that CID-miRNA was originally built for human pre-miRNAs. miRPara has a good sensitivity for the mammalian genomes (human and mouse), but its sensitivity dropped under 60% for zebrafish and sea squirt genomes. Finally, miRNAFold is the only algorithm giving a sensitivity always greater than 90% for all tested sequences. Unlike the other programs, it gives homogeneous and stable sensitivity results whatever the genomic sequence.

Selectivity results

Table 3 shows the selectivity results obtained by miRNAFold, CID-miRNA, miRPara and VMir on the four considered sequences.

Table 2. Sensitivity of CID-miRNA, miRNAFold, miRPara and VMir

	Human	Mouse	Zebrafish	Sea squirt
CID-miRNA	38	29. 58	19.30	28.26
miRPara	98	**98.59**	47.37	58.7
VMir	**100**	88.73	84.21	**100**
miRNAFold$_{70}$	**100**	**98.59**	**94.74**	91.30

Sensitivity results obtained by CID-miRNA, miRNAFold, miRPara and VMir on Human, Mouse, Zebrafish and Sea squirt genomic sequences. Values in bold denote best results.

Table 3. Selectivity of CID-miRNA, miRNAFold, miRPara and VMir

	Human	Mouse	Zebrafish	Sea squirt
CID-miRNA	0.69	0.82	0.75	**10.88**
miRPara	**0.93**	5.34	1.4	5.86
VMir	0.56	2.93	1.35	5.29
miRNAFold$_{70}$	0.89	**7.71**	**2.60**	7.98

Selectivity results of CID-miRNA, miRNAFold, miRPara and VMir obtained on Human, Mouse, Zebrafish and Sea squirt genomic sequences. Values in bold denote best results.

Compared with Vmir, miRNAFold has a higher selectivity in all sequences. The selectivity of miRNAFold is about two times better than the selectivity of VMir in mouse and zebrafish. This means that miRNAFold finds two times less pre-miRNAs than VMir. For example, for the mouse sequence, VMir predicted 2149 hairpins when miRNAFold predicted only 913 for the whole genomic sequence.

miRNAFold has also a higher selectivity than CID-miRNA in all sequences excepted in sea squirt. miRNAFold is nine times better than the selectivity of CID-miRNA in mouse and almost four times better in zebrafish. It is also better in the human sequence. The selectivity of CID-miRNA is better than the selectivity of miRNAFold in sea squirt: CID-miRNA predicted 323 putative pre-miRNAs while miRNAFold predicted 526 pre-miRNAs.

miRNAFold has also higher selectivity than miRPara in all sequences, except for the human genome. The sensitivity rate of miRNAFold was almost two times greater in the zebrafish genome.

To summarize, miRNAFold has better sensitivity and selectivity results than CID-miRNA, miRPara and Vmir on the mouse and zebrafish sequences. In human genomic sequence, miRPara has a slightly better selectivity than miRNAFold, but it has a lower sensitivity. In sea quirt genomic sequence, VMir predicts only four supplementary known pre-miRNAs compared with miRNAFold but VMir also predicted 344 false supplementary hairpins. CID-miRNA has better selectivity in this genome sequence but missed 33 pre-miRNAs when miRNAFold missed only four pre-miRNAs.

Table 4. Run time of CID-miRNA, miRNAFold, miRPara and VMir

	Human	Mouse	Zebrafish	Sea squirt	Average
CID-miRNA	54 h 58 m	54 h 48 m	54 h 40 m	55 h 29 m	55 h 08 m
miRPara	20 h 12 m	19 h 47 m	19 h 40 m	19 h 25 m	19 h 46 m
VMir	30 m	30 m	30 m	30 m	30 m
miRNAFold$_{70}$	**0 m 25 s**	**0 m 22 s**	**0 m 29 s**	**0 m 24 s**	**0 m 25 s**

Execution time of the algorithms CID-miRNA, miRNAFold, miRPara and VMir for predicting pre-miRNAs in genomic sequences of 1 million of nucleotides each in the four species Human, Mouse, Zebrafish and Sea squirt. The values of miRNAFold was rounded to the second. The values of CID-miRNA, miRPara, and Vmir were rounded to the minutes. The last column shows the average execution time for a sequence of 1 million of nucleotides. Values in bold denote best results.

Running time

With the increase of sequencing of large genomes, the running time is an important evaluation parameter of pre-miRNA searching algorithms.

To compare the run time of CID-miRNA, miRNAFold, miRPara, and VMir, we considered subsequences of 1 million nucleotides beginning at positions 54.000.000, 10.000.000, 34.000.000 and 5.400.000 from the Human, Mouse, Zebrafish and Sea squirt genomes, respectively, containing the clusters considered above.

Experiments were performed on a Linux machine equipped with an Intel Core Duo 2 T6600 of 2.2 GHz and 4 GB of RAM. The execution time of the three programs on the four sequences is given in Table 4.

miRNAFold is the fastest algorithm. Our average time execution is 25 s for a sequence of 1 million of nucleotides, when VMir, the second fastest algorithm, has an average time execution of 30 min. CID-miRNA has an average time execution of 55 h and the average time of miRPara is almost 20 h.

miRNAFold is then almost 60 times faster than VMir, about 2400 times faster than miRPara, and about 6600 times faster than CID-miRNA.

AVAILABILITY AND IMPLEMENTATION

miRNAFold takes in input a genomic sequence in a Fasta format. The size of the sliding window is a parameter which could be set by the user (by default equal to 150 nt). Another important parameter is the percentage of criteria that must be verified at each step of the algorithm. The value of this parameter is set by default to 70% and the user can vary it between 0% and 100%.

miRNAFold was implemented using the C++ language. The software can be used through the web server: http://EvryRNA.ibisc.univ-evry.fr.

CONCLUSION

We presented here an original *ab-initio* method called miRNAFold, which allows a fast search for miRNA precursors in genomes. This method first searches for the position of pre-miRNAs by approximating their structure before deducing the final structure. The interest of this first step is to reduce the run time. miRNAFold searches for long exact stems that are then extended into long non-exact stems. The position of a selected non-exact stem represents position of a possible pre-miRNA, which structure is then predicted in a fast way. miRNAFold uses a sliding window, where all possible pre-miRNAs are searched for. The algorithm has a time complexity of $O(L^2 \cdot N)$, where L is the length of the window, and N the size of the sequence.

miRNAFold was tested on several genomic sequences, and was compared with CID-miRNA, miRPara and VMir. miRNAFold has almost always better sensitivity. It is the only one algorithm giving a sensitivity always greater than 90%. Unlike the other programs, its sensitivity is homogeneous and stable whatever the genomic sequence. However, the selectivity of miRNAFold is not satisfactory, even if it is better than the selectivity of almost all other methods. Decrease substantially the number of false positives is a challenging problem. We are currently developing clustering and machine learning methods for improving the selectivity of our algorithm.

An important advantage of our method compared with existing ones is the run time. We obtain better (or similar) sensitivity and selectivity results than other existing methods but with an average running time at least 60 times faster than the fastest tested algorithm, i.e. Vmir. On the tested sequences, miRNAFold takes <30 s for a sequence of 1 million length, when VMir takes >30 min, miRPara takes about 20 h and CID-miRNA >55 h. Our method is the only one that permits whole genome analysis.

The different criteria thresholds were defined from observations we done on miRBase hairpins. One of our further work is to develop automatic learning methods in order to define automatically these thresholds. Another further work is to optimize and adapt our code for using it on HPC solutions, and more precisely on GPU solutions, in order to make it much faster for whole genomes. Finally, we are working with biologists in order to use miRNAFold for finding new pre-miRNAs in genomes, and more precisely on the *Xenopus tropicalis* genome.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR online: Supplementary File 1–2.

ACKNOWLEDGEMENTS

We would like to thank Mikael Trellet and Mederich Besnard for developing the web server of miRNAFold.

FUNDING

Council of Essonne Region (Pôle System@tic, OpenGPU project). Funding for open access charge: Council of Essonne Region.

Conflict of interest statement. None declared.

REFERENCES

1. Bartel,D. (2004) MicroRNAs: genomics, biogenesis, mechanism and function. *Cell*, **116**, 281–197.
2. He,L. and Hannon,G. (2004) microRNAs: small RNAs with a big role in gene regulation. *Nat. Rev. Genet*, **5**, 522–531.
3. Lee,Y., Kim,M., Han,J., Yeom,K., Lee,S., Baek,S. and Kim,V. (2004) microRNA genes are transcribed by RNA polymerase II. *EMBO J.*, **23**, 4051–4060.
4. Wang,Y., Stricker,H.M., Gou,D. and Liu,L. (2007) microRNA: past and present. *Front. Biosci.*, **12**, 2316–2329.
5. Lai,E.C., Tomancak,P., Williams,R.W. and Rubin,G.M. (2003) Computational identification of Drosophila microRNA genes. *Genome Biol.*, **4**, R42.
6. Huang,T.H., Fan,B., Rothschild,M.F., Hu,Z.L., Li,K. and Zhao,S.H. (2007) MiRFinder: an improved approach and software implementation for genome-wide fast microRNA precursor scans. *BMC Bioinformatics*, **8**, 341.
7. Hertel,J. and Stadler,P.F. (2006) Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, **22**, e197–e202.
8. Yousef,M., Nebozhyn,M., Shatkay,H., Kanterakis,S., Showe,L.C. and Showe,M.K. (2006) Combining multi-species genomic data for microRNA identification using a Naïve Bayes classifier. *Bioinformatics*, **22**, 1325–1334.
9. Terai,G., Komori,T., Asai,K. and Kin,T. (2007) miRRim: a novel system to find conserved miRNAs with high sensitivity and specificity. *RNA*, **13**, 2081–2090.
10. Griffiths-Jones,S., Saini,H.K., van Dongen,S. and Enright,A.J. (2008) miRBase: tools for microRNA genomics. *Nucleic Acids Res.*, **36**, D154–D158.
11. Wang,X., Zhang,J., Li,F., Gu,J., He,T., Zhang,X. and Li,Y. (2005) microRNA identification based on sequence and structure alignment. *Bioinformatics*, **21**, 3610–3614.
12. Legendre,M., Lambert,A. and Gautheret,D. (2005) Profile-based detection of microRNA precursors in animal genomes. *Bioinformatics*, **21**, 841–845.
13. Hofacker,I.L., Fontana,W., Stadler,P.F., Bonhoeffer,S., Tacker,M. and Schuster,P. (1994) Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, **125**, 167–188.
14. Hofacker,I.L., Priwitzer,B. and Stadler,P.F. (2004) Prediction of locally stable RNA secondary structures for genome-wide surveys. *Bioinformatics*, **20**, 186–190.
15. Zuker,M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, **31**, 3406–3415.
16. Markham,N.R. and Zuker,M. (2005) DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res.*, **33**, W577–W581.
17. Xue,C., Li,F., He,T., Liu,G.P., Li,Y. and Zhang,X. (2005) Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, **6**, 310.
18. Chang,D.T., Wang,C.C. and Chen,J.W. (2008) Using a kernel density estimation based classifier to predict species-specific microRNA precursors. *BMC Bioinformatics*, **9**, 12.
19. Jiang,P., Wu,H., Wang,W., Ma,W., Sun,X. and Lu,Z. (2007) MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res.*, **35**, W339–W344.
20. Batuwita,R. and Palade,V. (2009) microPred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, **25**, 989–995.
21. Kwang Loong,S.N.G. and Mishra,S.K. (2007) Unique folding of precursor microRNAs: Quantitative evidence and implications for de novo identification. *RNA*, **13**, 170–187.
22. Sewer,A., Paul,N., Landgraf,P., Aravin,A., Pfeffer,S., Brownstein,M.J., Tuschl,T., van Nimwegen,E. and Zavolan,M. (2005) Identification of clustered microRNAs using an ab initio prediction method. *BMC Bioinformatics*, **6**, 267.
23. Mathelier,A. and Carbone,A. (2010) MIReNA: finding microRNAs with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, **26**, 2226–2234.
24. Tyagi,S., Vaz,C., Gupta,V., Bhatia,R., Maheshwari,S., Srinivasan,A. and Bhattacharya,A. (2008) CID-miRNA: a web server for prediction of novel miRNA precursors in human genome. *Biochem. Biophys. Res. Comm.*, **372**, 831–834.
25. Wu,Y., Wei,B., Liu,H., Li,T. and Rayner,S. (2011) MiRPara: a SVM-based software tool for prediction of most probable microRNA coding regions in genome scale sequences. *BMC Bioinformatics*, **12**, 107.
26. Brameier,M. and Wiuf,C. (2007) Ab initio identification of human microRNAs based on structure motifs. *BMC Bioinformatics*, **8**, 478.
27. Xu,Y., Zhou,X. and Zhang,W. (2008) MicroRNA prediction with a novel ranking algorithm based on random walks. *Bioinformatics*, **24**, 50–58.
28. Kumar,S., Ansari,F.A. and Scaria,V. (2009) Prediction of viral microRNA precursors based on human microRNA precursor sequence and structural features. *Virol J.*, **6**, 129.
29. Grundhoff,A., Sullivan,C.S. and Ganem,D. (2006) A combined computational and microarray-based approach identifies novel microRNAs encoded by human gamma-herpesviruses. *RNA*, **12**, 733–750.
30. Joachims,T. (1999) Making large-scale support vector machine learning practical. *MIT Press*, **11**, 169–184.
31. Engelen,S. and Tahi,F. (2010) Tfold: efficient in silico prediction of non-coding RNA secondary structures. *Nucleic Acids Res.*, **38**, 2453–2466.
32. Tempel,S., Rousseau,C., Tahi,F. and Nicolas,J. (2010) ModuleOrganizer: detecting modules in families of transposable elements. *BMC Bioinformatics*, **11**, 474.
33. Helvik,S.A., Snove,O.J. and Saetrom,P. (2007) Reliable prediction of Drosha processing sites improves microRNA gene prediction. *Bioinformatics*, **23**, 142–149.
34. van der Burgt,A., Fiers,M.W.J.E., Nap,J.P. and van Ham,R.C.H.J. (2009) In silico miRNA prediction in metazoan genomes: balancing between sensitivity and specificity. *BMC Genomics*, **10**, 204.